Journal of Hydrology 259 (2002) 89–104

Journal of Hydrology

www.elsevier.com/locate/jhydrol

Identification of an urban fractured-rock aquifer dynamics using an evolutionary self-organizing modelling

Yoon-Seok Hong*, Michael R. Rosen[1]

Wairakei Research Centre, Institute of Geological and Nuclear Sciences, Private Bag 2000, Taupo, New Zealand

Received 22 May 2001; revised 11 October 2001; accepted 12 November 2001

Abstract

An urban fractured-rock aquifer system, where disposal of storm water is via 'soak holes' drilled directly into the top of fractured-rock basalt, has a highly dynamic nature where theories or knowledge to generate the model are still incomplete and insufficient. Therefore, formulating an accurate mechanistic model, usually based on *first principles* (physical and chemical laws, mass balance, and diffusion and transport, etc.), requires time- and money-consuming tasks.

Instead of a human developing the mechanistic-based model, this paper presents an approach to automatic model evolution in genetic programming (GP) to model dynamic behaviour of groundwater level fluctuations affected by storm water infiltration. This GP evolves mathematical models automatically that have an understandable structure using function tree representation by methods of natural selection ('survival of the fittest') through genetic operators (reproduction, crossover, and mutation).

The simulation results have shown that GP is not only capable of predicting the groundwater level fluctuation due to storm water infiltration but also provides insight into the dynamic behaviour of a partially known urban fractured-rock aquifer system by allowing knowledge extraction of the evolved models. Our results show that GP can work as a cost-effective modelling tool, enabling us to create prototype models quickly and inexpensively and assists us in developing accurate models in less time, even if we have limited experience and incomplete knowledge for an urban fractured-rock aquifer system affected by storm water infiltration.

Keywords: Urban fractured-rock aquifer; Storm water infiltration; Groundwater level fluctuation; Evolutionarily self-organising modelling; Genetic programming

1. Introduction

Urban storm water disposal via 'soak holes' drilled directly into the top of fractured-rock basalt has occurred in the Mt Eden area of Auckland, New Zealand and for at least the past 60 years. This method brings some benefits: (1) the decrease of surface water contamination in urban streams, and (2) an increase in recharge to the shallow aquifer. However, storm water infiltration on a fractured-rock aquifer can harm groundwater resources in two ways: (1) by changing natural groundwater flow patterns due to increased volume of storm water; and (2) by elevating pollutant concentrations and loadings.

Storm water moving through the numerous soak holes may contain or mobilise high levels of contaminants, such as sediment, suspended solids, nutrients,

* Corresponding author. Tel.: +64-7-374-8211; fax: +64-7-374-8199.

E-mail addresses: t.hong@gns.cri.nz (Y.-S. Hong), mrosen@usgs.gov (M.R. Rosen).

[1] Present address: US Geological Survey, 333 W. Nye Lane, Carson City, NV 89706-0866, USA. Tel.: +1-775-887-7683; fax: +1-775-887-7629.

PII: S0022-1694(01)00587-X

heavy metals, pathogens, toxins, and oxygen-demanding substances. Individually and combined, these pollutants can reduce groundwater quality and threaten beneficial uses as well as impact the ecology of surface water at discharge areas.

Until recently few studies have been done to determine the effect of this disposal method on a given aquifer system (Rosen et al., 1999; Rosen et al., 2000a,b; Hong and Rosen, 2001). Hong and Rosen (2001) have applied an artificial neural network-based pattern analysis to analyse the effect of the storm water infiltration on the groundwater quality, and to determine the response of the groundwater quality variables due to the storm water infiltration in an urban fractured-rock aquifer.

In order to better understand the impacts of storm water infiltration on the groundwater level, predictive groundwater models will be necessary for sustainable groundwater management. Thus, this study aims to develop a predictive model to identify the effect of the storm water infiltration on groundwater level fluctuation.

The fractured-rock aquifer system chosen for this study has many complex and poorly known phenomena. Theories or knowledge to generate the dynamic model that is able to describe the dynamic groundwater level fluctuation due to storm water infiltration are still incomplete and insufficient to construct fully physical-based models. Therefore, formulating an accurate physical-based model, usually expressed in the form of differential equations, requires time- and money consuming tasks. Moreover, the resulting model still has large sources of uncertainty such as closure assumptions, unmodelled phenomena, empirical formulas, and uncertainty in the model input.

As an alternative to the physical-based modelling approach, a new approach, which is called *genetic programming* introduced by Koza (1992) as a self-organising modelling tool, is implemented in this work for understanding the dynamic of groundwater level fluctuation affected by storm water infiltration.

Genetic programming (GP), a branch of the well known field of evolutionary computation, belongs to the class of *artificial intelligence* (AI) computation algorithms. GP is exactly what the name implies: a technique to evolve computer models automatically by methods of natural selection (survival of the fittest). There is a special form of GP, called *symbolic regression* (Koza, 1992), where the induced models are restricted to mathematical functions. The purpose of symbolic regression is to develop mathematical models that fit the input–output data to satisfy the complex problem.

GP has the advantages that no a priori modelling assumption has to be made. Moreover, this technique can discriminate between relevant and irrelevant system inputs, yielding parsimonious model structures that accurately represent system characteristics (McKay et al., 1996) and provide us with a descriptive solution. Due to its advantages, GP has successfully been used for engineering problems with good results such as process modelling and control (Bettenhausen and Marenbach, 1995; Marenbach et al., 1997; McKay et al., 1996; McKay et al., 1997; Willis et al., 1997; Hong, 2001a,b), robot control (Banzhaf et al., 1997), hydrological modelling (Babovic and Abbott, 1997; Keijzer and Babovic, 1999; Whigham and Craper, 1999), medical application (Gray et al., 1996), and applications for financial system (Oussaidene et al., 1996).

Firstly, this paper presents an overview of GP that includes the basic theory of GP, and a discussion of its power as a new modelling tool. Secondly, the practical implementation technique of GP for general modelling work is described in detail. Real data taken from monitoring bores in a fractured-rock aquifer are used to demonstrate how GP can automatically evolve models with relatively simple, understandable structures. The resulting models evolved by GP are used to understand the effect of the storm water infiltration on the groundwater level fluctuation. This shows the feasibility of using GP as an intelligent self-organising modelling tool for an aquifer system.

2. Genetic programming

2.1. Basics of genetic programming

Evolutionary algorithms (EA) are stochastic search methods that mimic the metaphor of natural biological evolution. A variety of EA have been proposed. The major ones are: *Genetic Algorithms* (Holland, 1975; Goldberg, 1989), *Evolutionary Programming* (Fogel, 1994), *Evolutionary Strategies* (Rechenberg, 1973), and *Genetic Programming* (Koza, 1992). Each of

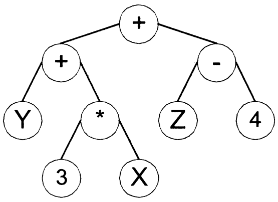

Fig. 1. An example of a function tree used in GP.

these constitutes a different approach. However, they all are inspired by the same principles of natural evolution and share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and reproduction.

GP, which has been introduced by Koza (1992) as a method for genetically breeding populations of mathematical models (function trees), is an extension of the genetic algorithms (GAs). Their main difference is that individuals (models) in GAs are represented by fixed-length strings (usually binary) whereas individuals in GP, which constitute a population, are symbol string codes for mathematical models. These mathematical models are coded as function tree expressions and thus, GP evolves function trees to solve a specific problem.

