Advances in Water Resources 24 (2001) 827–841

Advances in Water Resources

www.elsevier.com/locate/advwatres

Estimating unsaturated soil hydraulic parameters using ant colony optimization

K.C. Abbaspour [a,*], R. Schulin [b], M.Th. van Genuchten [c]

[a] *Swiss Federal Institute for Environmental Science and Technology (EAWAG), Ueberlandstrasse 133, P.O. Box 611, 8600 Dübendorf, Switzerland*
[b] *Swiss Federal Institute of Technology, Institute of Terrestrial Ecology, ETH Zurich, Switzerland*
[c] *George E. Brown, Jr. Salinity Laboratory, USDA, ARS, 450 West Big Spring Road, Riverside, CA 92507, USA*

Received 1 February 2001; received in revised form 25 March 2001; accepted 12 April 2001

Abstract

Although models are now routinely used for addressing environmental problems, both in research and management applications, the problem of obtaining the required parameters remains a major challenge. An attractive procedure for obtaining model parameters in recent years has been through inverse modeling. This approach involves obtaining easily measurable variables (model output), and using this information to estimate a set of unknown model parameters. Inverse procedures usually require optimization of an objective function. In this study we emulate the behavior of a colony of ants to achieve this optimization. The method uses the fact that ants are capable of finding the shortest path from a food source to their nest by depositing a trail of pheromone during their walk. Results obtained with the ant colony parameter optimization method are very promising; in eight different applications we were able to estimate the 'true' parameters to within a few percent. One such study is reported in this paper plus an application to estimating hydraulic parameters in a lysimeter experiment. Despite the encouraging results obtained thus far, further improvements could still be made in the parameterization of the ant colony optimization for application to estimation of unsaturated flow and transport parameters. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Inverse modeling; Unsaturated soil; Hydraulic parameter; Ant colony optimization; Parameter estimation

1. Introduction

During the past several decades, considerable advances have been made in the mathematical description of water flow and solute transport processes in variably saturated soils and groundwater systems. A large number of models of different degrees of complexity and dimensionality are now available for predicting subsurface flow and transport [21,23,31]. Still, effective application of such models to practical field problems suffers from the lack of knowledge of model parameters and their uncertainty. Model parameters are often difficult, if not impossible, to measure directly because of time, money, instrumentation, scale, and conceptual constraints. To deal with the issue of model parameter identification, inverse modeling (IM) has provided an attractive alternative method to direct measurement.

IM constitutes estimation of input parameters from readily measurable model output variables. Input parameters are obtained by minimizing an objective function describing the difference between measured and simulated data. Most optimization algorithms used for parameter estimation in variably saturated flow and transport studies [10,13,22,25,32] are gradient-type methods that have the disadvantage of being very sensitive to the initial guesses of the parameters, while also being prone to convergence to local minima [14,19]. In this paper we propose a global search algorithm for estimating model parameters based on optimization by a colony of ants.

Social insects, such as ants, bees, termites and wasps, often exhibit a collective problem-solving ability [5]. In particular, ants are capable of finding the shortest path between their nest and a food source [4]. This optimization capability is apparently achieved by secreting recruitment pheromone on their path when the ants are in a transport mode [20].

The process by which ants are able to establish the shorter of two paths between two points A and B is

[*] Corresponding author. Tel.: +41-01-823-5359; fax: +41-01-823-5375.
E-mail address: abbaspour@eawag.ch (K.C. Abbaspour).

0309-1708/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved.
PII: S0309-1708(01)00018-5

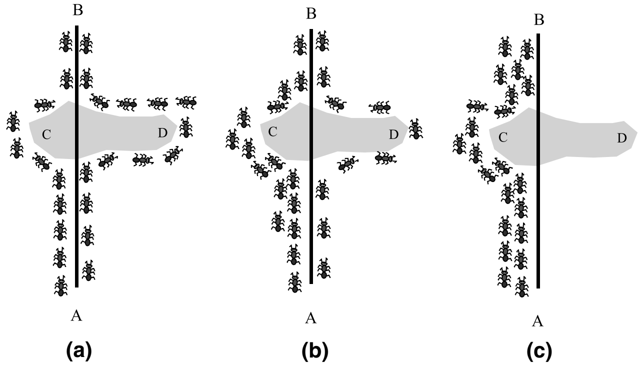

Fig. 1. Example of real ants finding the shortest path: (a) initially ants choose paths ACB and ADB with equal probability; (b) path ACB, being shorter, receives more pheromone with time; (c) more pheromone attracts more ants; hence, eventually all ants will follow path ACB.

illustrated in Fig. 1. Initially, ants have an equal chance of choosing between paths ACB and ADB. However, after some time path ACB, by virtue of being shorter, receives more pheromone from ants going in both directions (ACB, BCA). The shorter path with extra pheromone tends to attract more ants, which in turn will lead to more pheromone accumulation. The collective behavior that emerges is a form of autocatalytic (positive feedback) process, which will soon result in all ants following the shorter path.

Inspired by this ant-based optimization principle, researchers have proposed several algorithms for solution of classical combinational optimization problems such as the traveling salesman problem (TSP) [15], the quadratic assignment problem [11], or the job-shop scheduling problem [7], all of which were reasonably successful. This paper presents an adaptation and application of the ant colony optimization (ACO) technique to the inverse estimation of unknown parameters in unsaturated flow models.

2. The ant colonization search algorithm

Since, to the best of our knowledge, ACO has not been used before in the area of soil and hydrological sciences, we first briefly review its benchmark application to the TSP [9]. In the TSP, the objective is to find the shortest path that connects (visits) each of Y towns once. Let $z_i(t)$, $(i = 1, \ldots, Y)$ be the number of ants in town i at time t, and let Z be the total number of agents (ants). Each artificial ant in town i is an agent who places a substance called "trail" (pheromone for real ants) on a visited path, and then chooses to visit the next town j with a probability that is a function of the town distance, d_{ij}, and of the amount of trail present on the connecting path. Also, each agent is trained to visit a town only once by keeping track of towns visited in a "tabu" list.

Let $\tau_{ij}(t)$ be the intensity of the trail on edge (i,j) at time t. Each ant at time t chooses the next town, where it will be at time $t+1$. Let us define a "run" as Z moves required by the Z ants in the interval $(t, t+1)$. For Y towns it hence will take Y runs of the model to complete an "iteration" during which each ant has completed a tour. The trail intensity τ_{ij} is updated according to the equation

$$\tau_{ij}(t+1) = \rho\tau_{ji}(t) + \Delta\tau_{ij}, \tag{1}$$

where ρ (< 1) is a coefficient such that $(1-\rho)$ represents the evaporation (loss) of trail between times t and $t+1$, and

$$\Delta\tau_{ij} = \sum_{k=1}^{Z} \Delta\tau_{ij}^{k}, \tag{2}$$

where $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of trail substance (pheromone in the case of real ants) placed on edge (i,j) by the kth ant between times t and $t+1$. The quantity $\Delta\tau_{ij}^{k}$ is given by

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if } k\text{th ant uses edge } (i,j) \\ & \text{in its tour (between } t \text{ and } t+1\text{)}, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where Q is a constant related to the quantity of trail laid by ants, and L_k is the tour length of the kth ant.

To satisfy the constraint that an ant must visit all Y towns, a data structure called the tabu list is associated with each ant; this tabu list saves the towns already visited up to time t and forbids an ant to visit the same town again before Y runs (a complete tour) have been completed. When a tour is completed, the tabu list is used to compute the ants' current solution (i.e., the distance of the path covered by an ant). The tabu list is then emptied and the ant is free to again choose. We define tabu_k as the dynamically growing vector that contains the tabu list of the kth ant.

Let us define the "visibility" η_{ij} as $1/d_{ij}$ (i.e., the reciprocal of the distance between two towns). This quantity does not change during a run of the ACO, as opposed to the trail τ that changes according to Eq. (1). We can then define the transition probability from town i to town j for the kth ant as

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^e (\eta_{ij})^f}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^e (\eta_{ik})^f} & \text{if } j \in \text{allowed}_k, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\text{allowed}_k = \{Y - \text{tabu}_k\}$ and where e and f are parameters that control the relative importance of trail versus visibility. The transition probability is a trade-off between the visibility (which states that close towns will be selected with a higher probability, thus implementing a greedy heuristic strategy), and the trail intensity at time t (which states that paths with a lot of traffic must be highly desirable, thus implementing the autocatalytic process). A greedy heuristic strategy is a strategy that helps find acceptable solutions in the early stages of the search process.

After Y runs all ants have completed a tour, and their tabu lists will be full; at this point for each ant k the value of the tour length L_k is computed and the values $\Delta\tau_{ij}^k$ are updated according to Eq. (3). Also, the shortest path found by the ants (i.e., $\min_k L_k,\ k = 1, \ldots, Z$) is saved and all tabu lists are emptied. This process is iterated until the tour counter reaches the maximum (user defined) number of iterations.

Dorigo et al. [8] compared ACO with several other popular algorithms for solving the Oliver30 problem (a 30-town problem) and concluded that with few exceptions, ACO always found the best-known solution.

