To be cited as: *Giustolisi, O. and Savic, D.A. (2009) Advances In Data-Driven Analyses and Modelling Using EPR-MOGA, J. of Hydroinformatics, Vol. 11, No. 3-4, pp. 225–236.*

ADVANCES IN DATA-DRIVEN ANALYSES AND MODELLING USING EPR-MOGA

O. Giustolisi[1] *and D.A. Savic*[2]

[1] Department of Civil and Environmental Engineering, Technical University of Bari, II Engineering Faculty,
Taranto via Turismo n.8, 74100 Italy, o.giustolisi@poliba.it
[2] Centre for Water Systems, School of Engineering, Computer Science and Mathematics, University of Exeter, Harrison Building,
North Park Road, Exeter EX4 4QF, United Kingdom, d.savic@exeter.ac.uk

Abstracts

Evolutionary Polynomial Regression (EPR) is a recently developed hybrid regression method that combines the best features of conventional numerical regression techniques with the genetic programming/symbolic regression technique. The original version of EPR works with formulae based on true or pseudo-polynomial expressions using a single-objective genetic algorithm. Therefore, to obtain a set of formulae with a variable number of pseudo-polynomial coefficients, the sequential search is performed in the formulae space. This paper presents an improved EPR strategy that uses a multi-objective genetic algorithm instead. The paper demonstrates that multi-objective approach is a more feasible instrument for data analysis and model selection. Moreover, the paper shows that EPR can also allow for simple uncertainty analysis (since it returns polynomial structures that are linear with respect to the estimated coefficients). The methodology is tested and the results are reported in a case study relating groundwater level predictions to total monthly rainfall.

Introduction to EPR

Numerical regression is the most powerful and commonly applied form of regression that provides a solution to the problem of finding the best model to fit the observed data (e.g. fitting a line/curve through a set of points). However, the form of a function (linear, exponential, logarithmic, etc) has to be selected before the fitting commences. On the other hand, genetic programming uses a simple, but very powerful artificial intelligence tactics for computer learning inspired by natural evolution to find the appropriate mathematical model to fit a set of points. The computer produces and evolves a whole population of functional expressions based on how closely each of them fit the data. The automated induction of mathematical models (descriptions) of data using genetic programming (Koza, 1992) is commonly referred to as symbolic regression (Babovic and Keijzer, 2000). Evolutionary Polynomial Regression (EPR) is a recently developed hybrid regression method by Giustolisi and Savic (2006; 2004) that integrates the best features of numerical regression (Draper and Smith, 1998) with genetic programming (Koza, 1992).

EPR Strategy

The following vector form is the base for the development of EPR (Giustolisi and Savic,2006),

$$\mathbf{Y}_{N\times 1} = \left[\mathbf{I}_{N\times 1} \quad \mathbf{Z}^{j}_{N\times m}\right] \times \left[a_0 \quad a_1 \quad \mathrm{K} \quad a_m\right]^T = \mathbf{Z}_{N\times d} \times \boldsymbol{\theta}^T_{d\times 1} \tag{1}$$

where

$\mathbf{Y}_{Nx1}(\boldsymbol{\theta},\mathbf{Z})$ is the least squares estimate vector of *N* target values;

$\boldsymbol{\theta}_{1xd}$ is the vector of *d*=*m*+1 parameters a_j, *j*=1:*m*, and a_0; and

$\mathbf{Z}_{Nxd}$ is a matrix formed by **I**, unitary column vector for bias a_0, and *m* vectors of variables $\mathbf{Z}^j$ that for a fixed *j* are a product of the independent predictor vectors of variables/inputs, **X** = <$\mathbf{X}_1$ $\mathbf{X}_2$ … $\mathbf{X}_k$>.

The key idea of the EPR is to start from Eq. (1) and search first for the best form of the function, i.e. a combination of vectors of independent variables (inputs), $\mathbf{X}_{S=1:k}$, and then to perform least-squares regression to find the adjustable parameters **θ** for each combination of inputs. To avoid the pitfalls of hill-climbing search methodologies, a global search algorithm is implemented for both the best set of input combinations and related exponents simultaneously, according to the user-defined cost function.

