Darwin Calibrator Methodology

Computer models have become an essential tool for the management of water distribution systems around the world. There are numerous purposes for use of a computer model to simulate the flow conditions within a system. A model can be employed to ensure the adequate quantity and quality service of the potable water resource to the community, evaluate the planning and design alternatives, assess the system performance and to verify a operating strategy for better management of the water infrastructure system, as well as to be able to perform vulnerability studies to assess risks that may be presented and affect the water supply. For these purposes, a model is constructed in which data describing network elements of pipes, junctions, valves, pumps, tanks, and reservoirs are assembled in a systematic manner to predict pipe flow and junction hydraulic grade lines (HGL) or pressures within a water distribution system.

Computer models that have been established over last twenty years and that are to be constructed in future are significant investments for water companies. To ensure a good investment return or correct usages of the models, the model must be capable of correctly simulating flow conditions encountered at the site. This is achieved by calibrating the models. A calibration involves the process of adjusting model characteristics and parameters so that the model's predicted flows and pressures match actual observed field data to some desirable or acceptable level. This is described in more detail in Walski, Chase and Savic (2001).

Calibration of a water distribution model is a complicated task. There are many uncertain parameters that need to be adjusted to reduce the discrepancy between the model predictions and field observations of junction HGL and pipe discharges. Pipe roughness coefficients are often considered for calibration. However, there are many other parameters that are uncertain and affect junction HGL and pipe flow rate. To minimize errors in model parameters and eliminate the compensation error of calibration parameters (Walski 2001), you should consider calibrating all the model parameters, such as junction demand, operation status of pipes and valves, and pipe roughness coefficients.

Calibrating water distribution network models relies upon field measurement data, such as junction pressures, pipe flows, water levels in storage facilities, valve settings, and pump operating status (on/off) and speeds. Among all the possible field observation data, junction HGL and pipe flows are often used to evaluate goodness-of-fit of the model calibration. The other parameters of tank levels, valve settings, and pump operating status/speed are used as boundary conditions that are recorded when collecting a set of calibration observation of junction pressures and pipe flow rates.

Field observation data are measured and collected at different times of the day and at various locations on site, which may correspond to various demand loadings and boundary conditions. In order that that the model simulation results more closely represent the observed data, the simulation results must use the same demand loading and boundary conditions as the observed data. Thus, the calibration process must be conducted under multiple demand loading and operating boundary conditions.

Traditional model calibration of a water distribution model is based on a trial-and-error procedure, by which an engineer or modeler first estimates the values of model parameters, then runs the model to obtain a predicted pressure and flow, and finally compares the simulated values to the observed data. If the predicted data does not compare closely with the observed data, the engineer returns to the model, makes some adjustments to the model parameters, and runs it again to produce a new set of simulation results. This may have to be repeated many times to make sure that the model produces a close-enough prediction of the water distribution network in the real world. The traditional calibration technique is, among other things, quite time consuming.

In addition, a typical network representation of a water network may include hundreds or thousands of links and nodes. Ideally, during a water distribution model calibration process, the roughness coefficient is adjusted for each link and demand is adjusted for each node. However, only a small percentage of representative sample measurements can be made available for the use of model calibration, due to the limited financial and labor requirements for data collection. Therefore, it is of utmost importance to have a comprehensive methodology and efficient tool that can assist the modeler and engineer in achieving a highly accurate model under practical conditions, including various model parameters such as pipe roughness, junction demand, and link status, and also multiple demand and boundary conditions.

Calibration Formulation

An optimization calibrator is formulated and developed for facilitating the calibration process of a water distribution model. The parameters are obtained by minimizing the discrepancy between the model-predicted and the field-observed values of junction pressures (hydraulic grades) and pipe flows for given boundary conditions. The optimized calibration is then defined as a nonlinear optimization problem with three different calibration objectives.