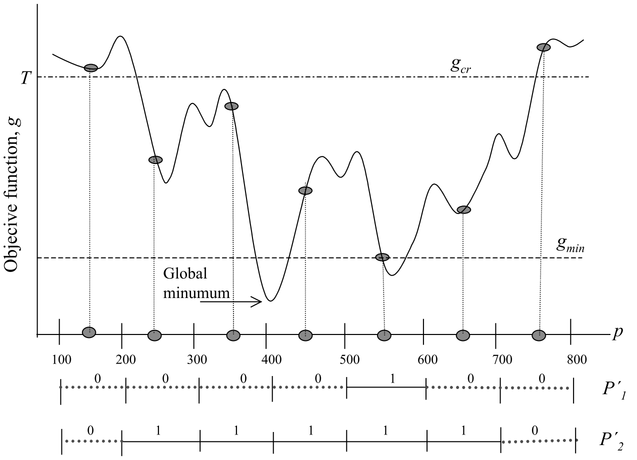

Fig. 2. Illustration of parameter discretization, values of objective function g for different parameter values, $g_{\min}, g_{\mathrm{cr}}$, and parameter updating based on the criteria $(g \leqslant g_{\min})$, and $(g \leqslant T)$.

3. Application of ACO to inverse modeling

The first step in any IM study is the definition of an objective function. Our objective in using IM is to find a parameter set β that minimizes the general function

$$g(\beta) = \sum_{i=1}^{l} w_i \sum_{j=1}^{r_i} \sum_{k=1}^{q_i} w_{ijk}^* \left(v_{ijk} - v'_{ijk} \right)^2, \tag{5}$$

where g is the objective function, β is the vector of input parameters, l is the number of observed variables, v_{ijk} is the measured value of the ith variable at the jth location and the kth time, v'_{ijk} is a corresponding simulated value, w_i is the weight of the ith variables, w_{ijk}^* is the weight associated with individual measurements, q_i is the total number of measurements over time and r_i is the total number of measurements over space for the ith variable.

The algorithm ACO–IM described below is an adaptation of a previously developed program, Sequential Uncertainty FItting (SUFI) [3], to ACO. The main difference between the two programs is in the updating of the prior parameter uncertainty as described in the following paragraphs.

The main concept behind the SUFI algorithm can be summarized in the following steps: (1) depict each unknown parameter by an interval based on available information, i.e., [100–800] for the imaginary parameter p in Fig. 2; (2) discretize each interval into a number of strata, let the middle of each stratum represent that stratum; (3) run the simulation model of choice for all (exhaustive sampling), or a randomly selected subset of all the possible parameter combinations (an iteration); (4) score each stratum on the bases of the smallness of the value of the objective function in such a way that small values of objective function receive higher scores; (5) eliminate from the ends of the parameter intervals the strata with small or no scores, thereby decreasing the initial range of parameter uncertainty, and finally; (6) repeat the process until a desired stopping rule is reached. In SUFI algorithm, each stratum, initially set to a score of zero, receives a score of 1 if the value of the objective function, g, for that stratum satisfies the condition $(g \leqslant g_{\mathrm{cr}})$, where g_{cr} is a user-defined value. In the illustrative example in Fig. 2, for the condition $(g \leqslant g_{\mathrm{cr}} = g_{\min})$, where $g_{\min}$ is the minimum value of the objective function obtained in an iteration, only one stratum receives score, and hence the updated parameter (p'_1), based on this criterion, would have a very narrow range [500–600]. This case will provide a fast but less than optimum solution as the global minimum (at about 400 in Fig. 2) will be missed. For the criterion $(g \leqslant g_{\mathrm{cr}} < T)$, where T has a value greater than $g_{\min}$, the updated parameter would look like the second line in Fig. 2 (p'_2) with a range of [200–700], which is smaller than the initial range of [100–800] but still much larger than the $(g \leqslant g_{\mathrm{cr}} = g_{\min})$ criterion. For this case, there is a larger probability of finding the global minimum but a larger number of iterations must be performed.

For parameter updating, the user in SUFI program is presented by score distributions for several g_{cr} conditions, and is asked to update each parameter by studying the scoring patterns of the different strata [3]. In the following paragraphs we present a more formal description of the above steps, and point out that the scoring, and hence the updating of the parameters in ACO–IM is determined by a function that tries to achieve a balance between different interests, mainly, a desire to reach a fast solution modified by a desire to avoid the local minima.

3.1. Formulating ACO–IM

The ACO–IM problem can be described as follows. Let $g(\beta) : D \to R$, as given by (5), be a continuous and bounded function, and β be a b-dimensional state vector. Also, let R be real numbers, and D be the search space of β. The objective is to find the state vector β that minimizes $g(\beta)$. Without loss of generality, we can take D as the hyperparallelepiped

$$D = \{\beta_i | \beta_i^- \leqslant \beta_i \leqslant \beta_i^+ ; i = 1, 2, \ldots, b\},$$

where β_i^- and β_i^+ denote, respectively, the lower and the upper bounds of the parameter β_i. The upper and the lower bounds are based on prior information using existing knowledge.

Since ACO has been developed for ordered problems, its application to a continuous problem requires first discretization of the search space, D. This is done by dividing the interval $[\beta_i^-, \beta_i^+]$ of each parameter β_i into a number, say m_i, of strata. If each stratum be represented by the value at the middle of the stratum, then there will be $M = m_1 m_2, \ldots, m_b$ permutations or possible pathways through the space of input parameters. We refer to M as the exhaustive sampling scheme (ESS). Let $\beta_{ij} (i = 1, \ldots, b$ and $j = 1, \ldots, m_i)$ be the stratum j of parameter β_i in the search space of D.

The above discretization scheme is schematically shown in the right-hand side of Fig. 3 for four parameters, with each parameter interval $[\beta_i^-, \beta_i^+]$ divided into five strata. There are no hard rules for choosing the stratification of the parameters, except that initially they should uniformly cover the complete parameter space, unless additional information exists that would favor one region of parameter space above another. While having more strata would speed up convergence of the ACO problem, this advantage should be balanced against the speed of the simulation program since the number of calls to a simulation program rapidly increases as the number of strata increases. To limit the number of function calls, one could select only a random subset of M, or N, which we refer to as a random sampling scheme (RSS), and/or invoke parallel processing. The ACO–IM algorithm easily lends itself to parallel processing techniques. Based on our experience so far, $N \geqslant 0.1$ M generally leads to optimum solutions, while and $N < 0.1$ M often leads to less than optimum solutions.

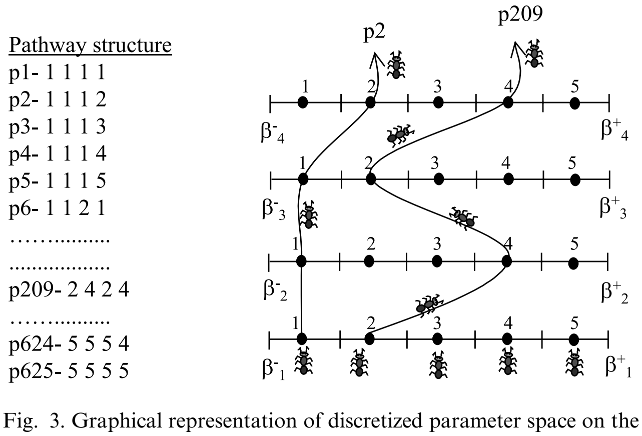

Fig. 3. Graphical representation of discretized parameter space on the right and the pathway structure on the left.

3.2. Assigning tasks to ant agents

To satisfy the constraint that each ant must follow only one pathway, we generate a pathway-structure list as shown in the left side of Fig. 3. Using this list, each ant will follow one of M, or N pathways, i.e., the pathway (1,1,1,2), or (2,4,2,4) as shown in Fig. 3. Therefore, at the most we need a total of M ants, with each ant acting like an agent having the following tasks:

- selecting a pathway from the pathway list and remembering the values of the parameter strata along its pathway,
- at the end of the pathway, passing the parameter values to a simulation program,
- getting simulated values back from the simulation program,
- calculating the value of the objective function for its pathway, and
- on the basis of the value of the objective function, placing a certain amount of trail (pheromone in the case of real ants) on each stratum visited along its pathway.

Once all ants complete the above tour, one ‘ant-cycle’ or one ‘iteration’ is completed. Eventually the trail will be used to score a pathway. The low scoring pathways will vanish (in the case of real ants the pheromone evaporates on pathways with less traffic) and new pathways evolve around high scoring paths.