The matrix of inputs **X** is given as:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1k} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2k} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3k} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{Nk} \end{bmatrix} = \left[\mathbf{X}_1 \quad \mathbf{X}_2 \quad \mathbf{X}_3 \quad \dots \quad \mathbf{X}_{\mathrm{k}}\right] \tag{2}$$

where the *k*-th column of **X** represents the candidate variables for the *j*-th term of Eq. (1). Therefore, the *j*-th term of Eq. (1) could be written as

$$\mathbf{Z}^{j}_{N\times 1} = \left[\left(\mathbf{X}_1\right)^{\mathbf{ES}(j,1)} \cdot \mathrm{K} \cdot \left(\mathbf{X}_k\right)^{\mathbf{ES}(j,k)}\right] \quad j = 1\mathrm{K}\; m \tag{3}$$

where, $\mathbf{Z}^j$ is the *j*-th column vector whose elements are products of candidate independent inputs and **ES** is a matrix of exponents. Therefore, the problem is to find the matrix $\mathbf{ES}_{m\text{x}k}$ of exponents whose elements can assume values within user-defined bounds.

For example, if a vector of candidate exponents for columns (inputs) in **X** is chosen to be **EX**=[-1, 0,1] and *m*=4 (the number of terms, bias excluded) and *k*=3 (the number of candidate independent variables/inputs), the polynomial regression problem is to find a matrix of exponents $\mathbf{ES}_{4\text{x}3}$. An example of such a matrix is given here

$$\boldsymbol{ES}_{m\times k=4\times 3} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \tag{4}$$

When this matrix is substituted into Eq. (3) the following set of expressions is obtained

$$\begin{aligned} \mathbf{Z}_1 &= \left(\mathbf{X}_1\right)^{-1}\left(\mathbf{X}_2\right)^{0}\left(\mathbf{X}_3\right)^{1} = \mathbf{X}_1^{-1}\mathbf{X}_3 \\ \mathbf{Z}_2 &= \left(\mathbf{X}_1\right)^{0}\left(\mathbf{X}_2\right)^{1}\left(\mathbf{X}_3\right)^{-1} = \mathbf{X}_2\mathbf{X}_3^{-1} \\ \mathbf{Z}_3 &= \left(\mathbf{X}_1\right)^{1}\left(\mathbf{X}_2\right)^{0}\left(\mathbf{X}_3\right)^{0} = \mathbf{X}_1 \\ \mathbf{Z}_4 &= \left(\mathbf{X}_1\right)^{1}\left(\mathbf{X}_2\right)^{1}\left(\mathbf{X}_3\right)^{0} = \mathbf{X}_1\mathbf{X}_2 \end{aligned} \tag{5}$$

Therefore, based on the matrix given in Eq. (4), the expression of Eq. (1) is given as

$$\mathbf{Y} = a_0 + a_1\mathbf{Z}_1 + a_2\mathbf{Z}_2 + a_3\mathbf{Z}_3 + a_4\mathbf{Z}_4 = a_0 + a_1\frac{\mathbf{X}_3}{\mathbf{X}_1} + a_2\frac{\mathbf{X}_2}{\mathbf{X}_3} + a_3\mathbf{X}_1 + a_4\mathbf{X}_1\mathbf{X}_2 \tag{6}$$

The adjustable parameters a_j could now be computed by means of the linear Least Squares (LS) method using the minimisation of the sum of squared errors (SSE) as cost function. Note that each row of **ES** determines the exponents of the candidate variables of *j*-th term in Eq. (1). Each of the exponents in **ES** corresponds to a value from the user-defined vector **EX**. This allows the transformation of the symbolic regression problem into one of finding the best **ES**, i.e., the best structure of the EPR equation, e.g. in Eq. (6).

The global search for the best form of Eq. (6) is performed by means of a standard GA (Holland, 1975, Goldberg, 1989). The GA is an algorithmic model of Darwinian evolution that begins with the creation of a set of solutions referred to as a population of individuals. Parameters being optimised are coded using 'chromosomes', i.e., a set of character strings that are analogous to the chromosomes found in DNA. Standard GAs use a binary alphabet (characters may be 0's or 1's) to form chromosomes. Instead, integer GA coding is used here to determine the location of the candidate exponents of **EX** in the matrix **ES**. For example the positions in **EX**=[-1,0,1] correspond to the following string for the matrix of Eq. (4) and the expression of Eq. (6):

$$[1\ 2\ 3,\ 2\ 3\ 1,\ 3\ 2\ 2,\ 3\ 3\ 2] \tag{7}$$

Additionally, it is clear that the presence of at least one zero in **EX** assures the ability to exclude some of inputs and/or input combinations from the regression equation. The following GA parameters were also used in the current EPR implementation: multiple-point crossover; single point mutation and; termination criterion as a function of the length of the chromosome, the maximum number of polynomial terms *j* and the number of inputs *k* in the matrix **X** (Giustolisi and Savic, 2006).