An example of a GP tree is shown in Fig. 1. The binary arity functions, ' $*$ ', ' $+$ ', ' $-$ ', each have two sub-trees. The sub-tree on the right containing ' $-$ ', 'Z' and 4, represents the arithmetic expression 'Z-4'. The tree as a whole represents $f(X, Y, Z) = Y + 3X + Z - 4$.

In Fig. 1, the connection points are called *nodes*. According to the position in the tree, these nodes are classified into: (1) Inner nodes are known as *functions*, Γ_0. These function nodes consume one or more input values and produce a single output value (e.g. $-$, $*$, $\sqrt{}$, etc.). These provide the internal cells in expression trees. (2) Nodes at the end of points of trees (leaf nodes) are called *terminal*, τ. Terminal nodes represent external inputs, constants, and zero augment functions.

For any particular problem, it is necessary to specify the list of functions and terminals that will be used to create the *function trees*. The solution space to be searched is constrained by the choice of *function set* and *terminal set*, together known as the *primitive set*.

In GP, an *individual* or *chromosome* is an element of the set of all possible combinations of functions that can be composed recursively from a set Γ_0 of basic function and a set τ of terminals. GP works on a population of individuals (mathematical models) applying the principle of survival of the fittest to produce a better and better model for achieving a solution. At each generation, new sets of models, called *offspring*, are created by the process of selecting models according to their level of fitness in the problem domain and breeding them together using genetic operators (reproduction, crossover, and mutation). These offspring (new models) then form the basis for the next generation. This process leads to the evolution of populations of models to produce better and better model to a solution.

Six essential components need to be designed in applying the GP for model induction:

2.1.1. Generating the initial population

In this work, the 'grow' method (Koza, 1992) is used to create random individuals for the initial generation. In this method creating random trees is easily done by a recursive process: For the root node select a random primitive from the combined set of functions and terminals. If a terminal is selected the process ends there. Otherwise, for each of the functions arguments produce a random sub-tree by repeating the same process. In practise, it is necessary to set a limit on the size to which the tree can grow by only selecting from the list of terminals after reaching a certain depth. Also, since trees consisting of only a single terminal are not likely to be very interesting they are usually excluded by forcing the choice of primitive at the root of the tree to be a function.

2.1.2. Assignment of fitness

A crucial first step for evolving any model under the GP is to design a fitness function that determines how well a model is able to solve the problem. In symbolic regression of GP, the fitness is a numeric value assigned to each member of the population based on the error between the model inducted by GP and the actual data. The performance of each model inducted is tested against a set of fitness cases. Based on the fitness, models of the population are selected for reproduction.

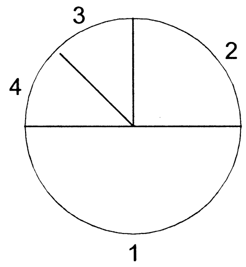

Fig. 2. Roulette wheel selection: individual 1 will be selected for reproduction with a probability of 50%.

2.1.3. Parent selection

Once this fitness measure is constructed, it is used to select models. This is done in a probabilistic manner, such that even the least fit model stands a small, but finite chance of making it through to the next generation. The process of selection ensures that models of higher fitness are more likely to be chosen for reproduction than those of lower fitness. The selection of parent models is based on the principle of survival of the fittest, which governs the extent to which a model can influence future generation. There are several ways of selecting models: fitness proportionate selection, roulette wheel selection, stochastic universal sampling, truncation selection, etc. Roulette wheel selection has been used in this work. Roulette wheel selection is a special case of fitness proportionate selection. Models are chosen for reproduction with probabilities directly proportional to their fitness. For example, if there is a population of four models with the following fitness:

Model	Fitness
1	0.8
2	0.4
3	0.2
4	0.2

then model 1 has a probability of 50% to be selected for reproduction each time a model is selected (each time the wheel is spun). In order to find the probability for each model one has to calculate the sum of qualities of all models and divide the quality of each model by this sum. In this example for model 1, this would be 0.8 divided by $(0.8 + 0.4 + 0.2 + 0.2 = 1.6)$ equals 0.5 (Fig. 2). Thus, those models of greater fitness are expected to receive more spaces in the new population than are those of lesser fitness and expected number of spaces filled by a model is given by the example above.

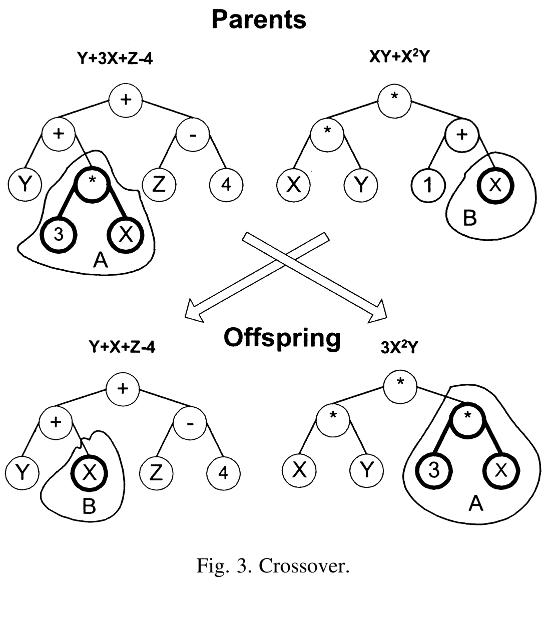

Fig. 3. Crossover.

2.1.4. Creating subsequent generations

Once an individual (a model) has been selected from the current population, three genetic operators (crossover, mutation, and direct reproduction) may be applied. The choice of each operator is probabilistic, with crossover probability P_c, mutation probability P_m, and reproduction probability $P_r(= 1 - P_c - P_m)$.

2.1.4.1. Reproduction. A single model is copied unaltered to the new population.

2.1.4.2. Crossover. The function of the crossover operator is to generate new or offspring models from two parent models by combining information extracted from the parents. This operator tries to combine parts of two models in order to create a superior model. In GP, this is done by selecting a random sub-tree from within each of the parent models, then swapping them over. Fig. 3 shows two function trees in the parent models before and after the crossover operation. This small difference between parents and offspring models shown in Fig. 3 is vital for the GP and allows selection pressure to drive the evolution of the population.

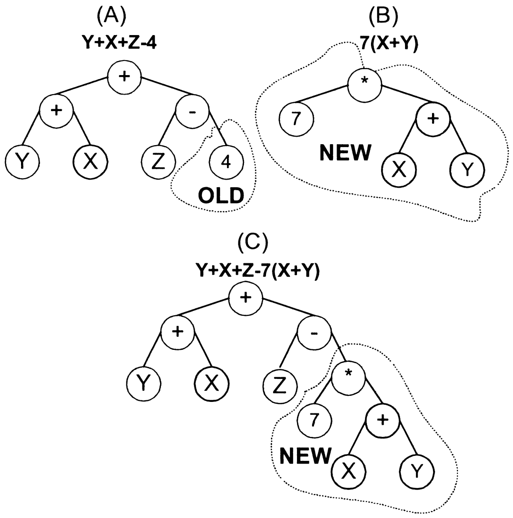

Fig. 4. Mutation genetic operation.

2.1.4.3. Mutation. Even though crossover comes up with many new offspring it does not introduce any new information into the population and the population tends to become more and more homogeneous as one begins to dominate. A mutation feature is often introduced to guard against premature convergence (to a non-optimal solution). In GP, a single model is chosen and a random sub-tree in it is replaced with a new, freshly generated sub-tree, then that model is placed in the new population.