3.3. Calculating the trail and scoring parameter strata

Let $\tau_u(I)$ be the intensity of trail on pathway u $(u = 1, \ldots, N)$ and iteration I. For the example in Fig. 3, ant number 209 places the same amount of trail on stratum 2 of the first parameter, stratum 4 of the second parameters, stratum 2 of the third parameter, and stratum 4 of the fourth parameter. In the TSP problem the trail intensity was defined as a function of the length of a path. In ACO–IM we use value of the objective function of a pathway to assign the trail. The trail intensity on each path is calculated in such a way that the larger values of the objective function (i.e., longer paths for real ants) receive smaller trails. This is accomplished according to the following equation

$$\tau_u(I) = \begin{cases} \exp\left(4.6\left(\frac{g_u - g_{\mathrm{cr}}}{g_{\min} - g_{\mathrm{cr}}}\right)\right), & g_u \leqslant g_{\mathrm{cr}}, \\ 0, & g_u > g_{\mathrm{cr}}, \end{cases} \qquad (6)$$

where g_u is the value of the objective function for the uth pathway, g_{cr} is a critical value (to be defined later) above which $\tau_u = 0$ (see Fig. 2), and $g_{\min}$ is the minimum of all g_us $(u = 1, \ldots, N)$ for an iteration. Note that in an

ACO–IM problem the trail intensity is no longer a function of time, as it was in the TSP, and is defined per iteration. The above equation states that pathways with values of the objective function larger than g_{cr} receive no trail (for real ants the trail on longer paths evaporates quickly), while below g_{cr}, the intensity of trail is exponentially larger for smaller values of the objective function (for real ants a shorter path from nest to food source receives increasingly more pheromone). At $g_{\min}, \tau_u(I)$ is set equal to an arbitrary value of 100 (this value would be equivalent to Q in the TSP, which is incidentally also equal to 100).

Note that any single stratum β_{ij} in D may be the crossroad of many ant pathways. The trail share (calculated with Eq. (6)) of each stratum β_{ij} from each pathway can be summed to yield ϕ_{ij} as

$$\phi_{ij}(I) = \sum_{u \in \text{crossing pathways}} \tau_u(I), \tag{7}$$

where crossing pathways are all the pathways that cross stratum β_{ij}. In addition, let σ_{ij} (I) be the standard deviation of all values of the objective function involving β_{ij}. The value of σ_{ij} (I) depicts the sensitivity of the β_{ij} stratum with respect to other strata in D. Fig. 4 illustrates a pattern of both ϕ_{ij} and σ_{ij} distributions over the strata of the hypothetical four-parameter example in Fig. 3.

To assess the evolution of new pathways through parameter spaces we must formulate a trade-off between the following three considerations: (1) the autocatalytic process that accounts for rapid discovery of good solutions. This objective is achieved through Eq. (6) where pathways with smaller values of the objective function receive exponentially higher trails, (2) the greedy heuristics strategy that helps one to find acceptable solutions in the early stages of the search process. This objective is achieved by selecting g_{cr} to be as close as possible to $g_{\min}$ (see Fig. 2), and (3) the distributed computation strategy that avoids premature convergence. This objective is achieved by giving more weight to strata with higher sensitivities σ_{ij}.

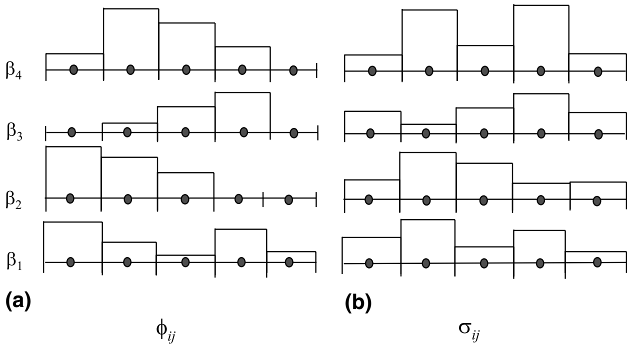

Fig. 4. Typical trail intensity distribution (a) and sensitivity distribution (b) for each parameter stratum. The larger bars indicate larger trail values and higher sensitivities.

In Fig. 4(a), regions of larger trail intensity (shown with larger bars) are generated by smaller values of the objective function, and hence are more likely to contain the desired optimum solution. This suggests that we may eliminate regions with zero or small amounts of trail (a move that may constitute a greedy behavior). On the other hand, strata with large sensitivities (Fig. 4(b)) may not have been represented well by their middle points and should not be eliminated.

In order to decide how to update the parameters, we define the term 'score', S_{ij}, for each β_{ij} calculated by the following expression similar to the transition probability of the TSP problem

$$\underset{g_{\mathrm{cr}}=T}{S_{ij}} = \frac{(\phi_{ij})^A (\sigma_{ij})^N}{\sum_i \sum_j (\phi_{ij})^A (\sigma_{ij})^N}, \tag{8}$$

where we let the value of $A = 1$, and calculate N and T by using

$$\begin{aligned} N &= c_N \frac{\sigma_{ij}}{\mu_{ij}}, \\ T &= g_{\min} + c_T \frac{\sigma_g}{\mu_g}, \end{aligned} \tag{9}$$

where μ_{ij} is the mean of trail of the β_{ij} strata, σ_{ij} is the standard deviation of trail on the β_{ij} strata, μ_g and σ_g are, respectively, the mean and the standard deviation of the objective function for an iteration, and c_N and c_T are constants that were determined to be about 0.3 and 0.5, respectively (see Appendix A for more details). The ratio of standard deviation to mean is known as the variability and our objective in scoring each stratum through Eq. (8) is to assign a larger value to T if the variability of the objective function is large (refer to Fig. 2) and to assign also a larger value to N if the sensitivity (σ_{ij}) of the stratum β_{ij} is large, hence satisfying the criteria for distributed computation.

The parameters A, N, and T in (8) control the relative importance of the trail intensity (which implements the autocatalytic process) versus sensitivity (which addresses the demand for distributed computation). Although the greedy nature of artificial ants dictates that parameters be updated on the basis of the smallest value of T (i.e., $T = g_{\min}$) alone, the addition of σ_{ij} and ϕ_{ij} influences an ant's decision, the effect of which is similar to the effect of a mutation operator in a genetic algorithm. [24] We refer henceforth to the above scoring strategy as the *ANT* scoring strategy. A definition of the parameters involved in ACO–IM appears in Table 1.

3.4. Updating of parameters and stopping rules

Once the scoring is completed, parameters are updated by eliminating the strata with small scores from

Table 1
Definition of the parameters involved in ACO–IM

Parameter	Definition
A, N, T	Parameters controlling the effect of ϕ_{ij}, σ_{ij}, and g_{cr}, respectively, on S_{ij}
c_N, c_T	Constants associated with parameters N and T
$g_{\min}$	Minimum value of the objective function in an iteration
g_{cr}	Critical value of objective function above which a pathway receives no trail
S_{ij}	Score of stratum j of parameter i
β_{ij}	Stratum j of parameter β_i
μ_g, σ_g	Mean and standard deviation of objective function in an iteration
τ_u	Amount of trail laid on pathway u by a crossing ant
ϕ_{ij}	Total amount of trail accumulated on stratum j of parameter i
σ_{ij}	Standard deviation of objective function for stratum j of parameter i

both ends of the intervals. This results in parameters with narrower ranges (uncertainties). Note that if the high scoring strata fall on either end of the parameter intervals, the parameters could be extended in that direction. This makes the initial guess of the parameter intervals not very crucial, and at the same time allows a deleted parameter stratum to re-enter the calculation. The system is subsequently reinitialized with updated parameters and the process repeated.

The iterations continue until a stopping rule is satisfied. This may occur when (1) a desired value of the objective function is reached, or (2) no more changes in the value of the objective function are obtained in consecutive iterations, or (3) the number of precision digits used to signify a parameter becomes more than the accuracy needed to represent that parameter value, or (4) the 95% confidence interval of the variable distributions, resulting from the propagation of uncertainty in the parameters, contains 95% of the measurements (plus their errors).

The above distribution is obtained by Monte-Carlo through the sampling of the ant paths at the end of each run by simply looking at the distribution of variable values corresponding to the different paths. The 95% confidence interval for a variable (95PCI) is obtained by calculating the cumulative distribution and marking the variable values at 0.025 and 0.975.

For environmental projects, where there are usually measurements errors and the models often do not capture all of the relevant process and boundary conditions, we propose using option 4 above as a more appropriate stopping rule. In Fig. 5 we illustrate the concept of 'conditioned parameter distribution'. The first iteration of the ACO–IM is based on a prior estimate of the uncertainty domain of the model parameters and therefore, because of the often large initial uncertainties, the 95PCI of any model variable is large. This is shown by the space within the example curves '*a*' in Fig. 5. As iterations proceed, the uncertainty domain of the parameters become smaller and the parameter distributions are more and more conditioned on the measurements of the variable(s) used in the objective function. Thus, conditioned parameter distribution in the context of this study refers to parameter distributions such that, when propagated stochastically through a simulation program, the 95PCI of the simulated variable would contain more than 95% of the measured points (i.e., curves '*b*' in Fig. 5). In ACO–IM, the iterations can be continued until the upper and the lower limits of the 95PCI coincide with a single curve. This curve is produced by a set of single-valued parameters generally referred to as the 'best-fit parameters' (curve '*c*' in Fig. 5). However, as illustrated by curve '*c*', fitted parameters produce simulations that often miss most of the measured points. In our opinion, fitted parameters are inadequate for analysis of environmental problems if used without the uncertainty associated with them. In gradient type optimization programs such as Levenberg–Marquardt (LM) [17], the covariance of the parameter matrix and hence the 95% confidence interval associated with each parameter are based on linear regression analysis. The problem, therefore, with the calculated parameter uncertainties are that they hold only