Calibration Objectives

The goodness-of-fit of model calibration is evaluated by the discrepancy between the model simulated and field measured junction HGL and pipe flow. The goodness-of-fit score is calculated by using a user-specified fitness-point-per-hydraulic head for junctions and fitness-point-per-flow for pipes. This allows a modeler to flexibly weight the evaluation of both pipe flow and junction hydraulic head. Three fitness functions are defined as follows.

Objective Type One: Minimize the sum of difference squares

$$minimize \qquad \frac{\sum_{np=1}^{NH} w_{nh}\left(\frac{Hsim_{nh}-Hobs_{nh}}{Hpnt}\right)^2+\sum_{nf=1}^{NF} w_{nf}\left(\frac{Fsim_{nf}-Fobs_{nf}}{Fpnt}\right)^2}{NH+NF} \qquad (1)$$

Objective Type Two: Minimize the sum of absolute differences

$$minimize \qquad \frac{\sum_{np=1}^{NH} w_{nh}\left|\frac{Hsim_{nh}-Hobs_{nh}}{Hpnt}\right|+\sum_{nf=1}^{NF} w_{nf}\left|\frac{Fsim_{nf}-Fobs_{nf}}{Fpnt}\right|}{NH+NF} \qquad (2)$$

Objective Type Three: Minimize the maximum absolute difference

$$minimize \qquad \max\left\{\max_{nh=1}^{NH} w_{nh}\left|\frac{Hsim_{nh}-Hobs_{nh}}{Hpnt}\right|, \max_{nf=1}^{NF} w_{nf}\left|\frac{Fsim_{nf}-Fobs_{nf}}{Fpnt}\right|\right\} \qquad (3)$$

where $Hobs_{nh}$ designates the *nh*-th observed hydraulic grade, $Hsim_{nh}$ is the *nh*-th model simulated hydraulic grade, $Hloss_{nh}$ is the head loss at observation data point *nh*, $Fobs_{nf}$ is the observed flow, $Fsim_{nf}$ is model simulated flow, *Hpnt* notes the hydraulic head per fitness point while *Fpnt* is the flow per fitness point, *NH* is the number of observed hydraulic grades and *NF* is the number of observed pipe discharges, W_{nh} and W_{nf} represent a normalized weighting factor for observed hydraulic grades and flows respectively. They are given as:

$$W_{nh} = \mathrm{f}(Hloss_{nh} / \sum Hloss_{nh}) \qquad (4)$$

$$W_{nf} = \mathrm{f}(Fobs_{nf} / \sum Fobs_{nf}) \qquad (5)$$

where f( ) is a function which can be linear, square, square root, log, or constant. An optimized calibration can be conducted by selecting one of three objectives above and the weighting factors between head and flow. The model parameters are calculated by using a genetic algorithm while minimizing the selected objective function and satisfying the calibration constraints.

Calibration Constraints

Optimized calibration is conducted by satisfying two type constraints, the hydraulic system constraints and calibration parameter bound constraints. The system constraints are a set of implicit equations that ensure the conservation of flow continuity at nodes and energy for the loops within a water distribution system. Each trial solution generated by the GA is analyzed using WaterCAD hydraulic network solver.

The calibration bound constraints are used to set the minimum and maximum limits for the pipe roughness coefficients and junction demand multiplier. They are given as follows.

$$RFmin_i \le RF_i \le RFmax_i \qquad i = 1,2,3,\ldots, nPipeGroup \qquad (6)$$

$$DMmin_i \le DM_i \le DMmax_i \qquad i = 1,2,3,\ldots, nDemandGroup \qquad (7)$$

where $RFmin_i$ is the minimum roughness coefficient or multiplier for roughness group *i*; $RFmax_i$ is the maximum roughness coefficient or multiplier for roughness group *i*; and RF_i is the roughness coefficient or multiplier for roughness group *i*, $DMmin_i$ is the minimum junction demand multiplier for demand group *i*; $DMmax_i$ is the maximum demand multiplier for demand group *i*; and DM_i is the demand multiplier for demand group *i.*

