Computational benefits using artificial intelligent methodologies for the solution of an environmental design problem: saltwater intrusion

Maria P. Papadopoulou, Ioannis K. Nikolos and George P. Karatzas

ABSTRACT

Artificial Neural Networks (ANNs) comprise a powerful tool to approximate the complicated behavior and response of physical systems allowing considerable reduction in computation time during time-consuming optimization runs. In this work, a Radial Basis Function Artificial Neural Network (RBFN) is combined with a Differential Evolution (DE) algorithm to solve a water resources management problem, using an optimization procedure. The objective of the optimization scheme is to cover the daily water demand on the coastal aquifer east of the city of Heraklion, Crete, without reducing the subsurface water quality due to seawater intrusion. The RBFN is utilized as an on-line surrogate model to approximate the behavior of the aquifer and to replace some of the costly evaluations of an accurate numerical simulation model which solves the subsurface water flow differential equations. The RBFN is used as a local approximation model in such a way as to maintain the robustness of the DE algorithm. The results of this procedure are compared to the corresponding results obtained by using the Simplex method and by using the DE procedure without the surrogate model. As it is demonstrated, the use of the surrogate model accelerates the convergence of the DE optimization procedure and additionally provides a better solution at the same number of exact evaluations, compared to the original DE algorithm.

Key words | artificial intelligence, computational cost, design, differential evolution, groundwater resources management, neural networks, saltwater intrusion, surrogate model

Maria P. Papadopoulou (corresponding author)
School of Rural and Surveying Engineering,
National Technical University of Athens,
Greece
E-mail: *mpapadop@mail.ntua.gr*

Ioannis K. Nikolos
Department of Production and Management Engineering,
Technical University of Crete,
Greece
E-mail: *jnikolo@dpem.tuc.gr*

George P. Karatzas
Department of Environmental Engineering,
Technical University of Crete,
Greece
E-mail: *karatzas@mred.tuc.gr*

THEORETICAL BACKGROUND

Over the years, the human desire and need to explain complex phenomena (e.g. earthquakes, tsunamis), analyse natural systems (e.g. planet motion) and predict their future behaviour (e.g. weather forecast, aquifer response) have grown significantly in an effort to better understand nature. In the last few decades, scientists were able to provide satisfactory solutions to the majority of the environmental problems that have tremendous impact on everyday life and affect human societies (e.g. gas emission, water contamination, toxic waste disposal) by using advances in science and technology. Recent advances in artificial intelligence and computer science have provided the scientific community with improved tools for predicting and controlling nature.

The theory of evolution and the survival of the fittest concept raised the use of Evolutionary Algorithms (EAs). The EAs theory is based on populations of individuals which evolve according to the principles of natural selection. Even though the study of evolutionary systems by computer scientists goes back to the 1960s, the research community started exploring the use of evolution as an optimization tool for engineering and complex environmental design problems in the 1980s (Mitchell 1999). EAs are a class of search methods with remarkable balance

doi: 10.2166/wst.2010.442

between exploitation of the best solutions and exploration of the search space. They combine elements of directed and stochastic search and are more robust than existing directed search methods (Michalewicz 1999). Also, they may be easily tailored to the specific problem at hand, taking into account its special characteristics by introducing specific operators that increase the convergence rate and the robustness of the procedure. Despite their advantages, EAs are characterized by a considerable computational cost, compared to other stochastic and non-stochastic methodologies. In an EA-based optimization procedure the large computational cost is due to the large number of evaluations of candidate solutions. In most cases, EAs request a large number of generations to converge and a population of candidate solutions is evaluated in each generation. In most engineering problems the evaluation of a candidate solution is a time consuming procedure as it is usually based on the simulation of the physical system using computational physics methodologies. In order to reduce their computational cost several approaches have been proposed for EAs, such as the use of parallel processing, the use of special operators, and the hybridization of EAs with other faster optimization methodologies. As the CPU cost of and EA-based optimization procedure is, more-or-less, proportional to the number of evaluations of candidate solutions, a useful alternative would propose the replacement of some of the costly exact evaluations by less exact and less time consuming ones. The appropriate construction of such "inexpensive" but effective "surrogate models" to replace the "exact" but time consuming computational models (at certain phases of the optimization procedure) can considerably accelerate the computation without sacrificing the accuracy of the optimal solution, as far as the surrogate model accurately represents the physical system, its combination with the EA and the exact physical model results in a robust procedure.

Surrogate models are auxiliary simulations that are less physically faithful, but also less computationally expensive than the expensive simulations described as "exact" and "accurate"; they are either physical models which solve the same problem with more simplistic approximations, or mathematical models which approximate the solution landscape over the search space, obtained from previous runs of the expensive simulation (Torczon & Trosset 1998; Giannakoglou 2002). Such approximations are the low-order polynomials used in Response Surface Methodology (Myers & Montgomery 1995; Shyy *et al.* 2001), the kriging estimates employed in the design and analysis of computer experiments (Ratle 1999), and the various types of Artificial Neural Networks (Giannakoglou 2002). Once the approximation has been constructed, it is typically inexpensive to use. The surrogate models can be used either "off-line" or "on-line". Off-line models are global models, in the sense that a single model covers the entire search space. Such models are built prior to the beginning of the optimization procedure, using available data either from physical experiments, or from successive calls to an "exact" physical model. On-line models are built-into the EA and evolve with its population. They are "re-trained" in each generation of the EA, taking into account the recent information about the search space, provided by the current population. In that sense, they are local approximation models; as a result increased accuracy is expected, compared to the off-line ones.

