6th International Conference on Hydroinformatics - Liong, Phoon & Babovic (eds)
© 2004 World Scientific Publishing Company, ISBN 981-238-787-0

COMPARISON BETWEEN A SIMPLE GA AND AN ANT SYSTEM FOR THE CALIBRATION OF A RAINFALL-RUNOFF MODEL

RAFAEL E. OLARTE
Engineering Faculty, Javeriana University, Cra. 7 No. 40 -62
Bogotá, Colombia

NELSON OBREGÓN
Engineering Faculty, Javeriana University, Cra. 7 No. 40 -62
Bogotá, Colombia

It is widely mentioned in hydroinformatics that hydrologic models serve as a decision support tool for the planning and management of water resources. One key step for the adequate implementation of these models, is their calibration, that is, finding the right parameters that correspond to the natural system that is to be studied. In this work, two meta-heuristic optimization methods were used for the calibration of the Thomas rainfall-runoff model applied to several catchments located near Bogotá, Colombia. The aim is to compare a simple genetic algorithm (with elitism) versus an ant system (adapted for continuous functions) in terms of their efficacy and efficiency. Knowing that for each method, different (and more effective) versions exist and that, for each version, different input values (algorithm's parameters) may be used, the aim is not to finally conclude which is the best, but to give a preliminary step and suggestions of what could be a rigorous comparison when applying these two methods to the Thomas model.

INTRODUCTION

Hydroinformatics is concerned, among other matters, with the implementation of decision support tools for the planning and management of water resources. This concern may involve for example, the application of hydrologic models that are correctly calibrated. Methods for calibrating hydrologic models are abundant, but making the right choice may save time and money. Some of the most attractive methods are the so called meta-heuristic optimization algorithms. One could define them as a mixture of heuristics and rigorous mathematics (for an illustrative definition, see Falkenauer [1]). Genetic algorithms (or GAs), tabu search, simulated annealing and ant colony optimization algorithms (or ACO algorithms) classify in this category. They are very robust and easy to implement, but still, much information must be gathered in order to know how they can be more suited for each specific problem.

The aim of this document is twofold. First, it pretends to serve as a reference when choosing between an ACO algorithm (the *MAX-MIN* Ant System) and a simple GA, in the calibration of the Thomas rainfall-runoff model. For this matter, three catchments were analyzed and a comparison between these two algorithms will be exposed, revealing details such as their parsimony, their efficacy, their efficiency and the values of the parameters that presented the best results. Second, this document is also proposing a way to adapt the *MAX-MIN* Ant System to continuous function optimization problems. The *MAX-MIN* Ant System (or *MAX-MIN* AS), as any other classical ACO algorithm, was designed to optimize combinatorial problems (such as the traveler salesman problem) and not continuous functions (such as calibrating a rainfall-runoff model). There are already some new ACO algorithms designed for continuous function optimization such as the API [2], the CACO [3] and the CIAC [4], but they distance much from the original concept of an ACO algorithm. This proposal is also encouraged by a more specific problem exposed by Bowden, Dandy and Maier [5] during the last International Conference on Hydroinformatics.

THE THOMAS MODEL AND THE AREA OF ANALYSIS

The Thomas rainfall-runoff model [6] allows to predict the average flow Q_i of a month *i,* as well as the groundwater discharge Sg_i and the soil moisture storage Sw_{i-1} by knowing the precipitation P_i, the evapotranspiration EP_i, the previous groundwater discharge Sg_{i-1}, the previous soil moisture storage Sw_{i-1}, and four parameters *a*, *b*, *c*, *d*.

The calibration of the Thomas model is a continuous function optimization problem. The idea is to minimize to zero, an objective function $F(\theta)$ where θ is the unknown vector composed of real numbers. For the present study, three catchments were analyzed. Their characteristics are shown in Table 1.

Table 1. Characteristics of the three catchments that were analyzed.

CATCHMENT	AREA [km^2]	Q_i-min [mm/month]	Q_i-min [mm/month]
Curibital	56,4	10.25	132.59
Susagua	31,0	8.11	44.48
Bogotá (Partial)	85,4	3.13	50.44
Localization: near Bogotá, Colombia.			