In Fig. 4, the function tree (A) chosen is now mutated. The sub-tree OLD is removed and replaced by the sub-tree NEW, which has been randomly generated, to produce the final function tree (C).

2.1.5. Reduction

After a population of new models has been generated through genetic operators, the population contains individuals from two generations—parents and offspring. During the reduction phase, a decision must be made which models advance to the next generation and which are discarded. In this work, the concept of 'generation gap', which is defined as how many percent of the old generation are replaced by the new generation, is used. In this case, the generation gap is 0.9, which means just 10% of the old generation survives.

2.1.6. Parameter optimisation of models evolved

During the evolution of models, one of the most important issues is the optimisation of the parameters included in the models evolved. Generally, the models generated by GP may include a number of internal model parameters, thus the optimisation process of these model parameters is required to prevent the possibility of removing a potentially good model due to poor model parameters. In an iterative genetic loop (Fig. 5), each model generated by GP is adapted to the observed data by optimising its internal model parameters using the non-linear gradient descent algorithm. The optimised models then replace the unoptimised model in the population. After this step, a fitness value describing the model performance of a model is evaluated. A similar approach is found in the works of McKay et al. (1997), Marenbach et al. (1997), and Greeff and Aidrich (1998).

The basic mechanics of GP for a specific problem that required finding a mathematical model are based on a repetitive computational process and can be summarised as:

1. *Initialisation*: Generate an initial population of N models (individuals) randomly. Generation $K = 0$
2. *Genetic Loop*: Repeat until termination criterion is met (maximum generation K)
   (a) Execute each model in the population
   (b) Evaluate fitness of each model of current population
   (c) Generate a new population N_{K+1} by reproduction, crossover, and mutation of models from current population N_K
      (i) Select three genetic operators (P_r, P_c, P_m) probabilistically
      (ii) Perform three genetic operators to generate new models
      (iii) Repeat steps (i)–(ii) to generate a total of $M(= 0.9 \times N)$ new models
      (iv) $(N - M)$ fittest models from the parent population are added to M new models to create whole new models of population for the next generation equal in number to the previous population, rather than replacing the entire parent population with new models
   (d) Go to next generation: $K + 1$

Fig. 5 shows a flowchart of these steps for the GP paradigm.

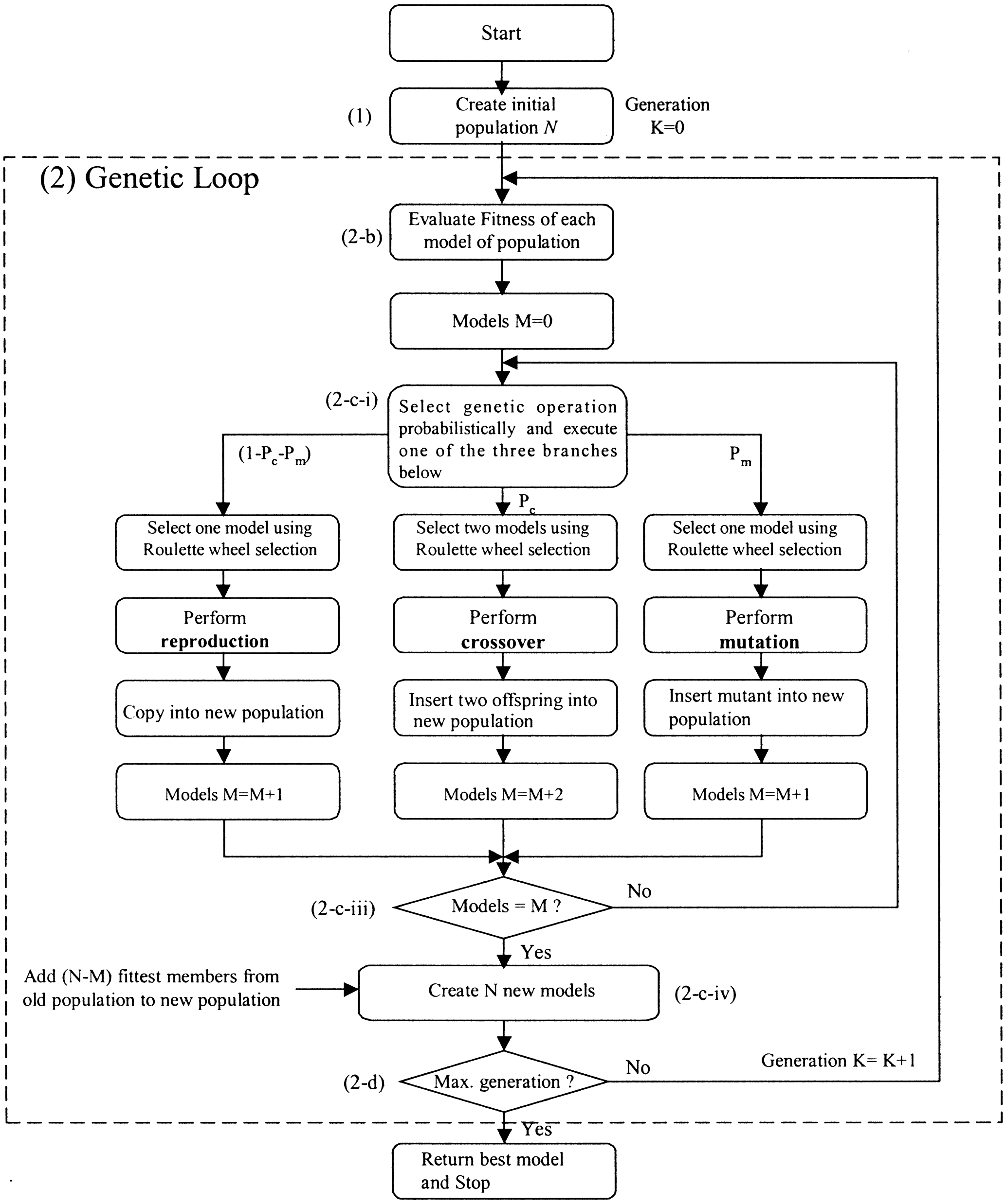

Fig. 5. Flowchart of the genetic programming paradigm.

3. Study area

3.1. Mt Eden aquifer system

The study area is in Mt Eden, which is an inner suburb of Auckland, New Zealand (Fig. 6). Auckland is New Zealand's largest city with a population of over one million people in the greater Auckland area. The geology of the shallow, unconfined groundwater aquifer system is comprised of fractured-basalt, scoria and tuff, from small volcanic cones (Mt Eden, Mt Albert, Three Kings) that were active about 20,000

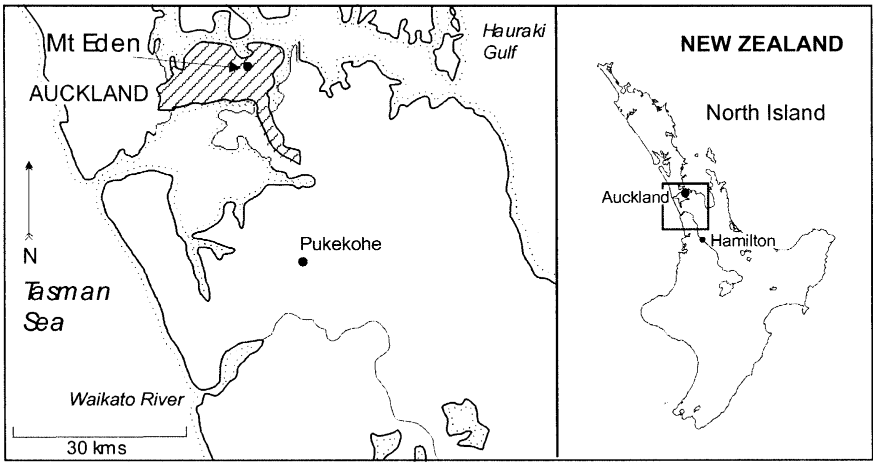

Fig. 6. Location of the Mt Eden aquifer in the Auckland metropolitan area.

years ago. Sandstone and siltstone sediments of the Miocene Waitemata Group underlie the basalt. Waitemata Group sediments have relatively low permeability and act as a barrier to groundwater flow (Russell and Rodgers, 1977). Groundwater flows through the fractured basalt in a shallow, unconfined aquifer system that is channelled through topographic lows in the Mt Eden area. The aquifer system has two separate arms that meet at the Western Springs outlet (Fig. 7). Land use in the catchment is mostly residential housing, but significant pockets of industry also occur. The area includes some of the most heavily travelled roads in New Zealand (Viljevac, 1998).