Environmental management problems as it is reported in the literature, require an extremely time consuming procedure to converge to an optimal solution since a large number of physical system simulations are necessary. Johnson & Rogers (2001) applied soft-computing methodologies involving ANN-Genetic Algorithm and ANN-Simulated Annealing based searches to improve the computational burden and achieve a solution. Both solutions were identical, and compared with the one obtained by the simulator-based searches. Despite the few thousand CPU hours required to complete the simulator-based searches, the resulting best scenarios failed to match the best scenarios uncovered by the ANN-based searches. Also, Kollat & Reed (2007) analysed the impacts of computational complexity using multi-objective evolutionary algorithms (MOEAs) as the dimensionality of the problem increases. Their study showed that in environmental engineering design problems if the decision-maker is willing to accept an approximation to the Pareto optimal solution set the computational scaling of soft-computing methodologies, such as multiple objective evolutionary algorithms, could be linear instead of quadratic as the dimensionality increases. Therefore, during the optimization procedure the use of surrogate models to compute the objective functions,

without any need of using the "exact" physical model is desirable (Nikolos *et al.* 2008). However, the application of surrogate models to evolutionary computation is not a straightforward procedure, as it is difficult to construct a surrogate model that is globally correct. If surrogate models are used for all or most fitness evaluations, it is likely that the evolutionary algorithm prematurely converges to a false optimum which may not even be a local optimal solution. Therefore, it is very essential that the surrogate model is used together with the original function evaluation (Zhang & Sanderson 2009). Many different combinations of surrogate models with various types of single-objective and multi-objective Evolutionary Algorithms have been reported in the literature (Jin *et al.* 2000; Jones 2001; Giannakoglou 2002; Ong *et al.* 2003; Regis & Shoemaker 2004; Runarsson 2004; Jin 2005; Knowles 2006; Zhou *et al.* 2006; Regis & Shoemaker 2007; Nikolos *et al.* 2008; Dhar & Datta 2009; Zhang & Sanderson 2009). The popularity of combining surrogate models with EAs is due to the fact that a considerable acceleration of their convergence rate can be achieved, although they add an extra computational cost to the optimization procedure, related to the construction and updating of the approximation model.

In the field of water resources management, many researchers have used ANNs in combination with other methodologies to simulate complex physical systems as surrogate models. Morshed & Kaluarachchi (1998) investigated the effects of ANN application to simulate the groundwater flow and transport responses in a 1-d unsaturated example problem under various conditions. Their results shown that ANNs have difficulty in predicting the response of dynamic independent variables. Arndt *et al.* (2005) in facility optimization groundwater engineering problems, Nikolos *et al.* (2008) and Dhar & Datta (2009) in saltwater intrusion management of coastal aquifers used the predictions obtained by a ANN to approximate the results of the simulation model in order to significantly reduce the necessary computational time. Yan & Minsker (2006) proposed a methodology in which ANNs were adaptively and automatically trained within a GA to obtain the same reduction by replacing the time-consuming flow and transport simulation model. Garcia & Shigidi (2006) applied ANN to estimate modelling parameters. ANNs provide a data-driven approximation of the poorly known relation between dependent and independent variables. The application of ANN to solve the inverse problem did not provide a unique solution but a set of equally likely distinct inverse solutions. In all the cases the successful application of ANN to solve the inverse problem is depending upon the training process.

Various environmental problems are reported in the literature to be handled using artificial intelligent methodologies such as irrigation planning to maximize the benefits by applying an efficient cropping pattern (Srinivasa Raju & Nagesh Kumar 2004); design of a looped irrigation water distribution networks (Reca & Martinez 2006); operation of water reservoir systems to optimize i) water allocation versus irrigation demand (Azamathulla *et al.* 2008) and ii) operational decisions over time (Reis *et al.* 2006); aquifer remediation in field-scale problems (Rogers *et al.* 1995) and by accounting for uncertainty issues in hydraulic conductivity (Aly & Peralta 1999); rainfall forecasting using hyetograph of recording rain gauges (Nasseri *et al.* 2008); subsurface water level fluctuations by simulating the decreasing trend of groundwater levels in shallow aquifer (Nayak *et al.* 2006) or karstic aquifer (Trichakis *et al.* 2009), and by forecasting empirical hydrological parameters (Daliakopoulos *et al.* 2005) and agricultural pollution to predict pesticide concentration in pre-specified monitoring locations (Sahoo *et al.* 2006).

In this paper, an optimization methodology based on evolution theory that uses a Differential Evolution (DE) algorithm is proposed for the solution of continuous optimization problems in the field of water resources management. The DE algorithm has demonstrated in the past very good performance in solving groundwater resources management problems (Karterakis *et al.* 2007) but slow convergence due to the extremely heavy computational burden. A first attempt of the authors to overcome this computational burden wasthe use of an off-line surrogate model based on a classic fully connected Multilayer Perceptron (MLP) ANN, trained in a supervised manner (Nikolos *et al.* 2008). The use of the off-line surrogate model with the DE optimization procedure allowed for the fast and easy testing of different scenarios of constraints for the same optimization problem, without the need for retraining the ANN model. However, the initial training of the surrogate model was computationally

costly. For the purposes of this analysis a different approach is adopted; an on-line surrogate model is utilized to approximate the objective function, based on a Radial Basis Function ANN. The surrogate model is completely integrated within the DE algorithm to provide a local approximation of the search space, by using the local information from the recently evaluated candidate solutions; the surrogate model is thus retrained in each generation of the DE algorithm. The methodology is applied to the karstified aquifer of the industrial zone of Heraklion in Crete. The complex fracture geomorphology of the physical system has caused serious water quality problems in the aquifer. The water demand in the area is extremely high due to industrial, urban and tourist activities. The goal of the proposed management scheme is to cover the water demand without reducing the subsurface water quality due to saltwater intrusion.

FIELD SITE: DESCRIPTION—SIMULATION MODEL

The area of interest, the industrial zone (BIPE), is located on the coastal aquifer east of the city of Heraklion in Crete. The coastline forms the northern boundary of the area which extends 4,500 m to the south. A preliminary work by Koukadaki *et al.* (2007) provides the geological mapping of the area where fractures and karstified geological formations are present (Figure 1(a)). Various works have been reported in the literature presenting modelling approaches of karstified aquifers as a) equivalent porous medium (Scanlon *et al.* 2003), b) discrete fracture model (Jaquet *et al.* 2004) and dual porosity model (Samardzioska & Popov 2005). The applicability of these models is a question of scale and requires the analysts to determine the effect of the fractures on the flow system. Some researchers have developed numerical models of coastal karstified aquifers using drain networks features to represent the conduit component (Quinn *et al.* 2006; Papadopoulou *et al.* 2010). Even though the aquifer is characterized as karstic, a conventional modelling approach based on Darcy's law was used, since the main fractures were not included in the area of study. For the development of an optimal management strategy, the simulation of the physical system is necessary in order to estimate the response of the system in various pumping scenarios. The domain of interest is vertically discretized into two approximately parallel horizontal layers. A finite element discretization of 1961 finite triangular elements is used within each layer allowing for the accurate representation of the irregular domain. The layers are connected vertically by a finite difference discretization. The groundwater numerical simulator employed in the present work was the Princeton Transport