The pipes that have the same physical and hydraulic characteristics are allowed to be grouped as one calibration link, and one new roughness coefficient or one roughness coefficient multiplier is assigned to all the pipes in the same group. The junctions that have the same demand patterns and within a same topological area can also be aggregated as one calibration junction, to which a same demand multiplier is calculated and assigned. Calibration parameters are bounded by prescribed upper and lower limits and adjusted with a user-prescribed incremental value. For example, a Hazen-Williams C value for a pipe or a group of pipes will be computed within a range of 40 to 140, and by an increment of 5. Demand multipliers may range from 0.8 to 1.2 by 0.1. Parameter aggregation is useful at reducing the calibration dimension, however caution needs to be excised at pipe and junction grouping, which may affect the accuracy of the model calibration.

Genetic Algorithm Optimized Calibration

Genetic algorithm (GA) is a robust search paradigm based on the principles of natural evolution and biological reproduction (Goldberg, 1989). For optimizing calibration of a water distribution model, a genetic algorithm program first generates a population of trial solutions of the model parameters. A hydraulic solver then simulates each trial solution. The resulting hydraulic simulation predicts the HGL (junction pressures) and pipe flows at a predetermined number of nodes (or data points) in the network. This information is then passed back to the associated calibration module. The calibration module evaluates how closely the model simulation is to the observed data, the calibration evaluation computes a "goodness-of-fit" value, which is the discrepancy between the observed data and the model predicted pipe flows and junction pressures or HGL, for each solution. This goodness-of-fit value is then assigned as the "fitness" for that solution in the genetic algorithm.

One generation produced by the genetic algorithm is then complete. The fitness measure is taken into account when performing the next generation of the genetic algorithm operations. To find the optimal calibration solutions, fitter solutions will be selected by mimicking Darwin's natural selection principal of "survival of the fittest." The selected solutions are used to reproduce a next generation of calibration solutions by performing genetic operations. Over many generations, the solutions evolve, and the optimal or near optimal solutions ultimately emerge. There are numerous variations of genetic algorithms over last decade. Many successful applications of GA to solving model calibration have been carried for optimized calibration of water resource systems (Wang 1992; Wu 1994; Babovic etc. 1994; Wu and Larsen 1996). More recently, a competent genetic algorithm (also called fast messy GA), which has been demonstrated the most efficient GA for the optimization of a water distribution system (Wu & Simpson 2001), has been used for the optimized calibration. A brief overview is given in the following section.

Competent Genetic Algorithms

The working mechanics of a genetic algorithm is derived from a simple assumption (Holland 1975) that the best solution will be found in the solution region that contains a relatively high proportion of good solutions. A set of strings that represent the good solutions attains certain similarities in bit values. For example, 3-bit binary strings 001, 111, 101 and 011 have a common *similarity template* of **1, where asterisk (*) denotes a "don't-care" symbol that takes a value of either 1 or 0. The four strings represent four good solutions and contribute to the fitness values of 10, 12, 11, and 11 to a fitness function of $f(x_1, x_2, x_3) = x_1 + x_2 + 10^{x_3}$, where x_1, x_2 and x_3 directly takes a bit value as an integer from left to right. In general, a short similarity template that contributes an above-average fitness is called a "building block." Building blocks are often contained in short strings that represent partial solutions to a specific problem. Thus, searching for good solutions uncovers and juxtaposes the good short strings, which essentially designate a good solution region, and finally leads a search to the best solution.