A period of twelve months was analyzed. Values were taken by averaging those corresponding from January 1997 to December 1999. The objective function chosen was the root mean squared error (see Eq. 1).

$$F(\theta)=\sqrt{\frac{1}{n}\cdot\sum_{month=1}^{12}\left[Q_i - Q_{simulated\,i}(\theta)\right]^2} \qquad (1)$$

Not only were the parameters *a*, *b*, *c* and *d* considered as the unknown values, but also the initial groundwater discharge Sg_0 and the initial soil moisture storage Sw_0. Therefore, any solution takes the form of θ = [a b c d Sg_0 Sw_0].

Unlike the calibration problem exposed by Bowden, Dandy and Maier [2], where each unknown value θ_k belonged to the same search space $S = \{0, \ldots, 100\}$, here each θ_k has a different search space as shown in Table 2.

Table 2. Search spaces for each θ_k that is to be found to calibrate the model.

Parameter or Initial Condition θ_k	*a*	*b*	*c*	*d*	Sw_0	Sg_0
Lower Bound	0,800	10	0,001	0,001	0	0
Upper Bound	1,000	350	0,900	1,000	500	500
Precision	0,001	1	0,001	0,001	1	1

CALIBRATING WITH THE *MAX-MIN* ANT SYSTEM

The *MAX-MIN* Ant System (or *MM*AS) was introduced by Stützle and Hoos [8]. It can be easily explained with the classical combinatorial problem of the traveler salesman (or TSP). It formulates this following question: given a number *n* of cities that are interconnected by *n* x *n* segments (*i, j*), in what order should the traveler salesman visit all the cities without repeating any, so that he makes the shortest possible tour?

Inspired by the way Argentine ants find the shortest route to get to a food source, the *MM*AS proposes the algorithm shown in Figure 3. In the TSP, the objective function $F(\theta)$ is simply the sum of the lengths of all the segments (*ij*) that compose a tour. θ is a vector in the form of $[\theta_1, \ldots, \theta_k, \ldots, \theta_n]$ where $\theta_k \in [1, \ldots, n]$ and indicates the order in which the cities should be visited. According to Figure 3, the algorithm involves these three mathematical expressions[†]:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + 1/F(\theta) \quad (2)$$

$$p_{ij} = \tau_{ij} / \sum_j \tau_{ij} \quad (3)$$

$$\tau^{max} = \left(\frac{1}{1-\rho}\right) \cdot \left(\frac{1}{F(\theta)}\right) \quad (4)$$

Figure 3 also reveals that five values, called parameters, must be set up before running the algorithm. These are: *m*, ρ, *globUpdate*, *# of iterations* and τ_{min}. They will be fully explained in Tables 3 and 4.

[†] It would not be mentioned here the variable η_{ij} which refers to heuristic information that may be used for solving the TSP. Stützle and Hoos [8] proposed an equation for τ_{min}, but for this study, it would be considered as another parameter. Bowden, Dandy and Maier [4] showed better results following this idea.

