A Framework for Evolutionary Optimization Applications in Water Distribution Systems

Submitted by Mark Stephen Morley, to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Engineering, March 2008.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

..... (signature)

Abstract

The application of optimization to Water Distribution Systems encompasses the use of computer-based techniques to problems of many different areas of system design, maintenance and operational management. As well as laying out the configuration of new WDS networks, optimization is commonly needed to assist in the rehabilitation or reinforcement of existing network infrastructure in which alternative scenarios driven by investment constraints and hydraulic performance are used to demonstrate a cost-benefit relationship between different network intervention strategies. Moreover, the ongoing operation of a WDS is also subject to optimization, particularly with respect to the minimization of energy costs associated with pumping and storage and the calibration of hydraulic network models to match observed field data.

Increasingly, Evolutionary Optimization techniques, of which Genetic Algorithms are the best-known examples, are applied to aid practitioners in these facets of design, management and operation of water distribution networks as part of Decision Support Systems (DSS). Evolutionary Optimization employs processes akin to those of natural selection and "survival of the fittest" to manipulate a population of individual solutions, which, over time, "evolve" towards optimal solutions. Such algorithms are characterized, however, by large numbers of function evaluations. This, coupled with the computational complexity associated with the hydraulic simulation of water networks incurs significant computational overheads, can limit the applicability and scalability of this technology in this domain.

Accordingly, this thesis presents a methodology for applying Genetic Algorithms to Water Distribution Systems. A number of new procedures are presented for improving the performance of such algorithms when applied to complex engineering problems. These techniques approach the problem of minimising the impact of the inherent computational complexity of these problems from a number of angles. A novel genetic representation is presented which combines the algorithmic simplicity of the classical binary string of the Genetic Algorithm with the performance advantages inherent in an integer-based representation. Further algorithmic improvements are demonstrated with an intelligent mutation operator that "learns" which genes have the greatest impact on the quality of a solution and concentrates the mutation operations on those genes. A technique for

implementing caching of solutions – recalling the results for solutions that have already been calculated - is demonstrated to reduce runtimes for Genetic Algorithms where applied to problems with significant computation complexity in their evaluation functions. A novel reformulation of the Genetic Algorithm for implementing robust stochastic optimizations is presented which employs the caching technology developed to produce an multiple-objective optimization methodology that demonstrates dramatically improved quality of solutions for given runtime of the algorithm.

These extensions to the Genetic Algorithm techniques are coupled with a supporting software library that represents a standardized modelling architecture for the representation of connected networks. This library gives rise to a system for distributing the computational load of hydraulic simulations across a network of computers. This methodology is established to provide a viable, scalable technique for accelerating evolutionary optimization applications.

Keywords

Evolutionary Optimization, Genetic Algorithms, Hydroinformatics, Caching, Multiple-Objective Optimization, Distributed Computing.

Acknowledgements

I am deeply grateful to my family for their enduring support over the period that I have been working towards this thesis. Their consistently positive attitude has ensured that this thesis has finally come to fruition.

In addition, I would like to thank my supervisor, Professor Dragan Savić, not least for his forbearance and perseverance in the face of persistent obduracy against completing this thesis, but also for (twice) rescuing me from the wilds of the commercial sector and introducing me to the academic environment at the Centre for Water Systems in Exeter.

Foremost amongst my (erstwhile) colleagues I would like to express my appreciation to Mr. Roger Atkinson and, latterly, Dr. Carla Tricarico for attempting to instil in me the belief that this thesis could be completed in a meaningful fashion – and that doing so was worthwhile. I would also like to thank Mr. Josef Bicik, Drs. Zoran Kapelan, Ed Keedwell, Francesco di Pierro and Darko Joksimović for their help and support along with all my other colleagues and friends that I have met over *so* many years at the Centre.

Finally, to my two sons, Alexander and Christopher, to whom this thesis is dedicated: without them, it would surely have been completed much sooner ☺

Mark S Morley
Exeter
March 2008

Table of Contents

List of Tables

List of Figures

Glossary

Definitions

Allele — The *alleles* of a gene are the set of values that this gene can take. The most straightforward example taken from biology is that of the gene that determines eye colour. The alleles of this gene are the different colours (Brown, Blue...etc...)

Application Programming Interface — A specification for the public interface to a software library to be used by developers.

Chromosome — Many workers in the field of evolutionary algorithms use the term chromosome instead of *organism* above. As will be shown, it is convenient to maintain a distinction between the two (as well as maintaining the biological analogue) and to preserve the *chromosome* moniker to describe a group of *genes* that are related in some fashion. Thus, a *chromosome* is some sub-division of an *organism's genome*.

Common Object Model — A technology developed by Microsoft for the implementation of a componentized software architecture featuring a standardized API for the introspection of methods and members. Microsoft Windows specific. Largely supplanted by the .NET technology but still underpinning many Microsoft Products (e.g. Office).

Distributed Common Object Model. — As *COM* but with additional functionality to allow the instantiation of objects remotely, across a network.