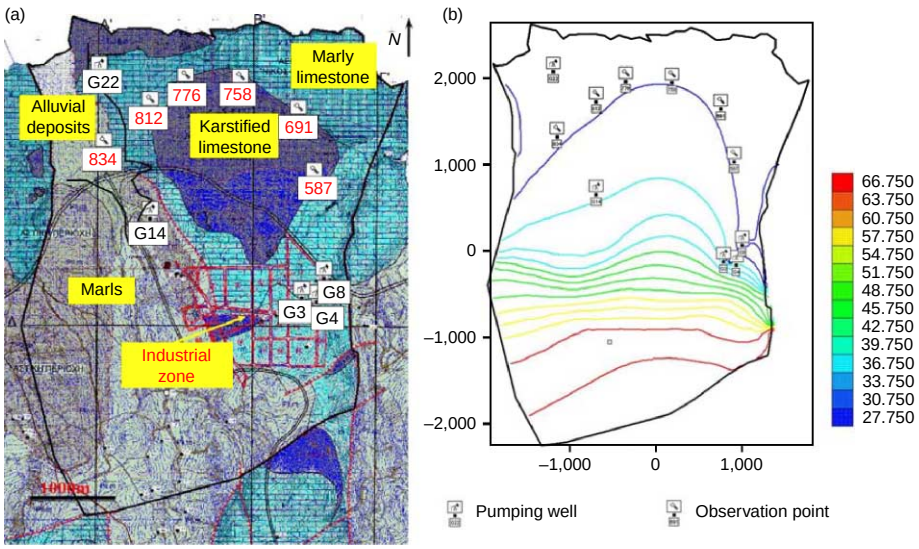

Figure 1 | (a) The geological map of the area (b) Simulated hydraulic head field. Subscribers to the online version of *Water Science and Technology* can access the colour version of this figure from http://www.iwaponline.com/wst.

Code (PTC) which uses a hybrid coupling of the finite element and finite difference methods that enables the application of the splitting procedure (Pinder & Gray 1977). PTC was used in combination with Argus One (Pinder 2002), a geographic information modelling (GIM) environment for entering data into the simulation code. The simulated hydraulic head field with the locations of the pumping wells are shown in Figure 1(b).

The sharp interface theory was used to predict the evolution of the seawater front. According to this theory, the sharp interface between the two immiscible fluids is determined by the difference between the saline and fresh water hydraulic heads and the volume of the fresh water that flows from inland towards the shoreline. As long as the hydraulic gradient of the fresh water in the coastal aquifer is high enough to maintain adequate water flow to the sea, the sharp interface is approximately vertical and located along the shoreline. As the fresh water hydraulic gradient decreases, the sharp interface is becoming less vertical and moves towards the inland in a shape of a water wedge. In this analysis, the seawater is assumed to be stagnant and only horizontal flow of freshwater is considered. The 2-D coastal physical system of interest is modelled using the standard groundwater flow equation for an unconfined aquifer and based on the sharp interface theory the saltwater intrusion front (interface) is hydraulically overestimated based on Ghyben-Herzberg relationship (Equation (1)).

$$\xi = \frac{\rho_f}{\rho_s - \rho_f} h_f \approx 40 h_f \tag{1}$$

where ξ the distance from the sea level to the interface; h_f the hydraulic head of the fresh water above the sea level; ρ_s, ρ_f the densities of saline and fresh water.

The hydrological year was divided into two stress periods (summer–winter) with different precipitation rates for each period. The unstable conditions in the coastal aquifer of interest and the transitional phenomena in the subsurface flow necessitate a simulation period of two years to reach steady state conditions. This time period was defined after the model calibration. For the purposes of this analysis, representative pumping well locations were selected to cover the industrial (G3, G4, G8), urban (G14) and tourist (G22) demand.

OPTIMAL MANAGEMENT DESIGN: PROBLEM FORMULATION

The objective of the optimization scheme is to cover the daily water demand, especially during the summer season when the required water quantities are high due to tourist activities, without reducing the subsurface water quality. The sudden descent of the water table along the coast results in saltwater intrusion phenomena. The mathematical formulation of the management problem is to maximize the water volume extracted from the five pre-selected pumping locations (Equation (2)). The constraints that ensure no further intrusion of the seawater front were imposed at six pre-selected observation locations (Figure 1). According to the sharp interface theory, adopted in this analysis to estimate the evolution of the saltwater intrusion front, the simulated hydraulic head at each observation location should be greater than or equal to 30.75 m at the end of the two year management period (Equation (3)). The value of 30.75 m was obtained from the Ghyben-Herzberg relationship (Ghyben 1888; Herzberg 1901) for an aquifer depth along the shore line equal to 30 m. The maximum extraction rate for each pumping well should not exceed its current(maximum) pumping rate (Equations (4a–e)). The current maximum pumping rates were considered as the upper bound of the pumping rates. The current total maximum pumping rate for all pumping wells is equal to 6,816 m^3/d. The pumping rates are in m^3/d.