Disposal of storm water in the Mt Eden area of

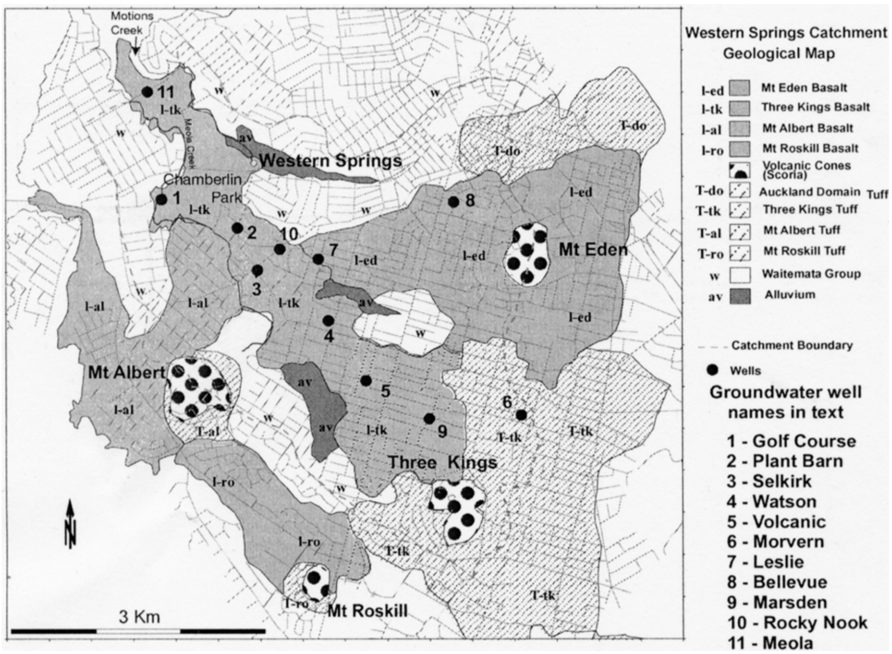

Fig. 7. Geologic map of the Western Springs catchment and aquifer showing the location of groundwater sampling sites.

Table 1
Information on sampling bores (RWL = relative water level)

	Watson Ave.	Volcanic Rd.	Bellevue Res.
Bore diameter (mm)	100	50	50
RL depth of well	15	11	31
RWL	28.18	26.41	34.2
Aquifer thickness (m)	38.3	24	34.3
Land use type	Commercial	Commercial	Residential
Storm water and rainfall drainage type	Storm water directly disposed into ground mainly through the soak holes	Storm water directly disposed into ground mainly through the soak holes	Storm water directly disposed into ground mainly through the soak holes

Auckland, New Zealand is via 'soak holes' drilled directly into the top of fractured-rock basalt. These soak holes collect storm water and sediment runoff from city streets throughout Mt Eden. There are hundreds of these soak holes in the Mt Eden area, many of which are up to 20 m deep, which provide direct access of storm water infiltration below the water table. Storm water disposal via soakage pits has occurred in Auckland for at least the past 60 years, but until recently few studies have been done to determine the effect of this disposal method on the aquifer (Rosen et al., 1999).

The groundwater in this area has a relatively high through-flow and is used for small business water supplies and irrigation water for industrial uses such as nurseries and golf courses in the region. Pump tests conducted by the Auckland Regional Council indicates that hydraulic conductivity ranges from 6 to 126 m/d, with an average of approximately 58 m/d. On average, 2500 m^3/d is contributed by rainfall within the Three Kings crater (Thompson, 1998) and Viljevac and Smaill (1999) have calculated a recharge of 68,800 m^3/d for the whole aquifer based on the present recharge including storm water soakage. A calculation of pre-urbanisation recharge rates indicates that recharge to the aquifer has doubled because of the storm water soak holes (Viljevac and Smaill, 1999).

3.2. Data

The three monitoring sites that were chosen for this work are the Watson Ave., Volcanic Rd., and Bellevue Res. sites in Fig. 7. Groundwater level fluctuation in the three monitoring wells was monitored using automatic data loggers equipped with pressure transducers. All groundwater level data were recorded on a 1-h time interval for the period 16/09/1998–16/11/1998. Rainfall records at 1-h intervals were available at a National Institute of Water and Atmospheric (NIWA) monitoring site, which is located near the Watson Ave. site. Characteristics of the groundwater monitoring sites are described in Table 1.

4. Application and results

4.1. Implementation aspects of GP

The data set consists of 891 hourly measured input–output data points. The data was split into two sets: (1) a training set including 60% of the data, and (2) a test set including the remaining 40%. The training set was used to construct a model. The remaining 40% of the data were used to test the constructed model in order to show how well a model generalizes or predicts unseen data not used during the training phase.

A crucial first step for evolving any model under the GP is to design a fitness function which determines how well an evolved model is able to solve the problem. In symbolic regression of GP, the fitness is based on the error between the model inducted by GP and the actual data. The performance of each function inducted is tested against a set of fitness cases.

In our case, the fitness is a numeric value assigned to each member of the population base on the error between the function expressed by the actual and predicted solutions. In this paper, the fitness

Table 2
GP parameters for all simulations (GWL(k), GWL($k-1$), GWL($k-2$), GWL($k-3$), and GWL($k-4$), represent the groundwater level at time k, $k-1$, $k-2$, $k-3$, and $k-4$, respectively. ra(k), ra($k-1$), ra($k-2$), ra($k-3$),…, and ra($k-10$) are the rainfall at time k, $k-1$, $k-2$, $k-3$,…, and $k-10$, respectively)

Generations	50	Population size	300
Mutation probability, P_m	0.25	Crossover probability, P_c	0.65
Operators	=	Generation gap	0.9
Terminals set τ	GWL(k), GWL($k-1$), GWL($k-2$), GWL($k-3$), GWL($k-4$), ra($k-1$), ra($k-2$), ra($k-3$), ra($k-4$), ra($k-5$), ra($k-6$), ra($k-7$), ra($k-8$), ra($k-9$), ra($k-10$)		
Functional set	+, −, ∗, /, power, sqrt, log, exp		

calculation used is:

$$f_n = \frac{1}{0.01 + \mathrm{RMSE}/B}, \quad \text{where RMSE} \sqrt{\frac{\sum_{n=1}^{N} (y(n) - \widehat{y(n)})^2}{N}} \tag{1}$$

where B is the variance of the output variable over the same data interval, 1–N, $y(n)$ is the actual value, and $\widehat{y(n)}$ is the predicted value by GP inducted model, and N is the number of the data.