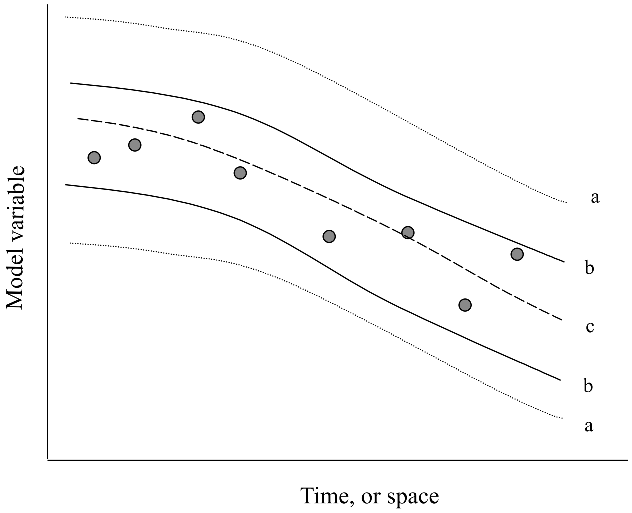

Fig. 5. Simulations based on the 'conditional parameter distribution' concept vs. the 'best-fit parameters'. Best-fit parameters produce simulations that usually miss most of the measurements, that is, curve *d*, whereas conditioned parameter distributions produce simulations that respect most of the measurements, that is, curves *c*.

approximately for the non-linear analysis [13]. Instead, to account for parameter uncertainties, we suggest obtaining conditioned parameter distributions, where most of the data points are respected within the 95PCI.

Furthermore, the problem of algorithm convergence, as the result of correlation among parameters is perhaps less severe in obtaining conditioned parameter distributions than fitted parameters. This is because the former considers ranges rather than point values. However, as with any other algorithm, in case of strong parameter correlation in ACO–IM it would also be necessary to constrain one of the dependent parameters.

4. Testing ACO–IM

4.1. Test case 1: application of ACO–IM to an unsaturated flow problem

To test the ACO algorithm, we used Example 2 from the HYDRUS-1D and HYDRUS-2D Windows-based software packages [26,27]. We chose this example, firstly, because it considers a simple and yet non-trivial one-dimensional water flow in a two-layered field profile of the Hupselse Beek watershed in the Netherlands, and secondly, because we can assume that we know the 'true' parameters and test how closely ACO–IM can estimate them.

The two-layered system being simulated is schematically described in Fig. 6. The equation being solved is a one-dimensional isothermal Darcian flow in a variably saturated rigid porous medium, expressed by the following form of the Richards equation:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x}\left(K\frac{\partial h}{\partial x} + K\right) - S, \tag{10}$$

where θ is the volumetric water content (cm^3 cm^{-3}), h is the pressure head (cm), S is the sink term (d^{-1}), K is the hydraulic conductivity (cm d^{-1}), z is a vertical coordinate (cm), and t is time (d). Atmospheric data and observed groundwater levels provided the required boundary conditions for the numeric model. The soil profile consisted of two layers: a 40-cm thick A-horizon, and a B/C-horizon that extended to a depth of 300 cm (Fig. 4). The depth of the root zone was 30 cm. The soil surface boundary conditions involved actual precipitation and potential transpiration rates for a grass cover. The surface fluxes were incorporated by using average daily rates distributed uniformly over each day. The bottom boundary condition consisted of a prescribed drainage flux-groundwater level relationship given by [27]

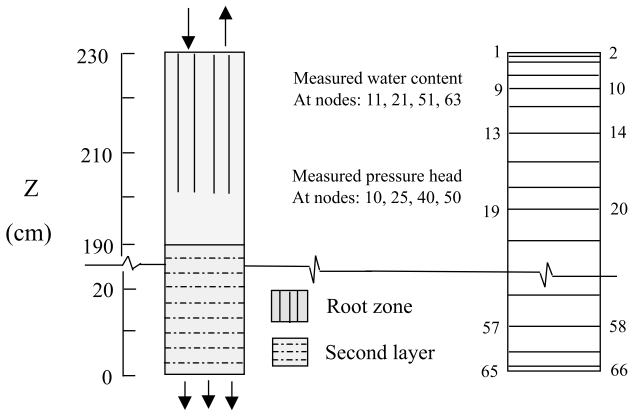

Fig. 6. Flow system and the finite element mesh for the first example.

$$q(h) = -a_1 \exp(a_2 \mid h - G \mid), \tag{11}$$

where a_1 and a_2 are empirical parameters, and G is the reference position of the groundwater level.

The initial moisture profile was taken to be in equilibrium with the initial groundwater level at 55 cm below the soil surface. Soil hydraulic characteristics were described according to the van Genuchten model [29] for the water retention curve,

$$S_e(h) = \frac{\theta(h) - \theta_r}{\theta_s - \theta_r} = \frac{1}{[1 + |\alpha h|^n]^m} \quad h < 0 \tag{12}$$

and

$$\theta(h) = \theta_s \qquad h \geqslant 0 \tag{13}$$

and the van Genuchten–Mualem model [18] for the hydraulic conductivity function

$$K(h) = \begin{cases} K_s S_e^{0.5}\left[1 - \left(1 - S_e^{1/m}\right)^m\right]^2, & h < 0, \\ K_s, & h \geqslant 0, \end{cases} \tag{14}$$

where α and n are shape parameters, S_e is effective water saturation, $m = 1 - 1/n, \theta_r$ and θ_s are the residual and saturated water contents, respectively; and K_s is the saturated hydraulic conductivity.

The water retention and flow curves for both soils are shown in Fig. 7. For the inverse problem we assumed that the following six parameters were unknown: $\alpha_1, n_1, K_{s1}\alpha_2, n_2$, and K_{s2}. The information usually available for inverse analyses are some variables (e.g., pressure head, water content, concentration, or flux from the bottom of a soil column or a lysimeter) measured at certain depths and at certain times. In this test case we arbitrary assumed to have 20 pressure head data measured on days 120, 160, 200, 240, 270, and at nodes 11, 25, 40, and 50; and 12 water content data measured on days 130, 180, and 220, and at nodes 11, 21, 51, 63. These 'measurements' were taken from the output files of Example 2 of HYDRUS_2D. Then, the objective function was written as

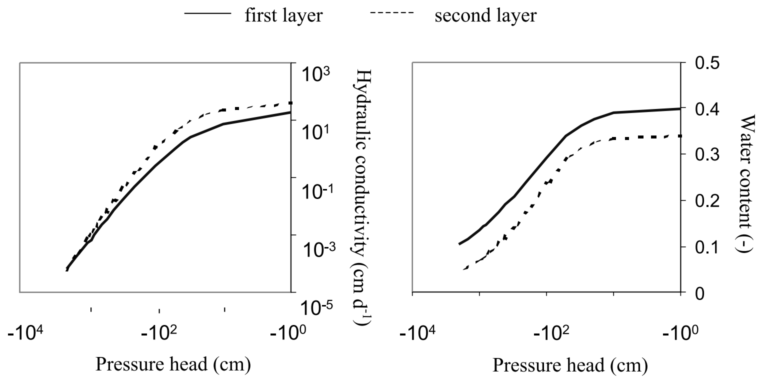

Fig. 7. Unsaturated hydraulic properties of the first and second layers of the first example.

$$g(\alpha_1, n_1, K_{s1}, \alpha_2, n_2, K_{s2}) = 6.72 \times 10^{-6} \sum_{j=1}^{4} \sum_{k=1}^{5} \left(h_{jk} - h'_{jk}\right)^2 + 40.48 \sum_{j=1}^{4} \sum_{k=1}^{3} \left(\theta_{jk} - \theta'_{jk}\right)^2, \tag{15}$$

where h is the measured pressure head, h' is the simulated pressure head, θ is the measured water content, θ' is the simulated water content, and the weights were calculated according to the relationship $(1/k\sigma^2)$ for each variable with σ being the standard deviation of a variable and k being the number of samples [6].

Initially, we assumed a large range of uncertainties for the unknown parameters, as listed in Table 2. We estimated the unknown parameters using various stratification strategies and various number of searching ant agents. All tests with the number of ant agents exceeding 10% of the exhaustive set M led to basically the same results (±2% differences in the estimated parameters).

In the run reported here, initially we divided each parameter into 5 strata, which produced a total of $M = 15\,625$ exhaustive runs. But we executed only 20% of the exhaustive set and used $N = 3125$ ant agents to do the search. We stopped after the sixth iteration with the results summarized in Table 2. The entire run took about two hours on a Pentium 600 MHz computer. The stratification and the scoring for the greedy and the *ANT* strategy are shown in Table 3 for the first iteration. Note once again that the difference between greedy and *ANT* strategy is that for greedy strategy we only use $g_{\min}$ as the only criteria for scoring the parameter strata. Hence, only one stratum per parameter receives score, while for the *ANT* strategy more strata combinations receive scores. The consequence of this is that the updated parameters based on the greedy strategy are more compact than a corresponding *ANT* strategy. For convenience, in *ANT* strategy, we divided the scores of all strata for each parameter by the smallest score; hence, there is always one stratum with a score of 1.