$$\max \sum_{i=1}^{5} q_i \tag{2}$$

Subject to

$$h_j \geq 30.75 \quad j = 1, \ldots, 6 \tag{3}$$

$$G8 : 0 \leq q_1 \leq 1,920 \tag{4a}$$

$$G4 : 0 \leq q_2 \leq 1,920 \tag{4b}$$

$$G3 : 0 \leq q_3 \leq 1,920 \tag{4c}$$

$$G22 : 0 \leq q_4 \leq 576 \tag{4d}$$

$$G14 : 0 \leq q_5 \leq 480 \tag{4e}$$

For the DE optimization procedure, the hydraulic head constraints were taken into account through a penalty

function formulation. The constrained maximization problem of Equations (2) and (3) was modified to an unconstrained minimization one, by implementing penalty functions for the implicit constraints, as:

$$\min f, f = \frac{7,000 - \sum_{i=1}^{5} q_i}{7,000} + \left(\sum_{j=1}^{6} g_j\right) + P_1 \tag{5}$$

$$g_j = \begin{cases} |h_j - h_{ref,j}| & \text{if } h_j < h_{ref,j} \\ 0 & \text{otherwise} \end{cases} \quad P_1 = \begin{cases} 1 & \text{if } \sum_{j=1}^{6} g_j \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where h_j is the predicted fresh water hydraulic head at the j-th observation well and $h_{ref,j}$ is the lowest acceptable fresh water hydraulic head at the same location. The value 7,000 in the first term of f (Equation (5)) is used in order this term to tend to a small number as the total pump rate reaches the maximum allowed value (6,816 m^3/d). The first term is devided by 7,000 in order to normalize this term, so that all three terms of the cost function f have the same order of magnitude. In Equation (5), P_1 is an additional penalty term which ensures that the feasible solutions have always better fittness than the infeasible ones. This term takes a non-zero value only when the constraints are violated and drives the optimization procedure towards the feasible area (in addition to the second term). In Equation (5) the relation between q_i and h_j is not explicit and is established through the solution of the flow equation using the PTC computational procedure (when PTC is called to compute "exactly" the cost function f during the optimization procedure). In the cases that the surrogate model is called by the optimization procedure, the cost function f is directly estimated by the RBFN model and the h_j values are not computed.

ARTIFICIAL INTELLIGENCE APPROACH

Differential evolution algorithm

In this work, Differential Evolution (DE) (Storn & Price 1995; Price *et al.* 2005) is used as the optimization tool. The DE algorithm, being a recent development in the field of optimization algorithms, represents a type of Evolutionary Strategy which can deal effectively with continuous optimization problems often encountered in engineering design. The standard DE algorithm uses a fixed size randomly initialized population. At each iteration (generation), a new population is produced; each element of the population (chromosome) can be replaced with a newly generated one, which is the result of successive mutation and crossover operators. The mutation operator performs a linear combination between a randomly selected element (the donor) and the difference between two other randomly selected elements of the current population. The crossover operator performs a modified random crossover between the current member of the population and the result of the mutation operator. The population for the next generation is selected from the current population and the final candidates. If a candidate vector is better fitted than the corresponding current one, the new vector replaces the vector with the one that was compared. A detailed description of the algorithm can be found in Nikolos *et al.* (2007).

Radial basis function network for DE assistance

Although DE has been demonstrated to be one of the most efficient and effective EAs, its convergence rate is still far from ideal, especially when it is applied in optimization problems with time-consuming objective functions. In the present analysis, an approximation model is used for the objective function in order to enhance the convergence rate of the DE algorithm; the model is based on a Radial Basis Function Artificial Neural Network (RBFN). In general, a RBFN (Figure 2) is a three layer, fully connected feed-forward network which performs a nonlinear mapping from the input space to the hidden space, followed by a linear mapping from the hidden to the output space (L is the number of input nodes, M is the number of hidden nodes, while the output layer has a single node). The output $y(x)$ for an input vector $x = [x_1, x_2, \ldots, x_L]$ is given by

$$y(x) = \sum_{i=1}^{M} w_i \cdot \varphi_i(x) \tag{6}$$

where $\varphi(x)$ is the output of the ith hidden unit:

$$\varphi_i(x) = G(\|x - c_i\|), \quad i = 1, \ldots, M \tag{7}$$

The connections (weights) to the output unit (w_i, $i = 1, \ldots, M$) are the only adjustable parameters. The RBFN

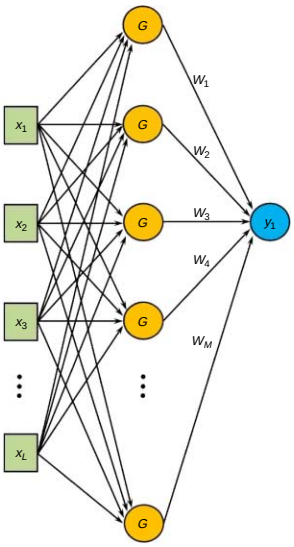

Figure 2 | A typical radial basis function ANN.

centres in the hidden units c_i, $i = 1, \ldots, M$ are selected in a way to maximize the generalization properties of the network. The nonlinear activation function G is chosen to be the Gaussian radial basis function

$$G(u, \sigma) = \exp\left(-u^2/\sigma^2\right) \tag{8}$$

where σ is the standard deviation of the basis function.

The correct selection of RBFN centres is very important for the predictive and generalization capabilities of the RBFN. The standard process is to select the input vectors in the training set as centres. In this case, $M = NR$, where NR is the number of training data. However, for large training sets (resulting in large values for M), this choice is expected to increase storage requirements and CPU cost and could also lead to over-fitting and/or bad generalization of the network. The solution to these problems is to select $M < NR$ and, consequently, to search for sub-optimal solutions which will provide a better generalizing capability to the RBFN.

The direct learning process used here is based on a matrix formulation of the governing equations of the RBFN. The presentation of the network with the NR input patterns allows the formulation of an $(NR \times M)$ matrix H, which becomes square in the special case where $NR = M$. Each line in the interpolation matrix H corresponds to a learning example and each column to a RBFN centre. The output unit values result in the form of the matrix product:

$$H(NR \times M)w(M \times 1) = y(NR \times 1) \tag{9}$$

where y is the desired output vector as provided by the training data set, and w is the synaptic weights vector, which consists of M unknowns to be computed. Matrix H is inverted by using the Gram-Schmidt technique.

Matrix inversion and, consequently, the computation of RBFN weights, is a very fast procedure, as far as the number of input patterns NR is kept sufficiently small. In the algorithm applied in this work NR was set equal to 80, while M was set at 2/3 of NR (equal to 53). For local approximation models, as the one used in this analysis, the number of input patterns can be retained small without sacrificing the accuracy of the network. Actually, the computational cost of training the RBFN is much lower than a single evaluation of a candidate solution using the exact physical model. As a result, the retraining of the RBFN model in each generation of the DE algorithm adds negligible computational cost to the complete optimization procedure, compared to the large benefit due to the reduction of exact evaluations. In order to evaluate the accuracy of the approximation model the training and testing errors are stored in each generation. The testing error in each generation is computed for 30 inputs taken randomly from the database, among its best individuals. Those inputs are different than the NR inputs used for the training of the network. The database is enriched in each generation with the currently evaluated individuals.