Other crucial steps in applying GP to evolve a population for solving a given problem are to: (1) define the set of program primitives, namely the set of terminals τ and the function set Γ_0; (2) define the fitness function f_n, which assigns a fitness to each individual program; (3) define the control parameters (population size N, maximum number of generation K, crossover probability P_c, and mutation probability P_m; (4) define a search termination criterion in terms of quality of solution to be obtained. Table 2 shows common GP parameters used for all the modelling processes in this work.

In this study, the GP program has been developed with a graphical user interface that allows the user to use the program in a sophisticated mouse point and click environment of MATLAB™. Some part of the GP program was originally written in C^{++} and compiled to generate C MEX files, which call C^{++} files directly from MATLAB. This has been done to increase the performance of the simulation for the model evolution process by GP program, reducing the simulation time. Some part of the GP program was also adopted from the work of McKay et al. (1996) and modified. All simulations were done on a SUN engineering workstation.

4.2. Simulation results and discussion

A multiple-input, single-output (MISO) GP model was first developed for the Watson Ave. site. Generally, the groundwater level fluctuation is dynamic due to rainfall events at a certain time instant that is a combined effect of various processes initiated at different moments in the past. Hence, the model of predicating the groundwater fluctuation due to storm water infiltration should be considered as depending not only on the current value of input, but also on past inputs. Thus, the numbers of past rainfall and past groundwater level as time delay embedding are used as inputs to the model.

Initially, a MISO GP model was initialised with 16 input variables given by:

$$\begin{aligned} \widehat{\mathrm{GWL}}(k+1) = f\Big(&\mathrm{GWL}(k), \mathrm{GWL}(k-1), \\ &\mathrm{GWL}(k-2), \mathrm{GWL}(k-3), \mathrm{GWL}(k-4), \\ &\mathrm{ra}(k), \mathrm{ra}(k-1), \mathrm{ra}(k-2), \mathrm{ra}(k-3), \mathrm{ra}(k-4), \\ &\mathrm{ra}(k-5), \mathrm{ra}(k-6), \mathrm{ra}(k-7), \\ &\mathrm{ra}(k-8), \mathrm{ra}(k-9), \mathrm{ra}(k-10)\Big) \end{aligned} \tag{2}$$

where $\widehat{\mathrm{GWL}}(k+1)$ is the predicted groundwater level at time $k+1$. GWL(k), GWL($k-1$), GWL($k-2$), GWL($k-3$), and GWL($k-4$), represent the groundwater level at time k, $k-1$, $k-2$, $k-3$, and $k-4$, respectively. ra(k), ra($k-1$), ra($k-2$), ra($k-3$), and ra($k-4$) are the rainfall at time k, $k-1$, $k-2$, $k-3$, and $k-4$, respectively. $f(\cdot)$ is a model inducted by GP.

For a given data set of Watson Ave., 10 runs were performed with different randomly generated initial models so that each run takes a different genetic loop to evolve the better performing models. For each run, an initial population of 300 candidate models was created using the functional and terminal sets in Table 2. In this work, the terminal set is the set of 16 input variables (e.g. past ra and past GWL) that are expressed in Eq. (2). Through the genetic loop

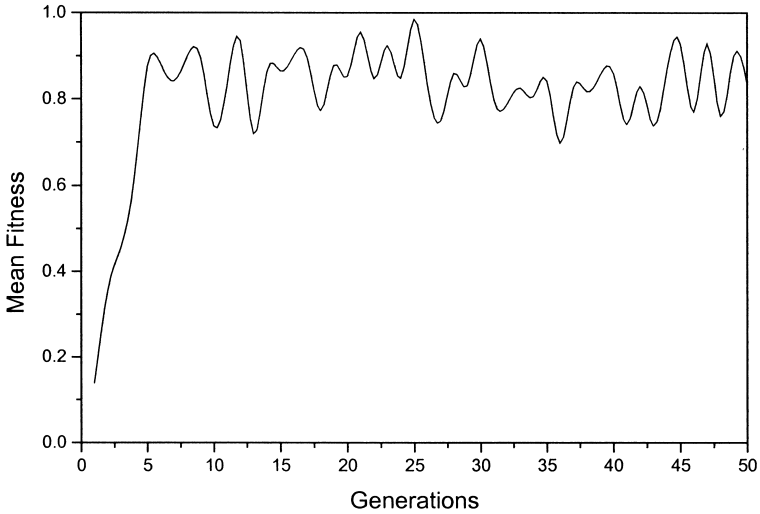

Fig. 8. The progression of the mean fitness in the run 4.

process shown in Fig. 5, the better performing models that have high fitness values tended to be promoted quickly and diffused into subsequent generations to evolve better and better models, while models that were not as good gradually died out. As this happened, the failure rate of new offspring virtually dropped to zero and the grading of models became more refined. This genetic loop process was carried out for 50 generations. Eventually, a range of different better performing models emerged, each capable of predicting the groundwater level with varying degrees of accuracy.

All evolved model's execution outputs were saved in a data file containing the best performing model at the end of each generation and also RMSE with the population mean fitness. Most of the 10 runs generated good prediction models, but the best performing model appeared in run 4. Fig. 8 shows the progression of the mean fitness value, which designates a value averaged over all models evolved of a given generation, over the 50 generations at run 4. The mean fitness value is zero in generation 0. Thus, the graph of mean fitness starts at generation 1 giving a value of 0.173. The graph of mean fitness rises after generation 5 indicating convergence of the search process of better performing models. It reaches the highest value at generation 25. Convergence is accelerated by the combination of the Roulette wheel selection and the low mutation rate. It was found that if the mutation rate P_m was over 0.5, the convergence was slowed and the loss of better performing models was observed. This was due to a negative influence of the high mutation rate on the crossover. The good results are obtained with a mutation rate of 0.25 in this work.

The overall best performing top five models that GP evolved after run 10 are displayed in Table 3. The top five models are sorted by the lowest RMSE. The models as expressed in Table 3 are exactly as evolved by the GP software without any simplification. The

Table 3
Best performing top five models evolved for Watson Ave. site

Rank	Models evolved
1	$[0.00001436] \times \mathrm{ra}(k-2)+[0.00002744] \times \mathrm{ra}(k-1)+\mathrm{GWL}(k)$
2	$[0.00001392] \times \mathrm{ra}(k-3)+[0.00002549] \times \mathrm{ra}(k-1)+\mathrm{GWL}(k)$
3	$[0.00002584] \times \mathrm{ra}(k-1)+[0.00001738] \times \mathrm{ra}(k-4)+\mathrm{GWL}(k)$
4	$[-0.0007843]+\mathrm{sqrt}([0.005603] \times \mathrm{ra}(k-3))+\mathrm{GWL}(k)$
5	$-[99.81]+\{\mathrm{ra}(k-3) \times [-0.00003717] \times (\log(\mathrm{GWL}(k))\} \times \mathrm{GWL}(k))+\mathrm{GWL}(k)$

