The Genetic Algorithm and Its Application to Calibrating Conceptual Rainfall-Runoff Models

Q. J. Wang

Department of Engineering Hydrology, University College, Galway, Ireland

The genetic algorithm is a search procedure based on the mechanics of natural selection and natural genetics, which combines an artificial survival of the fittest with genetic operators abstracted from nature. In this paper, a genetic algorithm for function optimization is introduced and applied to calibration of a conceptual rainfall-runoff model for data from a particular catchment. All seven parameters of the model are optimized. The results show that the genetic algorithm can be efficient and robust.

A GENETIC ALGORITHM

The genetic algorithm is a search procedure based on the mechanics of natural selection and natural genetics, which combines an artificial survival of the fittest with genetic operators abstracted from nature [Holland, 1975]. It has been applied to a number of problems including search, optimization and machine learning [De Jong, 1975; Grefenstette, 1985, 1987; Davis, 1987; Goldberg, 1989].

The genetic algorithm differs from other search methods in that it searches among a population of points and works with a coding of the parameter set rather than the parameter values themselves. It also uses probabilistic rather than deterministic transition rules. With the genetic algorithm, a population of m points are chosen initially at random in the search space. The objective function values are calculated at all points and compared. From these m points, two points are selected randomly, giving better points higher chances. The selected two points are subsequently used to generate a new point in a certain random manner with occasionally added random disturbance. This is repeated until m new points are generated. The generated population of points are expected to be more concentrated in the vicinity of optima than the original points. The new population of points, which can again be used to generate another one and so on, yielding points more and more concentrated in the vicinity of the optima.

Given a function \( f(x_1, x_2, \cdots, x_n) \) subject to \( a_i \leq x_i \leq b_i, \ i = 1, 2, \cdots, n \), the aim is to find the set of parameter values which lead to a minimum value of \( f \). The genetic algorithm works with the coding of the parameters.

A method of parameter coding that has been successfully used is binary coding. An \( l \)-bit binary variable is used to represent one parameter \( x_i \). The integer of the decoded variable ranges from 0 to \( 2^l - 1 \) and can be mapped linearly to the parameter range \( [a_i, b_i] \). The parameter range \( \Delta x_i \) is discretized into \( 2^l \) points and the discretization interval is

\[
\Delta x_i = \frac{b_i - a_i}{2^l - 1}
\]

For example, when \( l = 7 \), the mapping is shown in Table 1. Connecting the codings of all parameters forms the coding for each point in the space to be searched, for example,

\[
\begin{array}{ccccccc}
1000010 & 0010100 & \cdots & 0101001 & 1101001 \\
1 & 2 & \cdots & n-1 & n
\end{array}
\]

Note that the search range for each of the parameters must be specified. The search is then made as follows:

1. Locate \( m \) points randomly in the search space (\( m = 100 \) can be used).
2. Find the function value for each point.
3. Rank the points so that their function values are in descending order.
4. Assign a probability value \( p_j \) to each point \( j = 1, 2, \cdots, m \), giving higher probability to points of lower function value. The worst point after ranking is \( j = 1 \), and its probability value \( p_1 \) will be the smallest. The best point is \( j = m \), and its probability value \( p_m \) will be the largest. The probability values for other points are linearly interpolated as

\[
p_j = p_1 + \frac{p_m - p_1}{m-1} (j-1)
\]

The summation of probability values for all points, \( \sum_{j=1}^{m} p_j \), should be equal to unity. The average of probability values for all points is then \( 1/m \). A value of \( C/m \) can be assigned to \( p_m \) so that the probability value for the best point is \( C \) times the average, where \( C > 1 \). The corresponding probability value for the worst point \( p_1 \) is then \((2 - C)/m\). To ensure that all probability values are nonnegative, \( C \) should be less than or equal to 2. \( C = 2.0 \) can be used. Using a smaller value of \( C \) will result in a more robust but slower search.

5. Select two points \( A \) and \( B \) from the \( m \) points at random according to the probability distribution \( p_j, j = 1, 2, \cdots, m \).
6. Select two bit positions, \( k_1 \) and \( k_2 \), along the overall coding of the parameter set at random, giving each bit position the same chance. If \( k_1 > k_2 \), their values are exchanged.
7. Form a new point by taking the values of the bits from \( k_1 \) to \( k_2 - 1 \) of the \( A \) point coding and the values of the bits from \( k_2 \) to the end and from 1 to \( k_1 - 1 \) of the \( B \) point coding.
8. Occasionally change some of the bits of the newly formed point. A bit value 0 will become 1 and vice versa.
TABLE 1. Example of Parameter Coding for \( i = 7 \)