Dynamic Link Library — A mechanism for dynamically linking software functions into an application. Promotes componentization through code reuse – applications can be composed from smaller building blocks. Unlike COM, however, DLLs do not have a standard mechanism for implementing object classes – which has

	resulted in compiler vendors implementing their own, incompatible techniques for doing so.
Elitism	Applied to generational GAs, elitism allows the most fit individuals from a source population to transfer directly to the destination population without undergoing recombination – ensuring that they are preserved from generation to generation.
Fitness	The measure by which individual organisms are compared to each other to judge their relative suitability for the optimization problem at hand. For single objective algorithms this is a single value, often combined with a *penalty function* to penalise constraint violation.
Gene	Genes are the fundamental unit of genetic algorithms. Each gene represents a specific attribute that is encoded within the genome at a specific location known as a *locus*. This attribute normally represents a decision variable to be considered in the optimization but can also convey other information specific to an *organism*.
Generational GA	A type of Genetic Algorithm used for both single and multiple objective optimization. Following selection and recombination, child organisms are inserted into a *new* population rather than replaced into the existing population as with a Steady-State GA. The selection and recombination process continues until the new population reaches the nominal size of the previous generation. The new population then forms the basis for selection for the next generation.
Genome	The total genetic representation of an organism is described as its genome. In mammalian biology, a genome is ordinarily sub-divided into chromosomes.
Genotype	The representation of the information contained within the genome in its native form. For example, a decision variable

	containing the value 5 may represent a pipe diameter of 80mm. The genotypic value of the decision variable is 5.
Locus	The locus of a *gene* is the position it occurs in a *chromosome*. In Genetic Algorithms, the locus is usually fixed for a given gene. However, in certain types of GA with variable-length or *heterozygous chromosomes* the locus can vary.
Network Calibration	An optimization problem to calibrate a Water Distribution System hydraulic model to match observed field data – commonly by varying pipe friction factors or diameters.
Network Design	An optimization problem to layout new pipes for a Water Distribution System.
Network Rehabilitation/Reinforcement	An optimization problem that determines intervention strategies in a WDS for rehabilitating existing pipes or reinforcing the network through pipe duplication.
Organism	This is the representation of an individual in the *population*. It is a term not normally associated with genetic algorithms and is a by-product of the underlying object-oriented library described here. Conventionally, the term "chromosome" has been the preferred term for an individual in GAs. However, because this use of chromosome is significantly at variance with the biological analogue – and the necessity for the object-oriented library to have an appropriate naming strategy for classes – it was decided to break with convention and to use "*organism*" as the fundamental unit of a *population* instead. Because each *organism* always has exactly one *genome* and that *genome* can only belong to a single *organism* they can be considered concomitant.
Penalty Function	A function commonly used in single objective optimization to represent constraint violation by an individual organism – normally used in combination with a *fitness* value to give an overall relative measure of fitness for a solution.

Phenotype	The representation of the information contained within the genome in its appliedform. For example, a decision variable containing the value 5 may represent a pipe diameter of 80mm. The genotypic value of the decision variable is 80 when translated into the domain of the solution.
Population	A collection of organisms – the pool of genetic material operated on by a Genetic Algorithm.
Recombination	The derivation of child solutions from their parents – ordinarily as a result of crossover and mutation.
Steady-state GA	A type of single-objective Genetic Algorithm in which a pair of solutions are selected from a population, recombined to form two children and those children inserted, according to some rule, back into the original population. c.f. Generational GA.

List of Abbreviations

ACS	Ant Colony Simulation
ADSL	Asymmetric Digital Subscriber Line
AMS	Asset Management System
ANSI	American National Standards Institute
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CA	Cellular Automata
COM	Common Object Model.
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CWS	Centre for Water Systems, University of Exeter
DCOM	Distributed Common Object Model
DLL	Dynamic Link Library
DMA	District Metered Area
DSS	Decision Support System
DTD	Document Type Definition (XML)
EPS	Extended Period Simulation
FLV	Float Valve
GA	Genetic Algorithm
GAP	Generalized Assignment Problem
GIS	Geographic Information System
IP	Internet Protocol
ISO	International Standards Organisation
LAN	Local Area Network
LP	Linear Programming

LSB	Least Significant Bit (binary number)
MGA	Messy Genetic Algorithm
MOGA	Multiple Objective Genetic Algorithm
MSB	Most Significant Bit (binary number)
MTV	Motorised Throttle Valve
NLP	Non-Linear Programming
NRGA	Non-Repeating Genetic Algorithm
NRV	Non-Return Valve
NSGA	Non-dominated Sorted Genetic Algorithm
NYT	New York Tunnels (benchmark problem)
OLE	Object Linking & Embedding
OOTEN	Object-Oriented Toolkit for EPANET
OSGB	Ordnance Survey of Great Britain
PBV	Pressure Break Valve
PDD	Pressure-Driven Demand
PDF	Probability Density Function
PRV	Pressure Reducing Valve
PSG	Piedemonte San Germano (benchmark problem)
PSO	Particle Swarm Optimization
PSV	Pressure Sustaining Valve
RCV	Remote Control Valve
RDBMS	Relational Database Management Software
rNSGA-II	Robust NSGA-II
SA	Simulated Annealing
SDSS	Spatial Decision Support System
SFLA	Shuffled Frog Leaping Algorithm

SGML	Standard Generalized Markup
SMGA	Structured, Messy Genetic Algorithm
SMP	Symmetric Multi-Processing
SOGA	Single Objective Genetic Algorithm
STL	Standard Template Library
SWMM	Storm Water Management Model
TCP	Transmission Control Protocol
THV	Throttle Valve
UDP	User Datagram Protocol
UML	Unified Modelling Language
WAN	Wide Area Network
WAP	Wireless Access Point
WDS	Water Distribution Systems
XML	eXtensible Markup Language
XOR	eXclusive OR *(logical operation)*