After the first iteration, we followed separately the greedy path and the *ANT* path for parameter updating. Table 2 shows the results of both paths. For the greedy path the third and the fourth iterations produced the same value of the objective function, hence we considered the third iteration as the final one. While for the *ANT* strategy we stopped after the sixth iteration when the measured and expected simulated pressure head and

Table 2
Initial parameter uncertainty range and final estimates obtained using the *ANT* and the greedy strategies

Unknown parameter (true values)	Initial uncertainty range	Final parameter and uncertainty range based on ANT strategy	Final parameter and uncertainty range based on the greedy strategy
α_1 (0.0174)	[0.0–0.05]	0.0173 [0.0172–0.0175]	0.02425 [0.022–0.028]
n_1 (1.3757)	[1.0–2.0]	1.38 [1.371–1.382]	1.29 [1.28–1.36]
K_{s1} (29.75)	[0.0–200.0]	28.85 [28.74–29.82]	96.0 [80.0–112.0]
α_2 (0.0139)	[0.01–0.03]	0.01386 [0.01381–0.01393]	0.01435 [0.0112–0.0148]
n_2 (1.6024)	[1.5–3.0]	1.604 [1.5983–1.6044]	1.65 [1.56–1.8]
K_{s2} (45.34)	[0.0–200.0]	44.6 [44.43–45.76]	32.8 [16.0–40.0]
Objective function	0.53	0.000019	0.029

Table 3
Summary of parameter stratification, strata scores, and updated parameters based on greedy and *ANI* strategies for iteration 1 of example 1

Iteration 1 Parameter stratification						
α_1	0.0	0.01	0.02	0.03	0.04	0.05
n_1	1.0	1.2	1.4	1.6	1.8	2.0
K_{s1}	0.0	40.0	80.0	120.0	160.0	200.0
α_2	0.01	0.016	0.022	0.028	0.034	0.04
n_2	1.5	1.8	2.1	2.4	2.7	3.0
K_{s2}	0.0	40.0	80.0	120.0	160.0	200.0

Greedy strategy:

	Score						
α_1	0	0	1	0	0		$\alpha_1 = [0.02–0.03]$
n_1	0	1	0	0	0		$n_1 = [1.2–1.4]$
K_{s1}	0	0	1	0	0	$\xrightarrow{\text{Updated parameters}}$	$K_{s1} = [80–120]$
α_2	1	0	0	0	0		$\alpha_1 = [0.01–0.016]$
n_2	1	0	0	0	0		$n_2 = [1.5–1.8]$
K_{s2}	1	0	0	0	0		$K_{s2} = [0–40]$

ANT strategy:

	Score						
α_1	0	3	2	1	0		$\alpha_1 = [0.01–0.04]$
n_1	0	7	4	1	0		$n_1 = [1.2–1.8]$
K_{s1}	1	1	1	1	1	$\xrightarrow{\text{Updated parameters}}$	$K_{s1} = [0–200]$
α_2	8	5	3	2	1		$\alpha_1 = [0.01–0.034]$
n_2	4	2	2	1	1		$n_2 = [1.5–2.4]$
K_{s2}	1	1	1	1	1		$K_{s2} = [0–200]$

water content curves were practically indistinguishable as shown in Fig. 8. The sequential nature of the algorithm allows us to continue iterations until arriving at the exact solution. This would not be possible with the greedy strategy.

It appears that the *ANT* scoring strategy, while using the greedy force through *T* to obtain a solution at the early stages of the search, receives proper signals from the autocatalytic parameter *A* and the distributed computation parameter *N* to jump over local minima and arrive at an optimum solution.

Fig. 9 illustrates the 95PCI for the pressure head and water content based on the values of the third, fifth, and sixth iterations at a depth of 10 cm below the soil surface as a function of time. Also shown in Fig. 9 are the measured values and the expected values of the simulated variables for the sixth iteration. As iterations continue, the uncertainty ranges become smaller, leading to narrower 95PCI. Note that for this "well-defined" problem (i.e., the measurements were model outputs without any measurement errors, the boundary conditions as well as the processes were exactly defined) the expected simulation (i.e., at the 50% confidence interval) along with the 95PCI for the sixth iteration all coincide with the measured data. This normally would not happen in a real-world application.

4.2. Test case 2: comparison of ACO–IM with LM method

In this section we describe a comparison of the results of ACO–IM with the LM [17] method implemented in HYDRUS-1D [26]. The example used is based on an experiment carried out in a large lysimeter (3 m^2 by 1.5 m) planted with forest vegetation designed to study the effect of elevated CO_2 and nitrogen on forest ecosystems [28]. We chose this specific example because the data has all the noise that one would expect in a field experiment but with controlled boundary conditions. The important processes included flow, transport of bromide, and root water and solute uptake. Another reason for using this example was to demonstrate that IM could be more than parameter fitting, and has the potential for being used for system analyses.

The bromide ($CaBr_2$) tracer experiment began in June 1996, when 30 g of bromide was diluted in 10 l of water, and distributed homogeneously below the canopy. The drained water was collected for a period of over one year after the tracer application and the bromide concentration was measured on a regular basis. Pressure head was measured with vertically installed tensiometers at 25, 50, and 75 cm depth. Three replicate instruments were installed at each depth and their

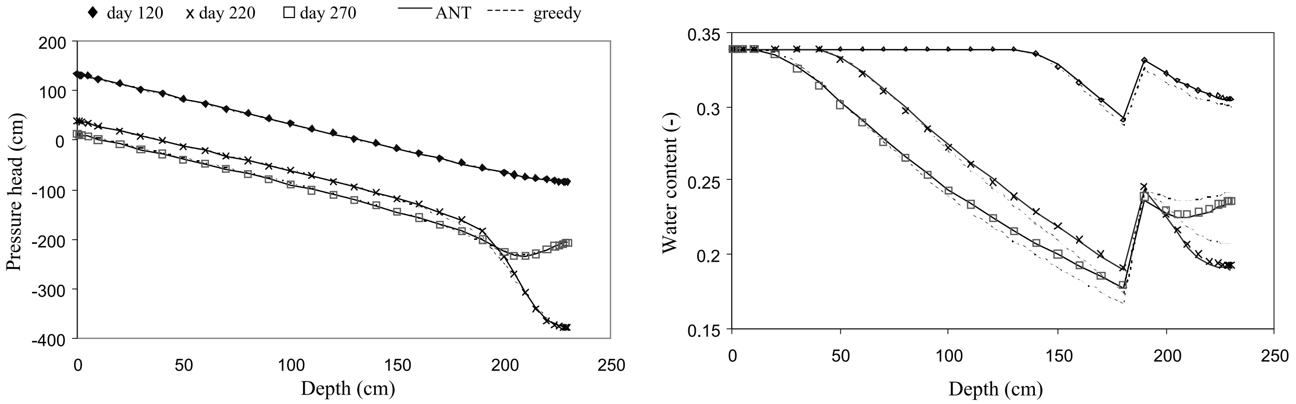

Fig. 8. Measured (symbols) and simulated pressure head and water content obtained by the greedy and *ANT* scoring strategies.

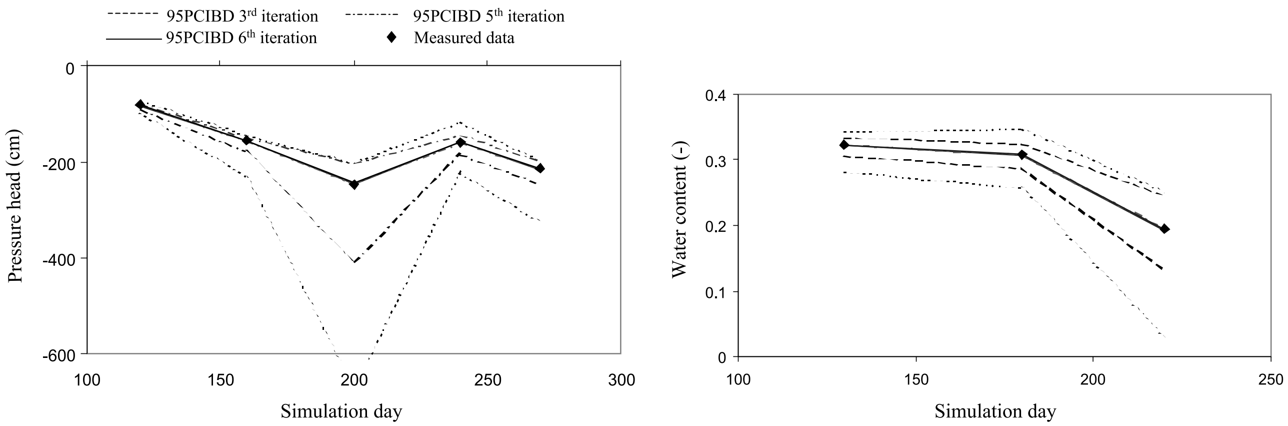

Fig. 9. Graphs of the 95% confidence intervals for pressure head and water content for the third, fifth, and sixth iteration showing the progression to a solution.

averages used to represent the pressure head at that depth. Water content was measured with time-domain-reflectometry (TDR) probes of 25 cm length, vertically installed with centers at depths 25, 50, and 75 cm. Again, all installations were in triplicates, and the averages were used to represent the water content at a depth. The lysimeter was regularly irrigated and actual evapotranspiration calculated using the following water balance equation

$$ET_a = I - D - \Delta\theta_z, \qquad (16)$$

where ET_a is actual evapotranspiration, I is irrigation, D is drainage, and $\Delta\theta_z$ is the difference in water storage, all expressed as amounts cumulated for the period of the ET_a measurement.