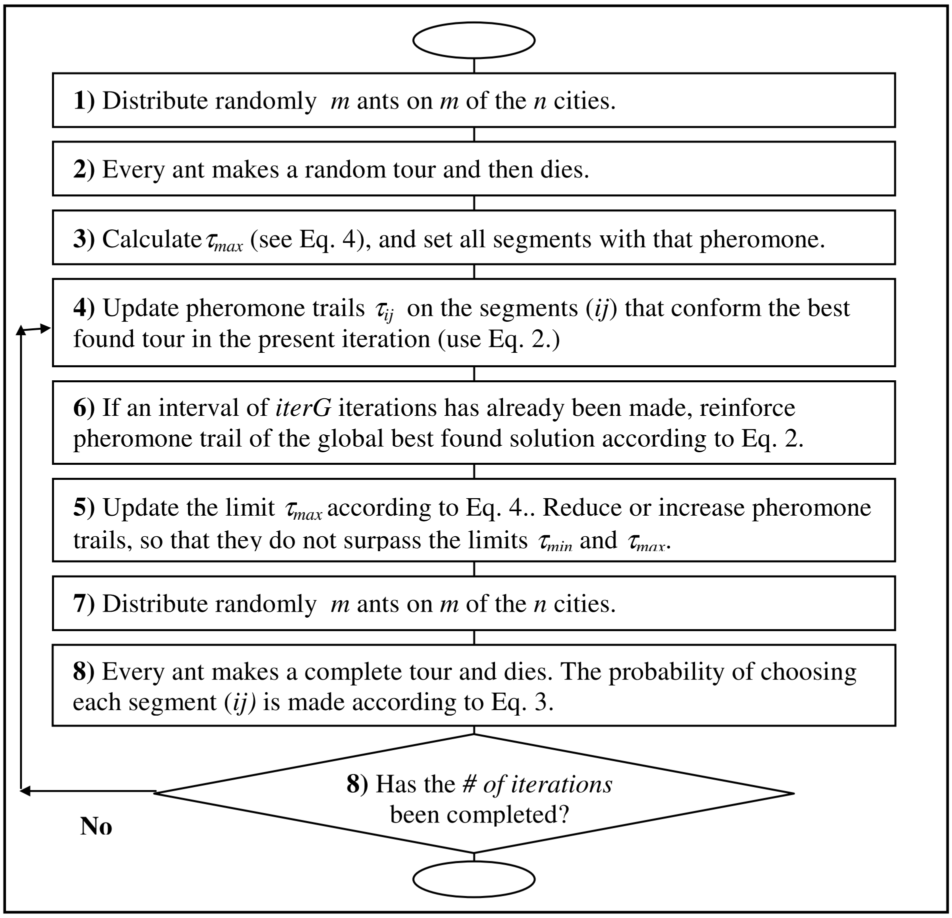

Figure 3. *MAX-MIN* Ant System when applied to the TSP

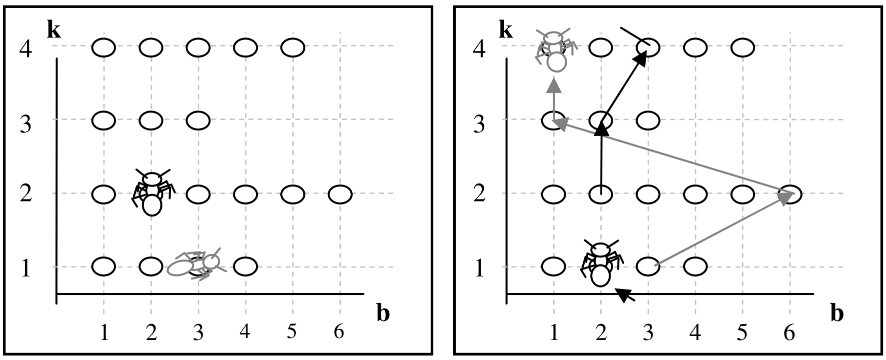

Figure 4. The way ants make their tours during each iteration in the calibration problem.

The *MAX-MIN* AS adapted to the calibration of the Thomas model

Figure 4 proposes how to apply the *MM*AS to this problem. The *k*-axis represents each of the θ_k values that have to be found. For each θ_k, there are b_k possible values.

One computational aspect is worth to be mentioned here. Since updating the pheromones τ_{ij} one by one for all the possible segments is very time consuming (there are in total $(b_1{\cdot}b_2)+(b_2{\cdot}b_3)+\cdots(b_{k-1}{\cdot}b_k)$ segments), it was decided to store in one single variable, the pheromone trail τ_{ij} of all the non visited segments (*ij*). Consequently, the algorithm is going to spend more time in every next iteration. This is proved in Figure 9.

Following Table 2, the values of b that were used are: b = [201 341 900 1000 501 501].