In each DE generation, during the evaluation procedure, each trial vector must be evaluated and then compared with the corresponding current vector, in order to select the better-fitted between them to pass to the next generation. The concept of the surrogate model is to replace the costly exact evaluations of trial vectors (using the PTC model) with fast inexact approximations (using the RBFN model) without sacrificing the robustness of the DE algorithm. During the evaluation phase, each trial vector is pre-evaluated by using the RBFN model. If it is pre-evaluated as lower-fitted (higher objective function in minimization problems) than the corresponding vector of the current population, then no further exact evaluation is needed and the current vector is transferred to the next generation. If the trial vector is

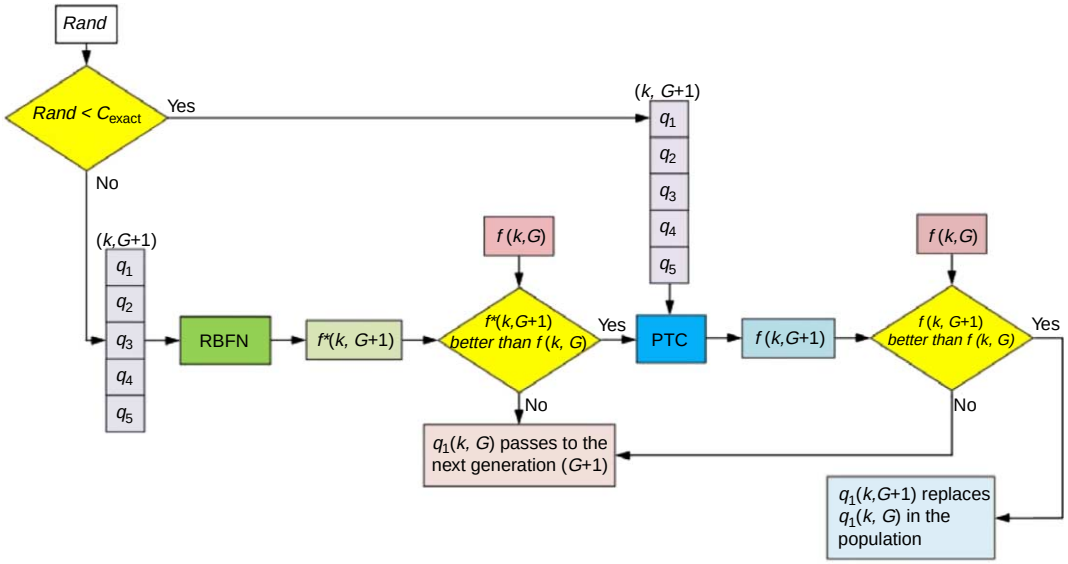

Figure 3 | The modified evaluation and selection procedure, including the use of the RBFN surrogate model.

pre-evaluated as better fitted than the corresponding current vector, then an exact re-evaluation is performed, followed by a new comparison between the two vectors. If the trial vector is re-evaluated as better-fitted as the current one, then the trial vector passes to the next generation. Otherwise, the current vector is the one that will pass to the next generation. Additionally, a small percentage (C_{exact}) of the candidate solutions are selected with uniform probability to be exactly evaluated, without accounting their performance provided by the approximation model. The corresponding procedure is illustrated in Figure 3.

In the first two generations of the DE algorithm, all vectors are exactly evaluated. The complete DE/RBFN algorithm is illustrated in Figure 4. In each generation, the

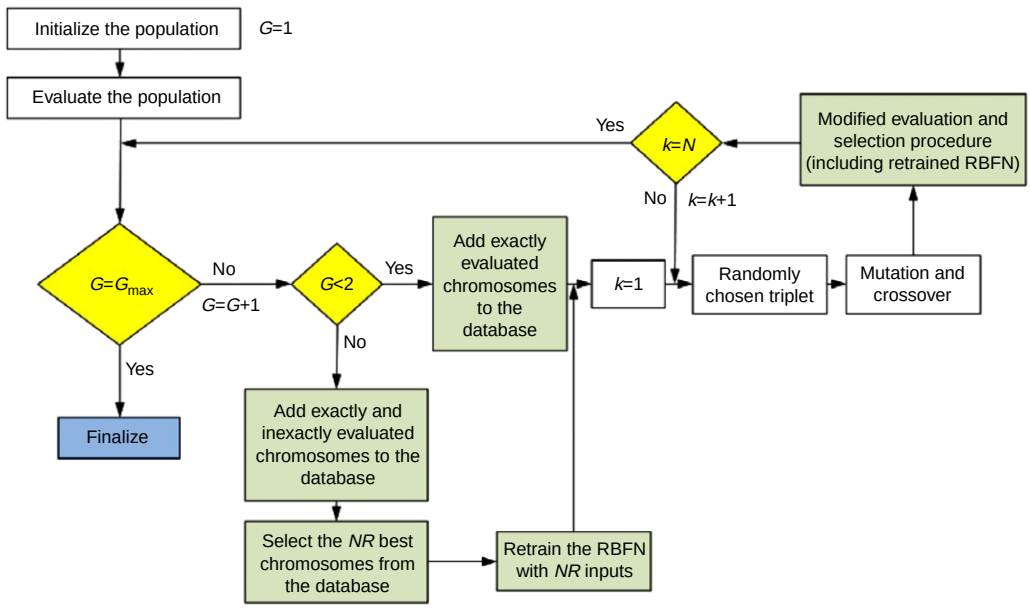

Figure 4 | The complete DE/RBFN algorithm.

RBFN is retrained to provide a local approximation model. In Figures 3 and 4, G is the current generation number, G_{max} is the maximum number of generations, k is the number of the individual in the population ($k = 1, \ldots, N$), N is the population size, f^* is the approximation to the cost function computed by the RBFN model, and f is the cost function (Equations (4a–e)). According to the aforementioned procedure, only exactly evaluated trial vectors have the opportunity to pass to the new generation, so the current population always comprises exactly evaluated individuals. In this way, one part of the comparison (the current vector) is always an exact-evaluated vector, which enhances the robustness of the procedure.