The governing flow equation and the soil hydraulic properties were as described for the previous test case. Transport was described by a convection–dispersion expression for a miscible, non-sorbing, and conservative component as given by

$$\frac{\partial \theta c}{\partial t} = \frac{\partial}{\partial z}\left(\theta D \frac{\partial c}{\partial z}\right) - \frac{\partial q c}{\partial z} - SC_s, \qquad (17)$$

where c is the solute concentration in solution (g cm^{-3}), D is the dispersion coefficient (cm^2 d^{-1}), q is Darcian flux (cm d^{-1}), S is the root water uptake, and C_s is the concentration of soil solution taken up by plant roots (g cm^{-3}).

The flow boundary conditions were expressed as:

$$-K\left(\frac{\partial h}{\partial z} - 1\right) = q_T(t), \quad z = 0, \qquad (18)$$

$$-K\left(\frac{\partial h}{\partial z} - 1\right) = q_B(t), \quad z = L, \qquad (19)$$

where q_T is the net infiltration (cm d^{-1}) at the top, and q_B is the measured flux (cm d^{-1}) at the bottom boundary. The initial and boundary conditions for the concentration were expressed as:

$$c(z,t) = 0, \quad t = 0, \qquad (20)$$

$$\theta D \frac{\partial c}{\partial z} + qc = qc_0(t), \quad z = 0, \qquad (21)$$

$$\frac{\partial c}{\partial z} = 0, \quad z = L, \qquad (22)$$

where c_0 is the concentration of the infiltrating water, and q_0 is the Darcian flux density evaluated at the soil surface.

The objective function was formulated as

$$g(\alpha_1, n_1, K_{s1}, \alpha_2, n_2, K_{s2}) = 1.13 \times 10^{-6} \sum_{j=1}^{3} \sum_{k=1}^{41} \left(h_{jk} - h'_{jk}\right)^2 + 4.16 \sum_{j=1}^{3} \sum_{k=1}^{41} \left(\theta_{jk} - \theta'_{jk}\right)^2, \qquad (23)$$

where the three space measurements were furnished with tensiometer and TDR data at 25, 50 and 75 cm depths. Since the TDR data represented an average over 25 cm vertical depth, the model outputs were also averaged over 25 cm. The measurement times for both variables were on days (13, 14, 16, 17, 20, 24, 26, 33, 40, 48, 54, 58, 65, 73, 79, 90, 97, 104, 111, 119, 126, 133, 139, 155, 162, 169, 174, 279, 284, 291, 300, 306, 312, 319, 327, 333, 340, 347, 354, 361, 368) where day 13 corresponded to June 5, 1996.

Although the lysimeter was originally uniformly packed, as the result of settling and root distribution a more compacted bottom layer was formed at about 40 cm depth. Therefore, we considered two layers and estimated six parameters as in the test example 1, mainly, $\alpha_1, n_1, K_{s1}, \alpha_2, n_2$, and K_{s2}. The stratification strategy for α, n, and K_s were 4, 4, and 10, respectively, and we simulated only 10% of the exhaustive set and used only $N = 2560$ ant agents.

Table 4 summarizes the initial and final parameters obtained by ACO–IM and the final results obtained by

Table 4
Initial parameter uncertainty range and final estimates obtained using Levenberg–Marquardt and ACO–IM algorithms

Unknown parameter	Initial uncertainty distribution used in ACO–IM	Expected parameter values and the final uncertainty distribution	Fitted parameter and the 95% CI determined by Levenberg–Marquardt
α_1	[0.0–1.0]	0.937 [0.5–1.0]	0.867 [0.478–1.256]
n_1	[0.1–3.0]	1.288 [1.1–1.6]	1.282 [1.242–1.321]
K_{s1}	[300–100 000]	39 110 [300–60 000]	31 627 [1804–61 450]
α_2	[0.0–1.0]	0.4062 [0.25–0.5]	0.124 [0.077–0.172]
n_2	[1.1–3.0]	1.288 [1.1–1.6]	1.518 [1.389–1.648]
K_{s2}	[100–100 000]	360 [100–7000]	159 [13–305]
Objective function	0.53	0.65	0.63

LM algorithm along with the associated uncertainties. Fig. 10 shows the comparison of LM and ACO–IM in simulating pressure head and water content at the depth of 25 cm. For the other depths Figures are not shown but the comparisons were similar with both models having practically indistinguishable curves. Only slight deviations of water content at 50 and 75 cm depths resulted in ACO–IM having a slightly larger objective function. A review of the parameters in Table 4 reveals that the identical simulation results in Fig. 10 were obtained with widely different parameters indicating that the set of hydraulic parameters considered was not very important in this particular study. This is also evident from the large uncertainties both algorithms determined for the parameters.

We began the ACO–IM simulation with very large initial parameter uncertainties so as to capture all the measured data within the 95% confidence band. But this was never achieved for the entire profile regardless of the degree of parameter uncertainties. Figs. 11 and 12 show the 95PCI for pressure head and water content for the second iteration at the depths of 25 and 75 cm, respectively. Although most of the data at the 25-cm depth was captured within the 95PCI, but for the 50-cm depth (Figure not shown) and the 75-cm depth a large number of pressure head data was missed. For this example we consider further conditioning (iteration) of parameters on the measured data not to be justified, and the parameters reported in Table 4 with the large range of uncertainties associated with them are the best results one could obtained under the given conditions.

To test the hydraulic parameters obtained in Table 4, we simulated the breakthrough of bromide. Keeping the hydraulic parameters constant, we fitted two transport parameters, dispersivity and maximum plant uptake. Again, these parameters were not very sensitive, and irrespective of the initial uncertainty distribution, the 95PCI could not contain the measured bromide concentration in leachate water. The measured values along with the simulations with LM and ACO–IM are shown in Fig. 13. It is evident that the measured Br^- breakthrough curve exhibits a more complex behavior than what was simulated. One explanation for the lack of significance in the parameters is that this particular problem, contrary to the previous example, is process- rather than parameter-controlled. In other words,

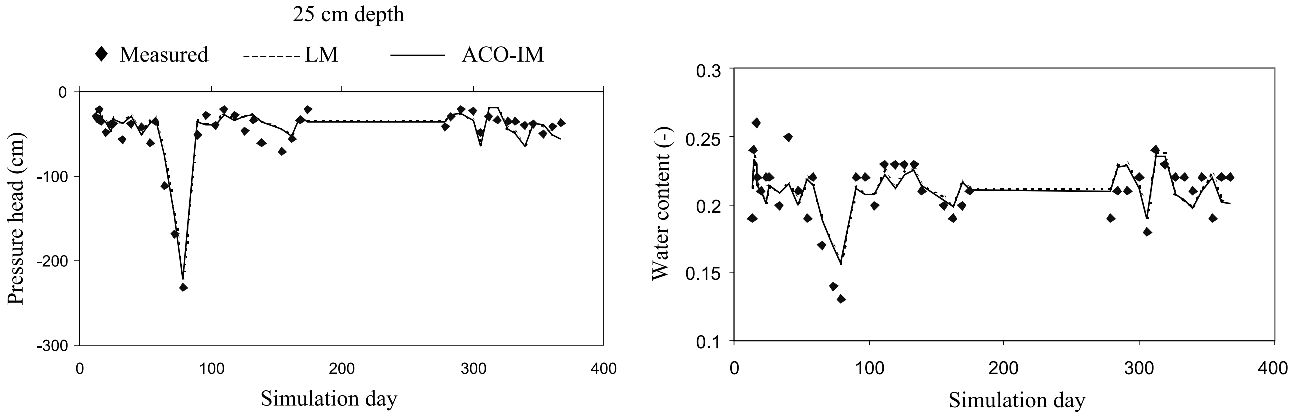

Fig. 10. Comparison of the final simulation results obtained by LM and ACO–IM algorithms for the second example.