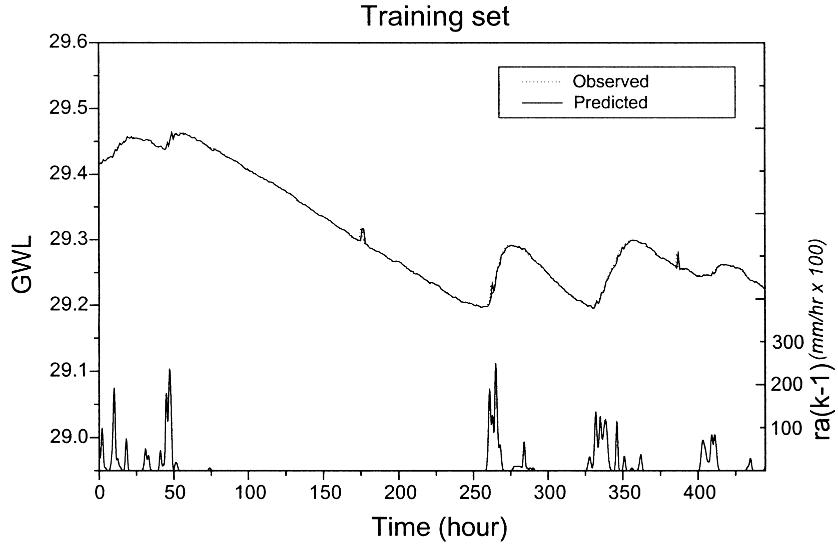

Fig. 9. Observed data versus output of the model evolved by GP at Watson Ave. site.

following model was evolved as the best performing model at generation 25 in the 4th run after 50 generations were completed:

$$\widehat{\mathrm{GWL}}(k+1) = 0.00001436 \times \mathrm{ra}(k-2) + 0.00002744 \times \mathrm{ra}(k-1) + \mathrm{GWL}(k) \quad (3)$$

The model evolved from Eq. (3) has a RMSE on the training set of 0.0031 m and an RMSE on the testing set of 0.00326 m. The model evolved from Eq. (3) also has the 98.54% of R-squared on the training set and the 97.87% of R-squared on the testing set. The RMSE value of 0.00326 m and 97.87% of R-squared for the testing data indicates very satisfactory performance of this model evolved indicating that the accuracy is extremely good. Results obtained from the best performing model on data for the training set are shown in Fig. 9. Results for the testing set are also plotted in Fig. 10. The

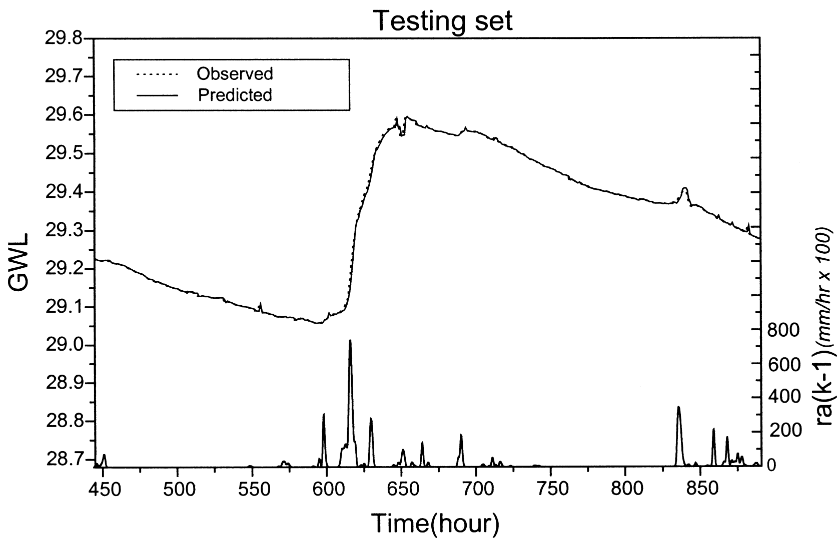

Fig. 10. Observed data versus output of the model evolved by GP at Watson Ave. site.

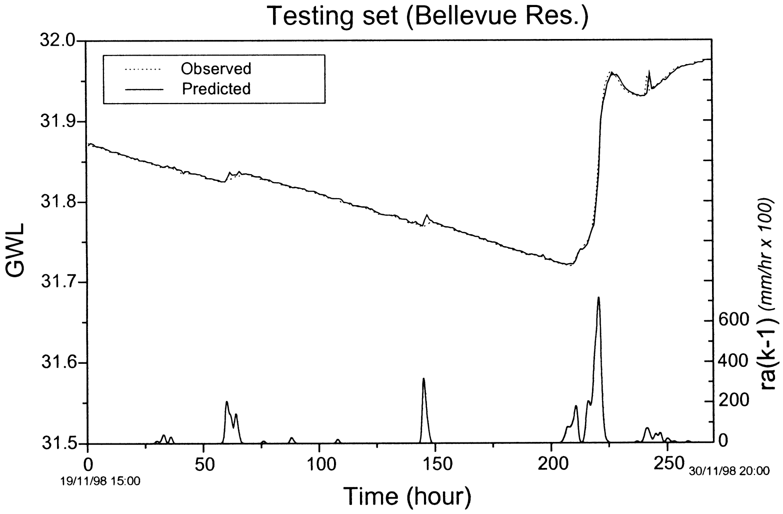

Fig. 11. Comparison of predicted values and observed for Bellevue Res. testing set.

small difference between the RMSE on the training and testing sets shows that the best performing model has a great generalisation capability. The maximum error was 0.0423 m and occurred at one of the transition points of the groundwater level due to a rapid increase in rainfall. Generally, the residual errors with an average of 0.0029 were very small and thus an excellent prediction has been obtained.

The same simulation procedure as that adopted for Watson Ave. site was applied to evolve models for Volcanic Rd. and Bellevue Res. sites. The same GP parameters and terminal sets as shown in Table 2 were used. When our GP algorithm was applied to Bellevue Res. site, the following best performing model at generation 5 in the 6th run after 50 generations was evolved:

$$\widehat{\mathrm{GWL}}(k+1) = 0.00002786 \times \mathrm{ra}(k-2) + 0.0000235 \times \mathrm{ra}(k-1) + \mathrm{GWL}(k) \tag{4}$$

The model evolved in Eq. (4) has an RMSE on the training set of 0.0033 m and an RMSE on the testing set of 0.00346 m. Fig. 11 displays the observed and predicted groundwater level on the testing set fitted by Eq. (4). Table 4 shows the overall best performing top five models that GP evolved after run 10 for the Bellevue Res. site.

For Volcanic Rd. site, the best performing model at generation 6 in the 3rd run after 50 generations was:

$$\widehat{\mathrm{GWL}}(k+1) = 0.00003354 \times \mathrm{ra}(k-2) + 0.00003333 \times \mathrm{ra}(k-1) + \mathrm{GWL}(k) \tag{5}$$

The model evolved in Eq. (5) has an RMSE on the training set of 0.00341 m and an RMSE on the testing set of 0.00354 m. Fig. 12 shows the observed and predicted groundwater level for the testing set using the evolved model of Eq. (5). The overall best performing top five models that GP evolved after run 10 are displayed in Table 5.