The result of each evaluation (exact or inexact), along with the corresponding chromosome, are stored in a database. In order to have a local approximation model, only the best-fitted individuals of the database entries are used in each generation to re-train the RBFN. In this way the approximation model evolves with the population and employs only the useful information for approximating the objective function. The surrogate model predictions replace exact and costly evaluations only for the less promising individuals, while the more promising ones are always exactly evaluated.

There are several reasons why one should choose RBFN as the approximation model; Haykin (1999) offers comparative remarks for RBFNs and Multilayer Perceptron (MLP) ANNs. However, the main reason for choosing RBFN here is its compatibility with the adopted local approximation strategy. The use of relatively small numbers of training patterns (i.e. small networks), helps creating local range RBFNs with, consequently, small approximation error. Additionally, it allows the inversion of matrix H using almost negligible CPU time. It should be kept in mind that the computational cost associated with the use of an ANN is the cost of training the network, while the use of an already trained ANN to evaluate a new input adds negligible computational cost.

A more detailed description of the proposed procedure can be found in (Nikolos *et al.* 2007) where it has been demonstrated that the RBFN can considerably reduce the DE computation time and provide deeper convergence to the optimization procedure.

Linear programming methodology: simplex method

For comparison purposes, the described management problem was linearized (piecewise) and solved using the MATLAB 7.1 Simplex Method algorithm following a mathematical transformation of the original management problem. A detailed description of the methodology was presented in Karterakis *et al.* (2007). The Simplex Method algorithm converged to an optimal solution after 3 steps (Table 1). The need of only three iterations to converge near the optimal solution is attributed to the smooth shape of the cost function, which allows for the local linearization of the non-linear problem. The target of using the Simplex Method was not to compare the DE/RBFN with this one, but to use the Simplex optimal solution as a reference one, in order to test the ability of the DE and DE/RBFN algorithms to converge near the global optimum.

The DE/RBFN algorithm was applied for 200 generations with a population size equal to 20. Parameters F and C_r (Storn & Price 1995; Price *et al.* 2005; Nikolos *et al.* 2007) of the DE algorithm were set equal to 0.9 and 0.65, respectively. The optimal solution of the DE/RBFN algorithm and the corresponding solution of the SIMPLEX method are presented in Table 1. As can be seen, all tested optimization methodologies converged to practically the same optimal solution with the DE/RBFN algorithm providing a slightly better result. Regarding computation time, the SIMPLEX method required minimal computation time, as only three iterations of the method were needed to converge to the final solution. However, a continuous interaction with the user is needed to separately run the PTC simulation model in order to compute the response matrices for each sequential run of the SIMPLEX method. Although the DE/RBFN method takes considerable

Table 1 | Comparison of the optimal solutions

Well ID	Optimal solution SIMPLEX (m^3/day)	Optimal solution DE/RBFN (m^3/day)	Optimal solution DE (m^3/day)
G8	1,920.0	1,920.0	1,920.0
G4	534.5	535.18	526.05
G3	0.0	0.0	7.69
G22	576.0	576.0	576.0
G14	0.0	0.0	0.0
Σq_i	3,030.5	3,031.18	3,029.74

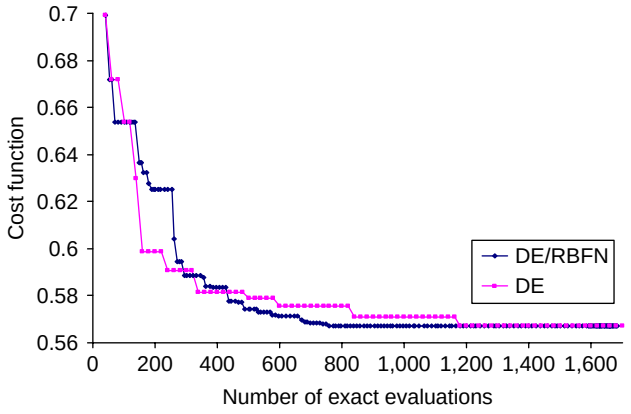

Figure 5 | Convergence histories for the DE and DE/RBFN runs.

computation time (approximately 14 hours on an AMD Athlon 64 × 2 2.01 GHz machine), the computation is completely automatic and can be performed overnight, without any interaction with the user.

The use of the RBFN surrogate model enabled the speed-up of convergence, without sacrificing the robustness of the DE algorithm, as the combined algorithm managed to converge to practically the same solution with the SIMPLEX method, providing a slightly better result. For the 200 generations, corresponding to 200 × 20 = 4,000 evaluations of candidate solutions, only 1,682 exact evaluations were performed, the rest evaluated only with the surrogate model. Additionally, the algorithm converged to the final solution practically at generation 107 (approximately after 1,050 exact evaluations), which implies that the computation time could be much less. In order to demonstrate the acceleration capabilities of the RBFN surrogate model, the same problem was run with the DE algorithm without the surrogate model. The DE algorithm was run for 85 generations in order to ensure that the number of exact evaluations between the two methods (and, hence, the computation time) is approximately the same. The convergence histories for the two methods are compared in Figure 5. The horizontal axis represents the number of exactly evaluated solutions, which is a measure of the computational time, as the exact evaluations by the PTC code is by far the most time-consuming part of the computational procedure. As shown, although the DE/RBFN method demonstrates slower convergence in the beginning, it converges faster towards the optimal solution and provides a better solution for the same number of exact evaluations.

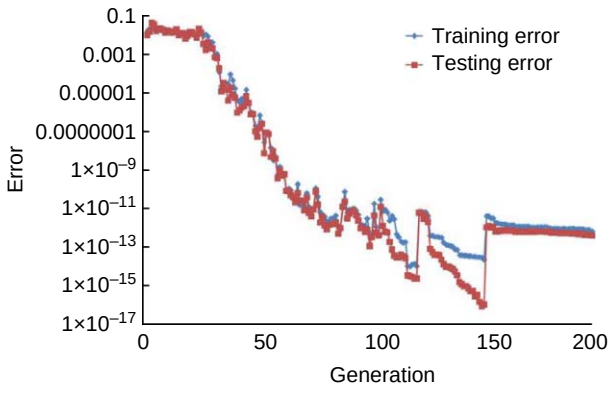

Figure 6 | History of training and testing errors of the RBFN model.