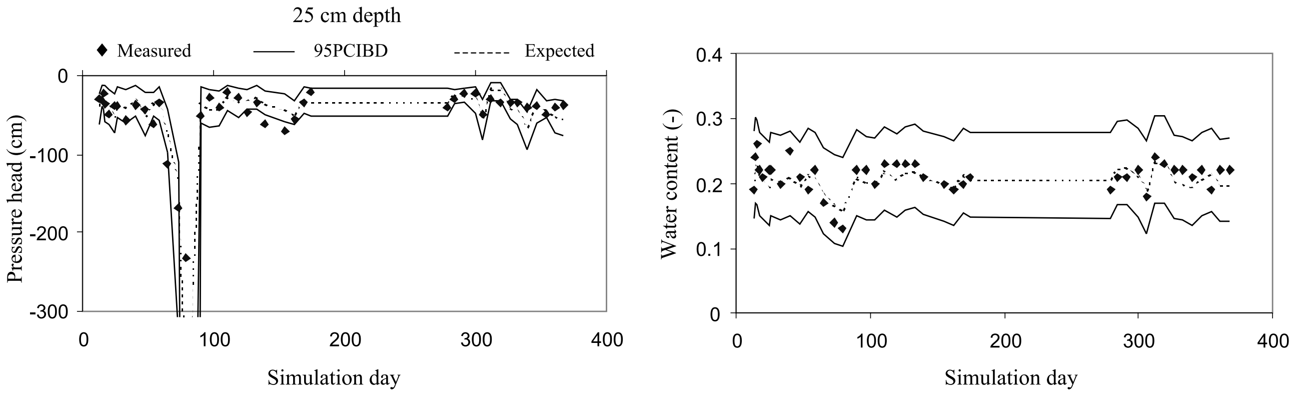

Fig. 11. The 95% confidence interval for the pressure head and water content based on the second iteration at the 25-cm depth.

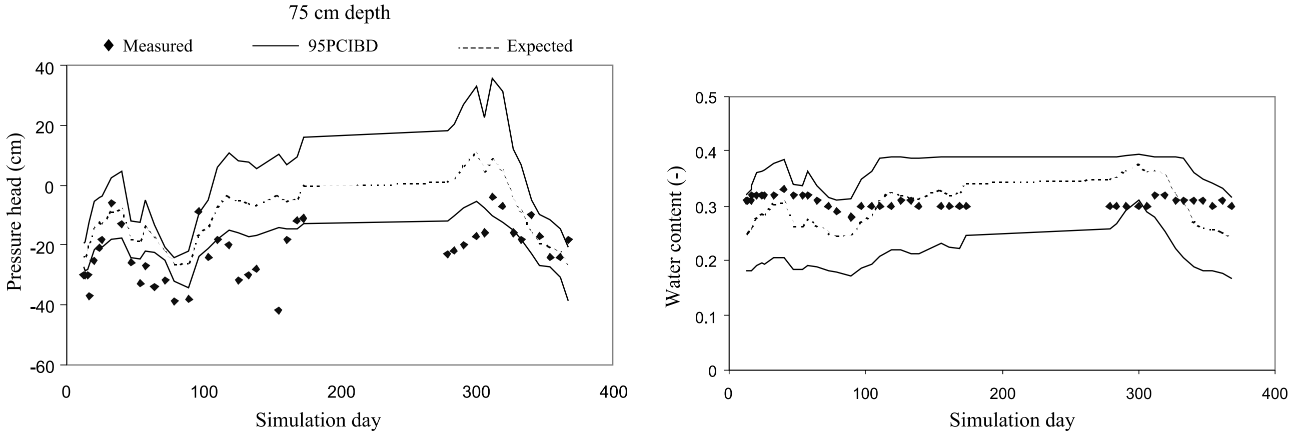

Fig. 12. The 95% confidence interval for the pressure head and water content based on the second iteration at the 75-cm depth.

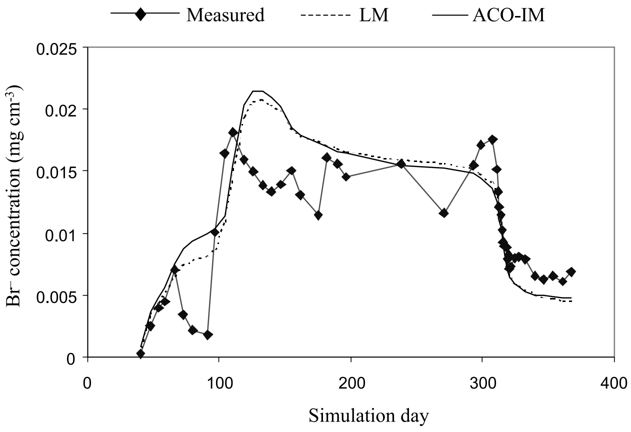

Fig. 13. Measured and simulated Br^- breakthrough curves.

variables water content, pressure head, and concentration are also partly governed by processes not considered by the simulation model. This revelation by inverse analysis is quite useful and indicates that IM can be used for process analysis as well as parameter estimation as also shown by the authors elsewhere [1]. In this example it would be meaningless to condition any further the parameters on the data and expect the parameters to account for responses that were caused by processes other than those included in the simulation model.

The peak dominated Br^- breakthrough curve is indicative of preferential flow in the soil system. This could also be inferred from the very large estimates of the saturated hydraulic conductivity produced by both inverse algorithms for the first layer. If there are processes in the system not accounted for by the simulation models, inverse models try to compensate for missing processes by unrealistically over- or underestimating some of the parameters. This is a pitfall of IM that needs more attention. Furthermore, the progressively worsening of simulations as a function of increasing depth could be a sign of inadequate modeling of root distribution and, hence, water and passive solute uptake by plants. Water content simulations are relatively better than pressure heads for all depths (Figs. 11 and 12) because they are averages over 25 cm depths.

Finally, some of the advantages and disadvantages of the different IM algorithms can be summarized as below: (1) In ACO–IM degree of parameter conditioning may be directly controlled through iterations. The more iterations we perform the more conditioned the parameters being sought will become on the set of measured data. In general, we do not recommend obtaining highly conditioned parameters, as they will perform poorly when simulating other variables not included in the objective function [2]. In LM algorithm, degree of conditioning can only be controlled indirectly by restarting the program with several initial guesses, and in general, a local minimum nearest to the initial estimates is produced. (2) In LM algorithms parameter estimates are given as single-valued parameters plus a summary statistics that expresses the parameter uncertainty usually in terms of the 95% confidence interval. The parameter uncertainties need yet to be propagated before we can assess the degree of conditionality, while in ACO–IM the uncertainties, which play a central role, are already propagated. After each iteration, the 95PCI is reviewed as a guide for further iteration. (3) The LM method requires many trials with initial guesses of the unknown parameters and at the end one is never sure whether to perform another iteration of not. While in ACO–IM this uncertainty does not exist. (4) In ACO–IM, the parameters A, N, and T need yet a more robust estimate, the process is more interactive than LM (which may be considered a disadvantage by some, although we favor this interaction), and subjective interference with the parameter updating may be required, which calls for some understanding of the problem and depends, more than the LM method, on the experience of the user. The subjective interference may be in the form of setting upper and lower limits for the parameters, although this option also exists in the inverse option of the HYDRUS-1D program. Finally, in the course of the above comparison, we realized that the two algorithms could actually compliment each other in a way that an initial run with ACO–IM could identify the general location of the initial parameter estimates, which can then be fed to LM to obtain a final solution.

5. Conclusions

We introduced a new algorithm for inverse parameter estimation using an optimization procedure inspired by ant colony foraging behavior. The results presented in this paper are very encouraging for ACO–IM method. In the first test, using a well-defined problem, the exact parameters were identified. While in the second test, a comparison of the ACO–IM and the popular LM algorithm produced practically identical simulation results.

Based on the elementary behavior of an individual ant, the greedy force combined with the autocatalytic process would lead to sub-optimal solutions. However, when an entire colony operates, complicated dynamics are exhibited through distributed computations where greed gives the right signals to the autocatalytic process to converge to an optimum solution. Introduction of the parameters A, N, and T achieved a balance between different dynamics that enables ants to jump over local minima.

Despite the encouraging results obtained in this first study, application of the ant optimization technique to environmental parameter estimation problems is still at its infancy. The ACO–IM system could benefit from improved computational efficiency, as well as a more robust *ANT* scoring strategy. This is due to the fact that the values of N and T through c_N and c_T coefficients seemed to vary slightly for different cases studied. Future application of the ACO may need to address these issues in more detail.

Appendix A

Parameters A, N, and T in Eq. (8) control the relative importance of the trail, sensitivity, and critical value of the objective function. For convenience and in order to have less parameters to determine, A was set equal to 1, N was expressed in terms of the variability of the trail received by each stratum, and T was expressed in terms of the variability of the objective function as follows:

$$N = c_{\mathrm{N}} \frac{\sigma_{ij}}{\mu_{ij}},$$

$$T = g_{\min} + c_T \frac{\sigma_g}{\mu_g},$$

where μ_{ij} is the mean of trails placed on the β_{ij} strata, σ_{ij} is the standard deviation of trail on the β_{ij} strata, μ_g and σ_g are, respectively, the mean and the standard deviation of the objective function for an iteration, and c_N and c_T are constants. Note that $g_{\min}$ is the smallest value that T can obtain. For large variability in the objective function, the value of T should be larger in order to have a better chance of avoiding local minima (refer to Fig. 2).