<table>
<thead>
<tr>
<th>Binary Code</th>
<th>Integer Value</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>0</td>
<td>( a_i )</td>
</tr>
<tr>
<td>00000001</td>
<td>1</td>
<td>( a_i + \Delta x_i )</td>
</tr>
<tr>
<td>0000010</td>
<td>2</td>
<td>( a_i + 2\Delta x_i )</td>
</tr>
<tr>
<td>1111110</td>
<td>126</td>
<td>( a_i + 126\Delta x_i )</td>
</tr>
<tr>
<td>1111111</td>
<td>127</td>
<td>( a_i + 127\Delta x_i = b_i )</td>
</tr>
</tbody>
</table>

This occurs to each bit only at a very small probability \( p_m \) (\( p_m = 0.01 \) can be used).

9. Repeat steps 5-8 \( m \) times so that \( m \) new points are produced. The original \( m \) points are then replaced by the new ones, forming a new data base for further search.

10. Repeat steps 2-9. The best point found so far is always recorded. Termination of the search can be made by specifying a total number of function evaluations.

Steps 6 and 7 form the core of the method. Better points have better chances to be chosen to form new points. This is an analogy to the survival of the fittest in the theory of natural selection. The better performing individuals produce more offspring. A new point is formed by taking different blocks of bits from the codings of the two original points. This is an analogy to crossover in the theory of genetics. An offspring takes some of the genes from one parent and some from the other one. Fit parents are likely to produce fit offspring. The combination of selection and reproduction improves the performance level of the population as the process moves on. The occasional change of bit values in step 8 is an analogy to mutation in the theory of genetics. It provides some background variation.

The genetic algorithm introduced here is only one of the many forms of the method. For example the selection probabilities may be directly related to the function values by a simple relationship rather than using the ranking procedures. More information can be found in the works by Holland [1975], De Jong [1975] and Goldberg [1989]. The genetic algorithm introduced is, however, particularly recommended by this author for its simplicity and robustness.

AN EXAMPLE

For demonstration purpose, the genetic algorithm was applied to minimize a simple function

\[
 f = \sum_{i=1}^{10} x_i^2
\]

where \(-2 \leq x_i \leq 10, i = 1, 2, \ldots, 10\). The function has a zero minimum at \( x_i = 0, i = 1, 2, \ldots, 10 \).

The following genetic algorithm operator parameter values were used: \( m = 100, l = 10, C = 2.0 \) and \( p_m = 0.01 \). The use of a 10-bit binary coding to represent each variable means that the range of each variable is discretized into \( 2^{10} = 1024 \) points.

A random number generator was used to produce random numbers distributed uniformly between 0 and 1. The random numbers were subsequently transformed into the required forms whenever randomness was involved. With a different initial "seed," a different series of random numbers was created.

Ten runs were made with different initial seeds resulting in different initial starting populations of points and different operations. Each run consisted of 5000 function evaluations. The minimum function values found in each run are 0.526, 0.428, 0.200, 0.299, 0.140, 0.175, 0.481, 0.228 and 0.365. Their average is 0.328. All of them are close to zero. Figure 1 shows the 10-run average of the best-so-far objective function values versus the number of evaluations. The trend toward improvement can be clearly seen.

The genetic algorithm searches within a population of points. The search is guided by an overall structure, and yet the detail involves many random actions. As a result, the method is both efficient and robust. It is especially useful for some difficult optimization problems where the standard methods fail. The search is also globally oriented, and this is useful for situations where multiple peaks exist in the response surfaces.

The above example function has 10 variables. Its response surface is well defined. The genetic algorithm has been applied to many less well defined response surfaces, and the results are promising [De Jong, 1975].

The final convergence from somewhere near the peak to the very peak is slow with the genetic algorithm. In the above example, none of the runs has reached the zero function optimum value. When high precision is required tuning by one of the standard function optimization methods after an initial search by the genetic algorithm may be necessary.

TABLE 2. Parameter Ranges for Bird Creek Catchment Data

<table>
<thead>
<tr>
<th>W (mm)</th>
<th>B</th>
<th>K_f</th>
<th>X</th>
<th>Y</th>
<th>n</th>
<th>aK (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower bound</td>
<td>50</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Upper bound</td>
<td>400</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>
TABLE 3. Results of Calibration of the Xinanjiang Model for the Bird Creek Catchment Data Using the Genetic Algorithm