Figure 6 contains the histories of the training and testing errors of the RBFN model, which is retrained in each generation. The training of the network becomes better as the algorithm converges towards the optimal solution, which demonstrates the local character of the approximation model. The computed errors (training and testing) are very good, especially at the last generations, reflecting a very good approximation of the cost function.

DISCUSSION AND CONCLUSIONS

A combination of a Differential Evolution algorithm and a Radial Basis Function Artificial Neural Network was utilized to solve an environmental water resources management problem. The RBFN was used as a surrogate model to the costly numerical model of the physical system to accelerate the optimization procedure. The application of such an on-line surrogate model to the evolutionary optimization procedure is not straightforward and automatic; it is rather difficult (or even impossible) to construct a globally correct surrogate model. In the case of a globally correct surrogate model no retraining of the model would be needed. However, in most cases (especially when the function that is simulated by the surrogate model is very irregular) if a global surrogate model is used without retraining and without using some exact function evaluations is likely that the optimization algorithm prematurely converges to a false solution. In this paper the surrogate model is a local one, while is retrained in each generation of the DE algorithm, using as training data the best individuals contained in the database, which is enriched with the

currently evaluated chromosomes. The use of the best individuals in the data base to retrain the network, rather than the individuals of the last population, was adopted in order to provide a better approximation of the most promising areas, which are actually the target of the optimization procedure. The adoption of a RBFN as a surrogate model was based on the excellent approximation capabilities of this type of networks, which in general perform better as local approximation models than the classic Multi Layer Perceptorn networks. These capabilities were demonstrated here by the very small training and testing errors of the RBFN model, especially at large generation numbers when the algorithm converges towards the optimal solution. The use of an RBFN as a surrogate model enabled the replacement of some of the costly evaluations of the accurate PTC model, resulting in a speedup of the DE convergence, without sacrificing its robustness; the DE/RBFN model not only converges faster towards the optimal solution but also provides a more accurate approximation of the global optimum.

The Simplex Method was used to provide a reference solution in order to test the ability of the DE and DE/RBFN algorithms to converge near the global optimum. The DE/RBFN method converged to a solution practically the same (and slightly better) than the one provided by the SIMPLEX method, while the DE method without the use of a surrogate model converged to a slightly worse solution for the same number of exact evaluations. Although the computation time for the DE/RBFN method is considerably larger than that for the SIMPLEX one, the former is a completely automated procedure with no need for interaction with the user, which is not the case for the SIMPLEX method. Authors' intention is to evaluate the proposed optimization procedure in more difficult environmental design problems in the near future, with irregular solution spaces, in order to test the ability of the surrogate model to drive the population towards the global optimum.

ACKNOWLEDGEMENTS

The authors thank the three anonymous reviewers for the insightful comments that considerably improved the manuscript.

REFERENCES

Aly, A. H. & Peralta, R. C. 1999 Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. *Water Resour. Res.* **35**(8), 2523–2532.

Arndt, O., Barth, T., Freisleben, B. & Grauer, M. 2005 Approximating a finite element model by neural network prediction for facility optimization in groundwater engineering. *Eur. J. Oper. Res.* **166**, 769–781.

Azamathulla, H. Md., Wu, F.-C., Ghani, A. A., Narulkar, S. M., Zakaria, N. A. & Chang, C. K. 2008 Comparison between genetic algorithm and linear programming approach for real time operation. *J. Hydro Environ. Res.* **2**, 172–181.

Daliakopoulos, I. N., Coulibaly, P. & Tsanis, I. K. 2005 Groundwater level forecasting using artificial neural networks. *J. Hydrol.* **309**, 229–240.

Dhar, A. & Datta, B. 2009 Saltwater intrusion management of coastal aquifers. I: linked simulation-optimization. *J. Hydrologic Eng.* **14**(12), 1263–1272.

Garcia, L. A. & Shigidi, A. 2006 Using neural networks for parameters estimation in groundwater. *J. Hydrol.* **318**, 215–231.

Ghyben, W. B. 1888 Nota in verband met de voorgenomen putboring nabij Amsterdam. *Tijdschrift van het Kononklijk Instituut van Ingenieurs* **9**, 8–22.

Giannakoglou, K. C. 2002 Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Prog. Aerosp. Sci.* **38**, 43–76.

Haykin, S. 1999 *Neural Networks, a Comprehensive Foundation*, 2nd edition. Prentice Hall.

Herzberg, A. 1901 Die Wasserversorgung einiger Nordseebader. *J. Gasbeleucht Wasserversorg* **44**, 815–819.

Jaquet, O., Siegel, P., Klubertanz, G. & Benabderrhamane, H. 2004 Stochastic discrete model of karstic networks. *Adv. Water Resour.* **27**, 751–760.

Jin, Y. 2005 A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput. Fusion Found. Methodol. Appl.* **9**(1), 3–12.

Jin, Y., Olhofer, M. & Sendhoff, B. 2000 On evolutionary optimization with approximate fitness functions, *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, San Francisco, pp. 786–792.

Jones, D. R. 2001 A taxonomy of global optimization methods based on response surface. *J. Glob. Optim.* **21**(4), 345–383.

Johnson, V. M. & Rogers, L. L. 2001 Applying soft computing methods to improve the computational tractability of a subsurface simulation-optimization problem. *J. Pet. Sci. Eng.* **29**(3–4), 153–175.

Karterakis, S., Karatzas, G. P., Nikolos, I. K. & Papadopoulou, M. P. 2007 Application of linear programming and differential evolutionary optimization methodologies for the solution of coastal subsurface water management problems subject to environmental criteria. *J. Hydrol.* **342**(3–4), 270–282.

Knowles, J. 2006 ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **10**(1), 50–66.

Kollat, J. B. & Reed, P. M. 2007 A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. *Adv. Water Resour.* **30**(3), 408–419.