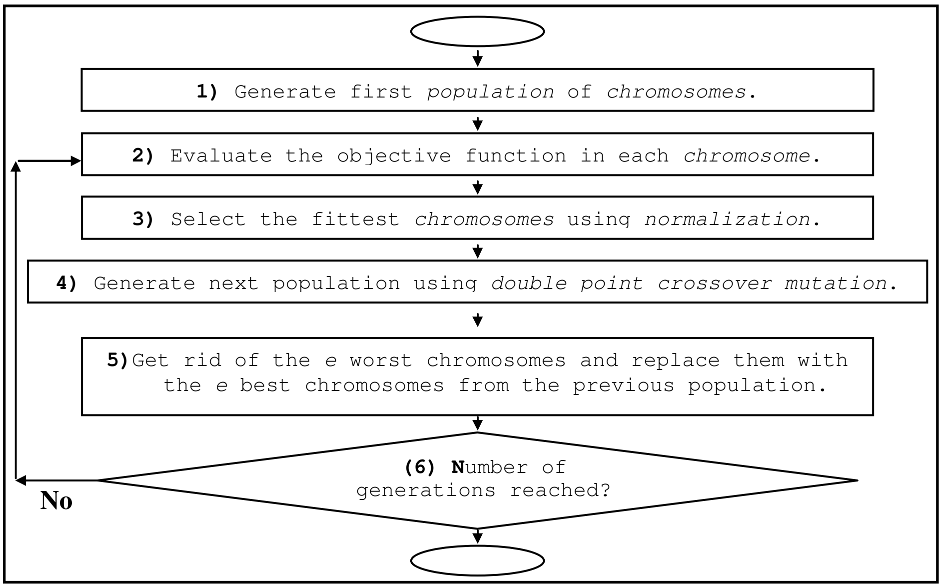

Figure 5. Simple Genetic Algorithm with some modifications for the calibration problem.

CALIBRATING WITH A SIMPLE GA

It was applied for the present study, the algorithm shown in Figure 5. Here, the term *iteration* is usually replaced by *generation*. Although there are some differences, it is based on the simple GA exposed in Goldberg [9]. Since the objective in the calibration problem is to make a minimization, it was needed to *normalize* the values obtained from the objective function *F(x)* according to Eq. 5. Then the typical *roulette wheel selection* is made using the values of *g(x)*.

$$g(x) = \frac{g_{\max}}{F_{\max-generation} - F_{\min-generation}} \cdot \left[F(x) - F_{\min-generation}\right] \quad (5)$$

The chosen value for g_{max} affects the probability in which the fittest chromosomes are selected. Therefore, when using *normalization*, g_{max} becomes one more parameter of the algorithm.

Following Table 2, the number of genes in each θ_k chromosome are [8 9 10 10 9 9].

COMPARISON BETWEEN THE TWO ALGORITHMS

To begin with, an analysis of all the parameters of the algorithms are going to be analyzed. Their definitions are mentioned in Table 3.

Table 3. Parameters required for the GA and the *MM*AS that were implemented.

Simple GA with elitism		*MM*AS for continuous functions	
PopSize	Size of the population of individuals.	m	Size of the population of ants.
maxGen	# of generations to be carried out.	*maxIter*	# of iterations to be carried out.
e	# of individuals that are preserved from generation to generation.	*globUpdate*	Interval in which the pheromone trails τ_{ij} of the global best found solution are reinforced.
p_m	Mutation probability.	ρ	Persistance.
p_c	Crossover probability.	τ_{min}	Minimum amount of pheromone τ_{ij} that a segment (ij) can store.

Making a sensitivity analysis for these algorithms involves the realization of extensive matrices, due to the strong interaction among them. For brevity, it will be mentioned here that, for the *MM*AS, the only sensible parameters were ρ and τ_{min}.. *maxIter* does not help much when increasing it above 100. More results about these analyses are mentioned in Table 4.

If the *MM*AS may seem more parsimonious, on the other hand, it does not show efficiency. Figure 6 shows the time spent when using a PC with a processor of 1000 MHz and 256 of RAM.

Finally, the GA showed to be more effective (see Table 5). The efficacy was calculated based on the following scale: 100% if the $F(\theta)$ equals to 0, and 0% if $F(\theta)$ is equal to the difference $Qmax - Qmin$ obtained from Table 2. The results shown in Table 5 correspond to averages performance after using 5 different seeds (values required for the generation of pseudo-random numbers).

Table 4. Comparison between the different parameters of the two algorithms.