<table>
<thead>
<tr>
<th>Run</th>
<th>Wm, mm</th>
<th>B</th>
<th>Ke</th>
<th>X</th>
<th>Y</th>
<th>n</th>
<th>$F^2$, mm²</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>153.0</td>
<td>0.608</td>
<td>0.903</td>
<td>0.110</td>
<td>0.115</td>
<td>5.30</td>
<td>1.57</td>
<td>2009</td>
</tr>
<tr>
<td>2</td>
<td>149.9</td>
<td>0.719</td>
<td>0.869</td>
<td>0.164</td>
<td>0.072</td>
<td>5.05</td>
<td>1.59</td>
<td>2025</td>
</tr>
<tr>
<td>3</td>
<td>126.0</td>
<td>0.949</td>
<td>0.983</td>
<td>0.110</td>
<td>0.096</td>
<td>4.70</td>
<td>1.59</td>
<td>2071</td>
</tr>
<tr>
<td>4</td>
<td>136.9</td>
<td>0.738</td>
<td>0.874</td>
<td>0.188</td>
<td>0.076</td>
<td>5.36</td>
<td>1.57</td>
<td>2035</td>
</tr>
<tr>
<td>5</td>
<td>150.2</td>
<td>0.640</td>
<td>0.910</td>
<td>0.126</td>
<td>0.098</td>
<td>5.45</td>
<td>1.56</td>
<td>2009</td>
</tr>
<tr>
<td>6</td>
<td>143.1</td>
<td>0.670</td>
<td>0.962</td>
<td>0.136</td>
<td>0.109</td>
<td>5.20</td>
<td>1.57</td>
<td>2013</td>
</tr>
<tr>
<td>7</td>
<td>147.8</td>
<td>0.639</td>
<td>0.866</td>
<td>0.152</td>
<td>0.090</td>
<td>4.82</td>
<td>1.58</td>
<td>2019</td>
</tr>
<tr>
<td>8</td>
<td>159.8</td>
<td>0.652</td>
<td>0.870</td>
<td>0.117</td>
<td>0.102</td>
<td>5.25</td>
<td>1.56</td>
<td>2025</td>
</tr>
<tr>
<td>9</td>
<td>135.2</td>
<td>0.691</td>
<td>0.866</td>
<td>0.202</td>
<td>0.063</td>
<td>4.74</td>
<td>1.63</td>
<td>2056</td>
</tr>
<tr>
<td>10</td>
<td>144.8</td>
<td>0.692</td>
<td>0.873</td>
<td>0.138</td>
<td>0.119</td>
<td>5.37</td>
<td>1.61</td>
<td>2023</td>
</tr>
</tbody>
</table>

AN APPLICATION TO CALIBRATING A CONCEPTUAL RAINFALL-RUNOFF MODEL

Conceptual rainfall-runoff models usually consist of a number of parameters. Most of the parameters have to be calibrated by examining the estimated and the measured discharge series. The use of function optimization methods for calibrating rainfall-runoff models has been studied by Libett and O'Donnell [1971], Johnston and Pilgrim [1976], Pickup [1977], Gupta and Sorooshian [1985], Hendrickson et al. [1988] and Brazil [1989]. They found that the standard optimization methods can be easily fooled into declaring convergence far short of the true optima because of high dimensionality and irregularities contained in the response surfaces such as multiple peaks, unsmoothness, discontinuity, elongated ridges, flat plateaus, and so on. The author believes that none of the methods currently used for calibrating conceptual rainfall-runoff models with even a moderate number of parameters is robust and yet efficient in locating or nearly locating the global optima.

The genetic algorithm has some distinctive characteristics as compared with the standard methods. It was thus applied to calibrating the Xinanjiang rainfall-runoff model for the Bird Creek catchment data.

The Xinanjiang rainfall-runoff model is a soil moisture accounting model developed by Zhao and his colleagues [Zhao et al., 1980; Y. L. Zhuang, unpublished manuscript, 1986]. It consists of two components dealing with water balance and routing, respectively.

There are five parameters in the water balance component. Parameters $W_m$ and $B$ describe a storage capacity distribution curve with $W_m$ as the average of soil moisture storage capacity over the catchment and $B$ as an exponent of the curve. Parameter $K_e$ is a coefficient converting open water evaporation into potential evaporation. The storage depth is divided into three layers which are subject to different rates of evaporation. Parameter $X$ is the fraction of the total storage depth taken by the upper layer. Parameter $Y$ is the ratio of the depth of the third layer to the depth of the second layer. Detailed descriptions can be found in the works by Zhao et al. [1980] and Y. L. Zhuang (unpublished manuscript, 1986).

The runoff generated from the water balance component is transformed into discharge by a linear system. The gamma function instantaneous unit hydrograph is used as the impulse response [Nash, 1958]. This introduces two more parameters, $K$ and $n$. If $n$ is integral, the system corresponds exactly to a series of $n$ equal linear reservoirs with storage coefficient $K$. The parameter pair $n$ and $nk$ should, however, be chosen rather than $n$ and $K$, because $n$ is a shape parameter and the product $nk$ is a scale parameter. Expressed in this way, the two parameters are likely to be more independent than would $n$ and $K$, both of which contribute to the scale.