Goldberg et al. (1989) developed the messy genetic algorithm as one of competent genetic algorithm paradigms by focusing on improving GA's capability of identifying and exchanging building blocks. The first-generation of the messy GA explicitly initializes all the short strings of a desired length *k*, where *k* is referred as to the order of a building block defined by a short string. For a binary string representation, all the combinations of order-*k* building blocks requires a number of $n = 2^k \binom{l}{k}$ initial short strings of length *k* for an *l*-bit problem. For example, the initial population size of short strings, by completely enumerating the building blocks of order 4 for a 40-bit problem, is more than one million. This made the application of the first-generation messy GA to a large-scale optimization problem impossible. This bottleneck has been overcome by introducing a building block filter procedure (Goldberg et al. 1993) into the messy GA. The filter procedure speeds up the search process and is called a fast messy GA.

The fast messy GA emulates the powerful genetic-evolutionary process in two nested loops, an outer loop and an inner loop. Each cycle of the outer loop, denoted as an era, invokes an initialization phase and an inner loop that consists of a building block filtering phase and a juxtapositional phase. Like a simple genetic algorithm, the messy GA initialization creates a population of random individuals. The population size has to be large enough to ensure the presence of all possible building blocks. Then a building block filtering procedure is applied to select better-fit short strings and reduce the string length. It works like a filter that "bad" genes not belonging to building blocks are deleted, so that the population contains a high proportion of short strings of "good" genes. The filtering procedure continues until the overall string length is reduced to a desired length *k*. Finally, a juxtapositional phase follows to produce new strings. During this phase, the processed building blocks are combined and exchanged to form offspring by applying the selection and reproduction operators. The juxtapositional phase terminates when the maximum number of generations is reached, and the cycle of one era iteration completes. The length of short strings that contains desired building blocks is often specified as the same as an era, starting with one to a maximum number of era. Because of this, preferred short strings increase in length over outer iterations. In another words, a messy GA evolves solutions from short strings starting from length one to a maximum desired length. This enables the messy GA to mimic the natural and biological evolution process that a simple or one cell organism evolves into a more sophisticated and intelligent organism. Goldberg et al. (1989, 1993) has given the detail analysis and computation procedure of the messy GA.

References

Babovic V., Wu Z. Y. & Larsen L. C. (1994), Calibrating Hydrodynamic Models by Means of Simulated Evolution, in *Proceeding of Hydroinformatics'94*, Delft, the Netherlands, pp193-200.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading, MA.

Goldberg, D. E., Korb, B., & Deb, K. (1989). "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Systems, 3*, 493-530.

Goldberg, D. E., Deb, K., Kargupta, H., & Harik G. (1993). "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms," *IlliGAL Report No. 93004*, Illinois Genetic Algorithms Laboratory, University of Illinios at Urbana-Champaign, Urbana, IL 61801, USA.

Walski, T.M. (2000) "Model Calibration Data: The Good, The Bad and The Useless," *J AWWA,* 92(1), p. 94*.*

Walski, T. M. (2001) "Understanding the adjustments for water distribution system model calibration." *Journal of Indian Water Works Association*, April-June, 2001, pp151-157.

Walski, T.M., Chase, D.V. and Savic, D.A. (2001) *Water Distribution Modeling,* Haestad Methods, Inc.

Wang Q.J. (1991), The Genetic Algorithm and its Application to Conceptual Rainfall-Runoff Models, *Water Resources Research*, Vol.27, No.9, pp2467-2482.

Wu Z.Y. (1994), *Automatic Model Calibration by Simulating Evolution*, M.Sc. Thesis*,* H.H. 191, International Institute for Infrastructure, Hydraulic and Environmental Engineering, Delft, the Netherlands.

Wu Z.Y. & Larsen C.L. (1996). "Verification of hydrological and hydrodynamic models calibrated by genetic algorithms." *Proc. of the 2nd International Conference on Water Resources & Environmental Research*, Vol. 2, Kyoto, Japan, pp175-182.

Wu, Z. Y. and Simpson A. R. (2001) "Competent Genetic Algorithm Optimization of Water Distribution Systems." *J. of Computing in Civil Engineering*, ASCE, Vol 15, No. 2, pp89-101.