Koukadaki, M. A., Karatzas, G. P., Papadopoulou, M. P. & Vafidis, A. 2007 Identification of the saline zone in a coastal aquifer using electrical tomography data and simulation. *Water Resour. Manage.* **21**(11), 1881–1898.

Michalewicz, Z. 1999 *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York.

Mitchell, M. 1999 *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.

Morshed, J. & Kaluarachchi, J. 1998 Application of artificial neural networks and genetic algorithm in flow and transport simulations. *Adv. Water Resour.* **22**(2), 145–158.

Myers, R. H. & Montgomery, D. C. 1995 *Response Surface Methodology: Progress and Product in Optimization Using Designed Experiments*. Wiley–Interscience, New York.

Nasseri, M., Asghari, K. & Abedini, M. J. 2008 Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network. *Expert Syst. Appl.* **35**, 1415–1421.

Nayak, P. C., Satyajirao, Y. R. & Sudheer, K. P. 2006 Groundwater level forecasting in a shallow aquifer using artificial neural network approach. *Water Resour. Manage.* **20**, 77–90.

Nikolos, I. K., Zografos, E. S. & Brintaki, A. N. 2007 UAV path planning using evolutionary algorithms. In: Chahl, J. S., Jain, L. C., Mizutani, A. & Sato-Ilic, M. (eds) *Innovations in Intelligent Machines–1*, *Studies in Computational Intelligence–SCI*, (Vol. 70). Springer, Berlin, Heidelberg, pp. 77–111.

Nikolos, I. K., Stergiadi, M., Papadopoulou, M. P. & Karatzas, G. P. 2008 Artificial neural networks as an alternative approach to groundwater numerical modeling and environmental design. *Hydrological Process.* **22**(17), 3337–3348.

Ong, Y. S., Nair, P. B. & Keane, A. J. 2003 Evolutionary optimization of computationally expensive problems via surrogate modeling. *Am. Inst. Aeronaut. Astronaut. J.* **41**, 687–696.

Papadopoulou, M. P., Varouchakis, E. A. & Karatzas, G. P. 2010 Terrain discontinuities effects in the regional flow of a complex karstified aquifer. *Environ. Model. Assess.* **15**, 319–328.

Pinder, G. F. 2002 *Groundwater Modelling Using Geographical Information Systems*. John Willey & Sons Inc, New York.

Pinder, G. F. & Gray, W. G. 1977 *Finite Element Simulation in Surface and Subsurface Hydrology*. Academic Press, New York.

Price, K. V., Storn, R. M. & Lampinen, J. A. 2005 *Differential Evolution, a Practical Approach to Global Optimization*. Springer-Verlag, Berlin, Heidelberg.

Quinn, J. J., Tomasko, D. & Kuiper, J. A. 2006 Modeling complex flow in a karst aquifer. *Sediment. Geol.* **184**, 343–351.

Ratle, A. 1999 Optimal sampling strategies for learning a fitness model. *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, Washington, DC, USA.

Reca, J. & Martinez, J. 2006 Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resour. Res.* **42**, W05416, doi:10.1029/2005WR004383.

Regis, R. G. & Shoemaker, C. A. 2004 Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Trans. Evol. Comput.* **8**(5), 490–505.

Regis, R. G. & Shoemaker, C. A. 2007 Improved strategies for radial basis function methods for global optimization. *J. Glob. Optimiz.* **37**(1), 113–135.

Reis, L. F. R., Bessler, F. T., Walters, G. A. & Savic, D. 2006 Water supply reservoir operation by combined genetic algorithm-linear programming (GA-LP) approach. *Water Resour. Manage.* **20**, 227–255.

Rogers, L. L., Dowla, F. U. & Johnson, V. M. 1995 Optimal field-scale groundwater remediation using neural networks and the genetic algorithm. *Environ. Sci. Technol.* **29**, 1145–1155.

Runarsson, T. P. 2004 *Proceedings of 8th Parallel Problem Solving From Nature (PPSN VIII)*, *LNCS 3242*. Springer-Verlag, Heidelberg, pp. 401–410.

Sahoo, G. B., Ray, C., Mehnert, E. & Keefer, D. A. 2006 Application of artificial neural networks to assess pesticide contamination in shallow groundwater. *Sci. Total Environ.* **367**(1), 234–251.

Samardzioska, T. & Popov, V. 2005 Numerical comparison of the equivalent continuum, non-homogeneous and dual porosity models for flow and transport in fractured porous media. *Adv. Water Resour.* **28**, 235–255.

Scanlon, B. R., Mace, R. E., Barrett, M. E. & Smith, B. 2003 Can we simulate regional groundwater flow in a karst system using equivalent porous media models? Case study, barton springs edwards aquifer, USA. *J. Hydrol.* **276**, 137–158.

Shyy, W., Papila, N., Vaidynathan, R. & Tucker, K. 2001 Global design optimization for aerodynamics and rocket propulsion components. *Prog. Aerosp. Sci.* **37**, 59–118.

Srinivasa Raju, K. & Nagesh Kumar, D. 2004 Irrigation planning using genetic algorithm. *Water Resour. Manage.* **18**, 163–176.

Storn, R. & Price, K. 1995 DE–a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Space. ICSI, Technical Report TR-95-012.

Torczon, V. & Trosset, M. W. 1998 Using Approximations to Accelerate Engineering Design Optimization, NASA/CR-1998-208460, ICASE Report No. 98-33.

Trichakis, I. C., Nikolos, I. K. & Karatzas, G. P. 2009 Optimal selection of artificial neural network parameters for the prediction of a karstic aquifer's response. *Hydrological Process.* **135**(2), 128–137.

Yan, S. & Minsker, B. 2006 Optimal groundwater remediation design using an adaptive neural network genetic algorithm. *Water Resour. Res.* **42**, W05407.

Zhang, J. & Sanderson, A. C. 2009 *Adaptive Differential Evolution, A Robust Approach to Multimodal Problem Optimization*. Springer-Verlag, Berlin, Heidelberg.

Zhou, Z., Ong, Y. S., Lim, M. H. & Lee, B. S. 2006 Memetic algorithm using multisurrogates for computationally expensive optimization problems. *Soft Comput.* **11**(10), 957–971.