Table 4
Top five best performing models evolved for Bellevue Res. site

Rank	Models evolved
1	$[0.00002786] \times \mathrm{ra}(k-2) + [0.0000235] \times \mathrm{ra}(k-1) + \mathrm{GWL}(k)$
2	$[0.00003547] \times \mathrm{ra}(k-3) + [0.00002283] \times \mathrm{ra}(k-1) + \mathrm{GWL}(k)$
3	$[-0.2265] + [0.00002821] \times \mathrm{ra}(k-2) + [0.00003002] \times \mathrm{ra}(k-1) + [1.007] \times \mathrm{GWL}(k)$
4	$[0.00002757] \times \mathrm{ra}(k-2) + [0.00003046] \times \mathrm{ra}(k-1) + [1.006] \times \mathrm{GWL}(k)$
5	$[-0.001139] + [0.00001677] \times \mathrm{ra}(k-3) + [0.00001729] \times \mathrm{ra}(k-2) + [0.0000294] \times \mathrm{ra}(k-1) + \mathrm{GWL}(k)$

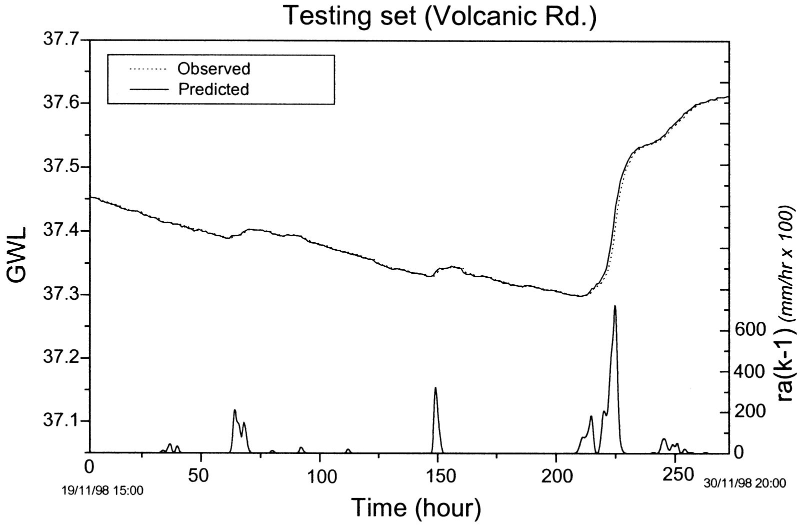

Fig. 12. Comparison of predicted values and observed for Volcanic Rd. testing set.

From Figs. 10–12, the predictions of the models evolved for the three monitoring sites closely follow the observed groundwater level patterns with high levels of accuracy, giving similar RMSE values. It can be seen that the models evolved by GP represent the dynamic characteristics of the Mt Eden groundwater system and serve as a good predictor.

Sensitivity analysis was done to show how changes in the rainfall will affect the groundwater level using the top performing models evolved by GP for three sites. Result of a sensitivity analysis for three sites is shown in Fig. 13. Fig. 13 shows graphically the range of variance of the expected groundwater level due to the variance of the rainfall intensity. The range of 29.31–29.35 m is an expected groundwater level at Watson Ave. site due to the range of 0–9 mm/h of variance of the rainfall intensity. This result shows that an increase of 0–9 mm/h in rainfall intensity would produce approximately 4 cm increase in groundwater level at the Watson site. The sensitivity analysis for Bellevue Res. site (Fig. 13(C)) shows a similar result as that observed for the Watson Ave. site showing that an increase of 8 mm/h in rainfall causes the groundwater level to rise approximately 4 cm at the Bellevue Res. site. For the Volcanic Rd. site, the sensitivity analysis indicates that the groundwater level response due to increase in rainfall is expected to be a little greater than the other two sites, and the change in groundwater level would be 6 cm per 9 mm/h change in rainfall intensity.

Looking at the models evolved in Tables 3–5, one of the most interesting results is that they provide information on which input variables and model structures GP evolved are important to achieve the high levels of accuracy. From Tables 3–5, GWL(k), ra($k-1$), ra($k-2$), and ra($k-3$) among the terminal

Table 5
Top five best performing models evolved for Volcanic Rd. site

Rank	Models evolved
1	$[0.00003354] \times \text{ra}(k-2) + [0.00003333] \times \text{ra}(k-1) + \text{GWL}(k)$
2	$[0.00001959] \times \text{ra}(k-2) + [0.00002438] \times \text{ra}(k-1) + [1.002] \times \text{GWL}(k-1)$
3	$[0.00002151] \times \text{ra}(k-1) + [1.0018] \times \text{GWL}(k) + [7.389\text{e}-008] \times \text{ra}(k-1) \times \text{ra}(k-2)$
4	$[-0.2978] + [0.00001687] \times \text{ra}(k-3) + [7.584\text{e}-006] \times \text{ra}(k-2) + [0.0000248] \times \text{ra}(k-1) + [0.9921] \times \text{GWL}(k)$
5	$[0.00002095] \times \text{ra}(k-3) + [0.00002776] \times \text{ra}(k-1) + \text{GWL}(k)$

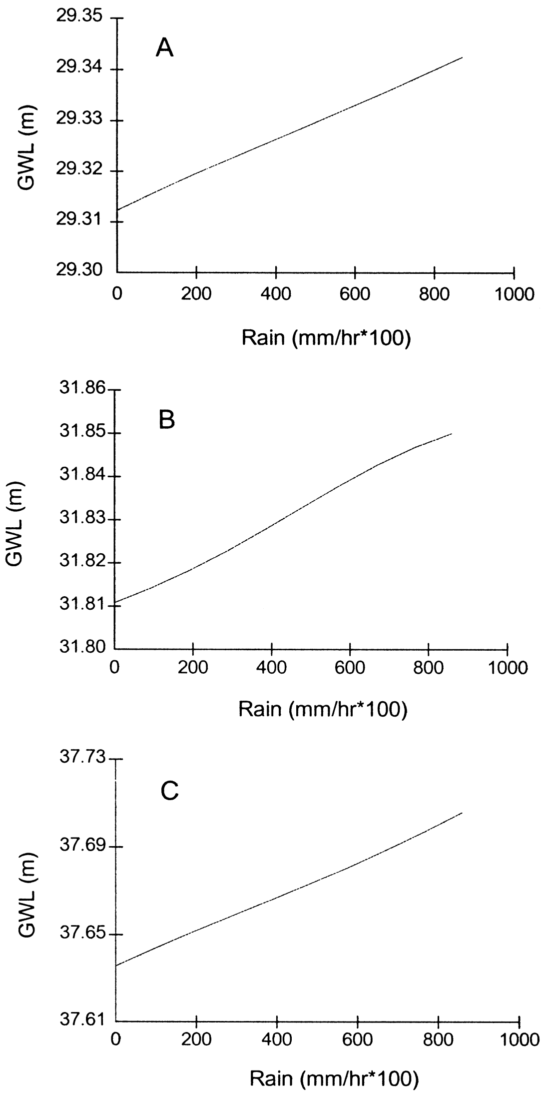

Fig. 13. Sensitivity analysis results: (A) Watson Ave., (B) Bellevue Res., and (C) Volcanic Rd.

set dominate the structure of the better performing models. With information of an understandable model structure, the following knowledge for this urban fractured-rock aquifer system can be extracted: (1) The time delay between a change in the rainfall intensity and the observed response in the groundwater level is 2 h, (2) The time span over which a momentary rainfall change persists in affecting the groundwater level is 2–3 h, and (3) The change in groundwater level would be approximately 4–6 cm per 9 mm/h change in rainfall intensity through sensitivity analysis.

It is also worth noting that an inspection of models evolved from Tables 3–5 reveals that most of models evolved by GP are linearly correlated with rainfall. These models evolved with a degree of linearity in the rainfall term providing the most accurate prediction results, indicating that there is likely to be a linear relationship between groundwater fluctuation and storm water infiltration during rainfall periods.

From Eqs. (3)–(5), GP evolved the best performing model as functions of GWL(k), ra($k-1$), and ra($k-2$) having similar model coefficients for three monitoring sites. This shows that this model structure can be used as a general model for predicting the groundwater level fluctuation with the averaged model coefficients in this aquifer system.

The general mathematical modelling approaches first require a form of specified model structure and then their parameters are then optimised by the optimisation algorithm. From Tables 3–5, GP evolves both the model structures and the model constants simultaneously by its ability to self-organise the system. Thus, an advantage of GP is that in principle, it does not require any form of specified model structure selection in the model parameter estimation process.