To determine the values of c_N and c_T we used three different models and tested three different problems. Full details of the procedure can be found in Lerch [16]. The three models used were HYDRUS_2D [27], HYDRUS-1D [26], and MACRO [12]. The first test was similar to the example 1 in this manuscript with the objective function written in terms of the water content and pressure head. In this test we assumed the parameters saturated hydraulic conductivity (K_s), and van Genuchten shape factors α, and n of the first layer were unknown. For the second test we used the program HYDRUS-1D and simulated the breakthrough of a boron (H_3BO_4) pulse through Glendale clay loam [30]. In this example we treated the parameters saturated hydraulic conductivity (K_s), adsorption coefficient (k_s), and fraction of exchange sites in equilibrium with solution concentration (f) as unknown and minimized an objective function based on the breakthrough

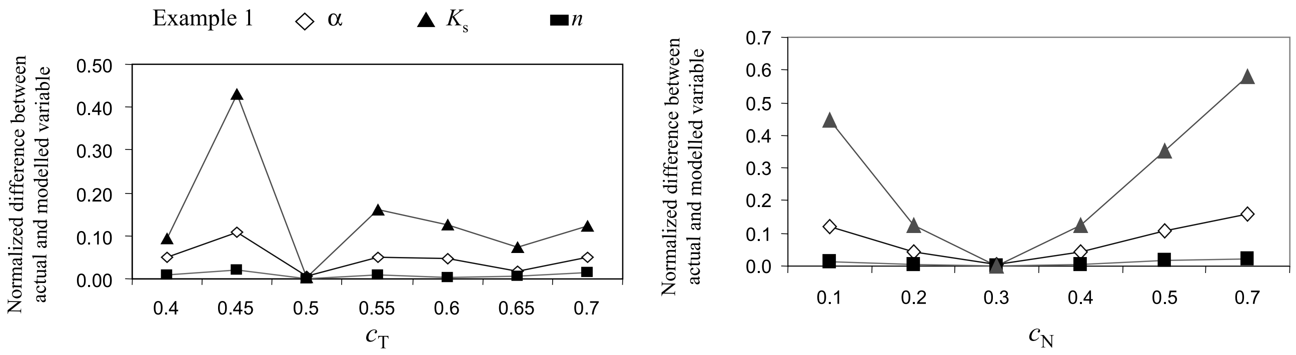

Fig. 14. Normalized difference between actual and modelled variables as a function of c_T and c_N. Typical run showing $c_T = 0.5$ and $c_N = 0.3$ to be the best values.

concentration of boron. The third example, simulated by MACRO [12], dealt with the break through of bromide from an unsaturated soil column. In this example we estimated boundary hydraulic conductivity (*KSM*), pore size distribution index (*ZLAMB*), and tortuosity factor (*ZN*) and minimized an objective function based on the breakthrough concentration of bromide.

In each example, we first ran the programs with a set of known parameters, noted the desired simulated variable as 'observation', assumed an uncertain range for the parameters we wished to treat as 'unknown', and determined the sensitivity of c_T and c_N in a classical way by keeping one of them constant and varying the other. After some preliminary runs, we tested values: $c_T \in \{0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7\}$, and $c_N \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.7\}$. For each case we determined the 'unknown' parameters by running ACO–IM algorithm. Our objective was to obtain a set of c_T and c_N parameters that would in each case lead to parameters closest to the real parameters.

Fig. 14 illustrates an example of the results obtained for the first test. This Figure shows the normalized differences between the estimated and actual parameters by ACO–IM versus different values of c_T and c_N. With small variations, in each case the best values of c_T and c_N were centered around 0.5 and 0.3, respectively. The above values were used for examples one and two in the manuscript. However, we feel because of the small deviations in c_T and c_N in the three examples considered, more work may be needed to identify a better formulation for A, N, and T and a more robust estimation of c_T and c_N parameters.

References

[1] Abbaspour KC, Kasteel R, Schulin R. Inverse parameter estimation in a layered unsaturated field soil. Soil Sci 2000;165(2): 109–5123.

[2] Abbaspour KC, Sonnleitner MA, Schulin R. Uncertainty in estimation of soil hydraulic parameters by inverse modeling: example lysimeter experiments. Soil Sci Soc Am J 1999;63(3):501–9.

[3] Abbaspour KC, van Genuchten MTh, Schulin R, Schläppi E. A sequential uncertainty domain inverse procedure for estimating subsurface flow and transport parameters. Water Resour Res 1997;33(8):1879–92.

[4] Beckers R, Deneubourg JL, Gross S. Trails and U-turns in the selection of the shortest path by the ant Lasius niger. J Theor Biol 1992;159:397–415.

[5] Bonabeau E, Theraulaz G, Deneubourg JL, Aron S, Camazine S. Self-organization in social insects. Trends Ecol Evol 1997;12: 188–93.

[6] Clausnitzer V, Hopmans JW. Non-linear parameter estimation: LM_OPT. General-purpose optimisation code based on Levenberg–Marquardt algorithm. Land Air and Water Resources Paper No. 100032, University of California Davis, CA, USA, 1995.

[7] Colorni A, Doringo M, Maniezzo V, Trubian M. Ant system for job-shop scheduling. Belg J Oper Res, Stat Comput Sci 1993;34:39–53.

[8] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern – Part B 1996;26(1):29–41.

[9] Dorigo M. Optimization learning and natural algorithms. PhD Thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy, 1992.

[10] Eching SO, Hopmans JW. Optimization of hydraulic functions from transient outflow and soil water pressure head data. Soil Sci Soc Am J 1993;57:1167–75.

[11] Gambardella LM, Doringo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In: Prieditis A, Russell S, editors. Proceedings of ML-95, 12th International Conference on Machine Learning, 1995; Morgan Kaufmann, Sanfransisco, CA. p. 252–60.

[12] Jarvis NJ. The MACRO model: Technical description and sample simulations. Reports and Dissertation 19, Department of Soil Science, SLU, Uppsala, Sweden, 1994.

[13] Kool JB, Parker JC. Analysis of the inverse problem for transient unsaturated flow. Water Resour Res 1988;24(6):817–30.

[14] Kool JB, Parker JC, van Genuchten MTh. Parameter estimation for unsaturated flow and transport models – a review. J Hydrol 1987;91:255–93.

[15] Lawler EL, Lenstra JK, Rinnooy-Kan AHG, Shmoys DB, editors. The traveling salesman problem. New York: Wiley; 1985.

[16] Lerch M. Determination and sensitivity analyses of ant colony optimization – inverse modeling parameters. Diploma Thesis, Swiss Federal Institute of Technology, Zurich, 2001.

[17] Marquardt DW. An algorithm for least-squares estimation of non-linear parameters. SIAM J Appl Math 1963;11:431–41.

[18] Mualem Y. A new model for predicting the hydraulic conductivity of unsaturated porous media. Water Resour Res 1976;12(3): 513–22.

[19] Murtagh BA, Saunders MA. MINOS/AUGMENTED user's manual. Tech. Rep. 80-4, System Optimization Laboratory, Department of Operational Research, Stanford University, Stanford CA, 1980.

[20] Nakamura M, Kurumatani K. A mathematical model for the foraging of an ant colony and pattern formation of pheromone trail. In: Fundamental theories of deterministic and stochastic models in mathematical biology, 1995; Institute of Statistical Mathematics. p. 120–31.

[21] NRC, Groundwater models, scientific and regulatory applications. Committee on Groundwater Modeling Assessment, Water Science and Technology Board, National Research Council, National Academy Press, Washington, DC, 1990.

[22] Romano N, Santini A. Determining soil hydraulic functions from evaporation experiments by a parameter estimation approach: experimental verifications and numerical studies. Water Resour Res 1999;35(11):3343–59.

[23] Ségol G. Classic groundwater simulations; proving and improving numerical models. Englewood Cliffs, NJ: Prentice-Hall; 1994.

[24] Sheble GB, Brittig K. Refined genetic algorithm – economic dispatch example. IEEE Trans Power Systems 1995;10(1):117–23.

[25] Simunek J, van Genuchten MTh. Parameter estimation of soil hydraulic properties from multiple tension disc infiltrometer data. Soil Sci 1997;162:383–98.

[26] Simunek J, Huang K, Sejna M, van Genuchten MTh. The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably saturated media, version 1.0. IGWMC – TPS – 70. International Groundwater Modeling Center, Colorado School of Mines, Golden, Colorado, 1998.

[27] Simunek J, Sejna M, van Genuchten MTh. The HYDRUS-2D software package for simulating two-dimensional movement of water, heat, and multiple solutes in variably saturated media, version 2.0. IGWMC – TPS – 53. International Groundwater Modeling Center, Colorado School of Mines, Golden, Colorado, 1999.

[28] Sonnleitner M, Günthardt-Goerg M, Bucher-Wallin I, Attinger W, Reis S, Schulin R. Influence of soil type on the effect of elevated atmospheric CO_2 and N deposition on the water balance and growth of a young spruce and beech forest. Water, Air, Soil Pollut 2001;126:271–90.

[29] van Genuchten MT. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci Soc Am J 1980;44:892–8.

[30] van Genuchten MT. Non-equilibrium transport parameters from miscible displacement experiments. Research Report No. 119, US Salinity Laboratory, Riverside, CA, 1981.

[31] van Genuchten MT, Sudicky EA. Recent advances in vadose zone flow and transport modeling. In: Parlange MB, Hopmans JW, editors. Vadose zone hydrology cutting across disciplines. New York: Oxford University Press; 1999. p. 155–93.

[32] van Genuchten MTh, Leij FJ, Yates SR. The RETC code for quantifying the hydraulic function of unsaturated soils. EPA/600/2-91-065, US EPA, Office of Research and Development, Washington, DC, 1991.