A total of seven parameters are thus contained in the model. All of them are to be calibrated by minimizing the residual variance $F^2$ defined as the sum of squares of differences between computed and observed discharges. The model efficiency is measured by the proportion of the initial variance accounted for by the model $R^2 = (F_0^2 - F^2)/F_0^2$, where the initial variance $F_0$ is defined as the sum of squares of differences of the observed discharges from its mean value.

The Xinanjiang model was applied to the Bird Creek catchment using the genetic algorithm for optimization. The catchment is located near Sperry, Oklahoma, and has an area of 2344 km² with gently rolling terrain and moderately humid climate. Six years' data, starting from October 1, 1955, of daily values of rainfall, pan evaporation and discharge were used for calibration. The parameter ranges used are as given in Table 2.
A total of 60 days was taken as the warm-up period at the end of which it is assumed that the effect of the initial assumed soil moisture condition will be reduced to an insignificant level. The memory length of the pulse response function was taken as 10 days.

The same genetic algorithm operator parameter values were used as in the example. As before, the use of a 10-bit binary coding to represent each parameter means that the range of each parameter is discretized into $2^{10} = 1024$ points.

To examine the ability of the genetic algorithm to optimize the parameters of the Xinanjiang model, 10 runs were made with different initial seeds resulting in different initial starting populations of points and different operations. Each run consisted of 5000 objective function evaluations. Table 3 shows the best points found in each of the 10 runs and their associated $F^2$ and $R^2$ values. Figure 2 shows the 10-run average of the best-so-far objective function values versus the number of evaluations. The trend toward improvement can be clearly seen.

Further tuning was provided by the sequential simplex method [Beveridge and Schechter, 1970]. The results are shown in Table 4. Except for the third and the ninth runs, the $F^2$ values of all other runs converge to or very close to 2007 mm$^2$. The parameter values of these eight runs are less convergent than their $F^2$ values but are still in the same vicinity, and are most likely of one peak. The $F^2$ value 2007 mm$^2$ has not been excelled by further extensive searches, indicating that it is perhaps the minimum value that the model can achieve on this catchment with the given data set.

The third and the ninth runs have slightly higher $F^2$ values and different sets of parameter values as shown in Table 4. Further searches around these two points have not found any improvement, indicating that they are two local optima. The values of parameters $B$ and $K_e$ of the third run and $X$ and $Y$ of the ninth run are notably different from those of other runs. The $F^2$ values of these two runs are, however, only marginally larger than 2007 mm$^2$. When there exist several similar peaks in the response surface, it may be unreasonable to expect any search method to always pick out the same single peak which is only slightly superior to others. In practice, the three peaks here are indistinguishable as far as the $F^2$ and $R^2$ values are concerned.

Indeed, the results shown in Table 3 obtained by applying only the genetic algorithm are practically indistinguishable from the results shown in Table 4 obtained by further tuning using the sequential simplex method. Tuning by the sequential simplex method may, however, be useful if the optimum point is actually located outside the given parameter search space. Thus, one run of the genetic algorithm with further tuning by one of the standard search methods, such as the sequential simplex method used here, can provide an efficient and robust means for calibrating the Xinanjiang rainfall-runoff model.

### Conclusion

A genetic algorithm for function optimization has been introduced. A set of points may be chosen at random initially and the objective function calculated at each point. The genetic algorithm technique is applied to this set of points to generate a new population of points using a partially random selection but yielding a concentration of points in the vicinity of those which have already shown good values of the objective function. The new population of points then forms a new data base for further search. A simple example was chosen to demonstrate the use of the algorithm for function optimization.

The algorithm was then applied to calibration of the Xinanjiang rainfall-runoff model for the Bird Creek catchment data. All of the seven parameters of the model were optimized. Out of 10 runs of optimization, each starting from a different set of randomly selected initial points in the search space and with 5000 objective function evaluations, eight runs were able to locate the global peak. Although the other two runs located two other peaks, the objective values of these two peaks are only marginally higher than, and practically indistinguishable from, the objective function value of the global peak. Therefore, all 10 runs can be regarded as successful. This shows that the genetic algorithm can provide an efficient and robust means for calibrating the Xinanjiang model. The algorithm can also be applied to calibration of other hydrological models.

### References


Q. J. Wang, Department of Civil Engineering, University of Dublin, Earlsfort Terrace, Dublin 2, Ireland.

(Received July 31, 1990; revised April 26, 1991; accepted May 7, 1991.)