Simple GA with elitism	*MM* AS	How are they alike?	How do they differ?
PopSize	*m*	As this number increases, solutions are searched more intensively at each iteration.	*PopSize* not only allows searching for more solutions, but gives chance for more exploitation. *PopSize* helps increasing the efficacy more than *m* does, but it also increases the time expenditure.
maxGen	*maxIter*	As ths number increases, more chance is given for finding the optimum.	*maxIter* showed to be less sensible than *maxGen*. With values above 100, *maxIter* did not increase the efficacy..
e	*globUpdate*	As *e* increases or *globUpdate* decreases (depending on the algorithm), more exploitation is given to the best found solutions.	Applying elitism to the GA shows a real increase in exploitation, whilst *globUpdate* didn't show to be very useful.
p_m	ρ, τ_{min}	Increasing p_m, increasing ρ, or decreasing τ_{min}, increases exploration of new solutions.	A very high value of p_m, impedes the normal functioning of a GA. Results showed that τ_{min} was not very sensible. p_m under 0,5 is recommended.
p_c	This parameter increases exploitation of building blocks.		

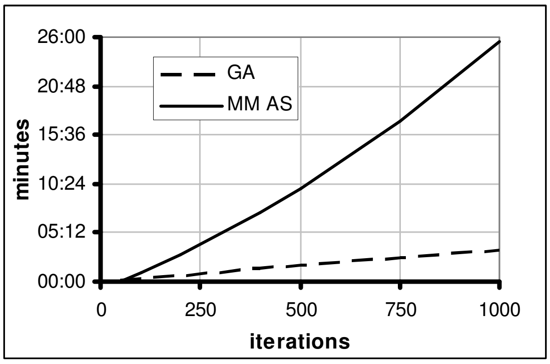

Figure 6. Time spent in each of the algorithms.

Table 5. Best found results.

Simple GA with elitism				*MM*AS for continuous functions			
Catchment	Curibital	Susagua	Bogotá	Catchment	Curibital	Susagua	Bogotá
PopSize	50	75	75	m	150	150	150
maxGen	1000	2000	750	*maxIter*	500	500	500
e	1	1	1	*globUpdate*	10	10	10
p_m	0,01	0,01	0,01	ρ	0,95	0,95	0,95
p_c	0,8	0,7	0,6	τ_{min}	0,1	0,1	0,1
Efficacy	**95,67%**	**91,62%**	**92,18%**	**Efficacy**	**95,42%**	**91,14%**	**91,98%**

CONCLUSIONS

Two important results can be taken from this paper. First, some parameter values can now be recommended when applying the two proposed algorithms to the calibration of the Thomas rainfall-runoff model. Second, the proposed ACO algorithm did not show the expected efficacy and efficiency. This gives more arguments for those who are studying ACO algorithms that solve continuous function optimization problems in a much different way than the classical versions of ant systems.

REFERENCES

[1] Falkenauer E., "*Genetic algorithms and grouping problems",* John Wiley & Sons, (1998). pp 8-11.

[2] Monmarché N., Venturini G. and Slimane M., "*On how Pachycondyla apicalis ants suggest a new search algorithm", Future Generation Computer Systems*, Vol. 16 (2000), pp 937-946.

[3] Bilchev G. and Parmee I.C., "*The ant colony metaphor for searching continuous design spaces", Lecture Notes in Computer Science*, Vol. 993 (1995), pp 25-39.

[4] Bowden G.J., Dandy G.C. and Maier H.R., "Ant colony optimization of a general regression neural network for forecasting water quality", *Proc. 5th International Conference on Hydroinformatics*, Cardiff, Vol. 1, (2002), pp 693-698.

[5] Dréo J. and Siarry P.., *"A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions"* [online], [2002], [*cited* 15-July-2003] *Available at:* http://nojhan.free.fr/IMG/pdf/ants2002.dreo.pdf

[6] Serrano S., "Hydrology for engineers, geologists and environmental professionals*",* HydroScience, (1997).

[7] Dorigo M. and Di Caro G., "*The ant colony metaheuristic" in* Corne D., Dorigo M. and Glover F. "*New ideas in optimization",* Mc-GrawHill, (1999). p 9.

[8] Stützle T. and Hoos H., "*Max-Min ant system", Future Generation Computer Systems*, Vol. 16 (2000), pp 889-914.

[9] Goldberg D., "Genetic algorithms in search, optimization, and machine learning*",* Addison-Wesley, (1989).