Furthermore, the main advantage of the GP for modelling processes is the ability to produce models that build an understandable structure using function tree representation; i.e. a formula. With this information of an understandable structure, the models evolved can be compared to observed knowledge and judged as to whether they make sense. The transparency of the models evolved in Tables 3–5 can allow the activity of what-if scenario simulation for resource management planning.

5. Conclusions

In this paper, our results have demonstrated the applicability of GP to analyse and identify the dynamic behaviour of the groundwater level due to change in rainfall intensity in an urban fractured-rock aquifer.

The fractured-rock aquifer system chosen for this study has a highly dynamic nature where theories or knowledge to generate the model are still incomplete and insufficient. Instead of a human developing the mathematical model, the GP can self-modify, through the genetic loop, a population of function trees in order to finally generate an accurate model that predicts the groundwater level fluctuations due to changes in rainfall intensity.

GP is not only capable of predicting the groundwater level fluctuation due to storm water infiltration but provides insight into the dynamic behaviour of a partially known fractured-rock aquifer system by allowing knowledge extraction of the evolved models. Our results show that GP can work most efficiently where the possible model is unknown and the understanding of the resulting model is important.

The greatest benefit in GP lies in the flexibility of the model induction process combined with the ability to integrate mathematical models and logical structures into a self-organising system. GP can find from simple linear to complex non-linear relationships in input–output data. It is easy to use because it in principle does not require any form of specific pre-defined model and automatically constructs mathematical models directly from the data set. Thus, it is cost-effective, enabling us to create prototype models quickly and inexpensively. GP assists us in developing accurate models in less time, even if we have limited experience and incomplete knowledge. By introducing the knowledge of the human expert and modifying the functional sets, the human expert can guide the GP process to derive more sophisticated models.

Due to great advantages of the GP technique implemented in this work, there is a great deal of potential using it as a general modelling tool, suitable for a variety of other complex process systems such as hydrological and environmental processes.

Acknowledgements

This work was funded by the Foundation for Research Science and Technology, New Zealand and the Auckland Regional Council. We would like to thank Alastair Smaill and Zeljko Viljevac at the Auckland Regional Council for their continued support, advice and encouragement for this work.

References

Babovic, V., Abbott, M.B., 1997. Evolution of equation from hydraulic data: part i—theory. Journal of Hydraulic Research 35 (3), 1–14.

Banzhaf, W., Nordin, P.J., Olmer, M., 1997. Generating adaptive behaviour for a real robot using function regression within genetic programming. Proceedings of the Second Annual Conference on Genetic Programming, Stanford, USA.

Bettenhausen, K.D., Marenbach, P., 1995. Self-organizing modelling of biotechnological batch and fed-batch fermentations. In: Breitenecker, F., Husinsky, I. (Eds.). EUROSIM'95. Elsevier, Vienna, Austria, pp. 445–450.

Fogel, D.B., 1994. An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks 5 (1), 3–14.

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

Gray, H.F., Maxwell, R.J., Martinez-Perez, I., Arus, C., Cerdan, S., 1996. Genetic programming for classification of brain tumours from nuclear magnetic resonance biopsy spectra. In: Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L. (Eds.). Genetic Programming 1996: Proceedings of the First Annual Conference on Genetic Programming. MIT Press, Cambridge, MA.

Greeff, D.J., Aidrich, C., 1998. Empirical modelling of chemical process systems with evolutionary programming. Computers and Chemical Engineering 22, 995–1005.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Michigan.

Hong, Y.-S., 2001a. Evolutionary self-organising modelling and optimisation: an application of a paper coating process. 2001 Joint Conference of Society of Chemical Engineering N.Z. (SCENZ)/Food Engineering Association of N.Z. (FEANZ)/ Engineering Material Group (EMG), Auckland, New Zealand, pp. 77–81.

Hong, Y.-S., 2001b. Evolutionary self-organizing modeling of a municipal wastewater treatment plant. Water Research, in press.

Hong, Y.-S., Rosen, M.R., 2001. Intelligent characterisation and diagnosis of the groundwater quality in an urban fractured-rock aquifer using an artificial neural network. Urban Water 3, 193–204.

Keijzer, M., Babovic, V., 1999. Dimensionally aware genetic programming. Gecco'99: Proceedings of the Genetic and Evolutionary Computation Conference, July 13–17, Orlando, USA.

Koza, J., 1992. Genetic Programming: On the Programming of Computers by Natural Selection. MIT Press, Cambridge, MA.

Marenbach, P., Bettenhausen, K.D., Freyer, S., Nieken, U., Rettenmaier, H., 1997. Data-driven structured modelling of a biotechnological fed-batch fermentation by means of genetic programming. Proceedings of the Institution of Mechanical Engineers Part I 211, 325–332.

McKay, B., Elsey, J., Willis, M.J., Barton, G., 1996. Evolving input–output models of chemical process system using genetic programming. IFAC 96 World Congress, San Francisco, July.

McKay, B., Willis, M., Barton, G.W., 1997. Steady-state modelling of chemical process systems using genetic programming. Computers and Chemical Engineering 21 (9), 981–996.

Oussaidene, M., Chopard, B., Pictet, O., Tomassini, M., 1996.

Parallel genetic programming: an application to trading models evolution. In: Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L. (Eds.). Genetic Programming 1996: Proceedings of the First Annual Conference on Genetic Programming. MIT Press, Cambridge, MA.

Rechenberg, I., 1973. Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart.

Rosen, M.R., Reeves, R., Sheppard, D.S., Roberts, K., Viljevac, Z., Smaill, A., 1999. Effect of storm water infiltration on a fractured-rock aquifer, Mt Eden Auckland. New Zealand Hydrological Society Symposium, Napier 23–26 November, pp. 70.

Rosen, M.R., Sheppard, D.S., Roberts, K., Viljevac, Z., Smaill, A., Reeves, R., 2000a. Effects of storm water infiltration on groundwater quality in a fractured rock aquifer in Auckland. Storm Water Conference and Workshop Proceedings, New Zealand Water and Waste Association, 12–14.

Rosen, M.R., Hong, Y.-S., Sheppard, D., Roberts, K., Viljecvac, Z., Smaill, A., Reeves, R.R., Final report for the IAEA urban aquifers RCA: determining the effects of storm water infiltration on groundwater quality in an urban fractured rock aquifer, Auckland, New Zealand. Institute of Geological and Nuclear Sciences Science Report 2000/20.

Russell, W.J., Rodgers, K.A., 1977. Waters of the Western Springs catchment, Auckland. New Zealand Journal of Marine and Freshwater Research 11, 713–728.

Thompson, I.C., 1998. The three kings groundwater resource. Water and Waste in New Zealand 101, 22–23.

Viljevac, Z., 1998. Western Springs aquifer. Hydrological characteristics and computer model. Masters Thesis, Department of Geology, University of Auckland, pp. 144.

Viljevac, Z., Smaill, A., 1999. The impact of stormwater disposal on groundwater resources, an example of shallow volcanic aquifers in Auckland (New Zealand). Comprehensive Stormwater and Aquatic Ecosystem Management Conference, Auckland, New Zealand.

Whigham, P.A., Craper, P.F., 1999. Time series modelling using genetic programming: an application to rainfall-runoff modelling. Advances in Genetic Programming 3, vol. 5. MIT Press, Cambridge, MA, pp. 89–104.

Willis, M., Hiden, H., Hinchliffe, M., McKay, B., Barton, G.W., 1997. Systems modelling using genetic programming. Computers and Chemical Engineering 21, S1161–S1166.