Least Squares Solution by SVD

Computing a_j in Eq. (6) is an inverse problem that corresponds to solving an over-determined linear system as a LS problem. This problem is traditionally solved by Gaussian elimination. However, an evolutionary search procedure may generate candidate solutions (e.g. a combination of exponents of **X**) that correspond to an ill-conditioned inverse problem. This often means that the rectangular matrix $\mathbf{Z}_{Nxd}$:

$$\mathbf{Z} = \left[\mathbf{I}_{N\times 1} \quad \mathbf{Z}^1_{N\times 1} \quad \mathbf{Z}^2_{N\times 1} \quad \ldots \quad \mathbf{Z}^m_{N\times 1}\right]_{N\times(m+1)=N\times d} \tag{8}$$

may not be of full rank (if a solution contains a column of zeros) or the columns $\mathbf{Z}^j$ are linearly dependent. This could pose serious problems to Gaussian elimination and a more robust solver is therefore needed. Parameter estimation of a_j (or θ) in EPR is performed by means of the Singular Value Decomposition (SVD) of the matrix **Z**. This approach makes the process of finding the solution to the LS problem more robust, although in general the SVD is slower than Gaussian elimination (Golub and Van Loan, 1993). Finally, the Moore-Penrose pseudo-inverse (Golub and Van Loan, 1993) as regularization method can be used (when **Z** is not full rank the solution is that corresponding to the minimum value of the Euclidean norm).

Extension of EPR

EPR allows pseudo-polynomial expressions as in Eq. (1), allowing structures such as

$$\begin{array}{ll}
\hat{\mathbf{Y}} = a_0 + \sum_{j=1}^{m} a_j \left(\mathbf{X}_1\right)^{\mathbf{ES}(j,1)} \cdot \ldots \cdot \left(\mathbf{X}_\mathrm{k}\right)^{\mathbf{ES}(j,\mathrm{k})} f\left(\left(\mathbf{X}_1\right)^{\mathbf{ES}(j,\mathrm{k}+1)}\right) \cdot \ldots \cdot f\left(\left(\mathbf{X}_\mathrm{k}\right)^{\mathbf{ES}(j,2\mathrm{k})}\right) & \text{case 0} \\
\hat{\mathbf{Y}} = a_0 + \sum_{j=1}^{m} a_j f\left(\left(\mathbf{X}_1\right)^{\mathbf{ES}(j,1)} \cdot \ldots \cdot \left(\mathbf{X}_\mathrm{k}\right)^{\mathbf{ES}(j,\mathrm{k})}\right) & \text{case 1} \\
\hat{\mathbf{Y}} = a_0 + \sum_{j=1}^{m} a_j \left(\mathbf{X}_1\right)^{\mathbf{ES}(j,1)} \cdot \ldots \cdot \left(\mathbf{X}_\mathrm{k}\right)^{\mathbf{ES}(j,\mathrm{k})} f\left(\left(\mathbf{X}_1\right)^{\mathbf{ES}(j,\mathrm{k}+1)} \cdot \ldots \cdot \left(\mathbf{X}_\mathrm{k}\right)^{\mathbf{ES}(j,2\mathrm{k})}\right) & \text{case 2} \\
\hat{\mathbf{Y}} = g\left(a_0 + \sum_{j=1}^{m} a_j \left(\mathbf{X}_1\right)^{\mathbf{ES}(j,1)} \cdot \ldots \cdot \left(\mathbf{X}_\mathrm{k}\right)^{\mathbf{ES}(j,\mathrm{k})} \right) & \text{case 3}
\end{array} \qquad (9)$$

where $\hat{\mathbf{Y}}$ is the vector of model predictions.

Thus, EPR's model space may be extended by the structures in Eqs. (9), which remain based on pseudo-polynomial regression as in Eq. (1). User-specified functions *f* reported in Eqs. (9) may be natural logarithmic, exponential, tangent hyperbolic, etc. Note that the last structure in Eqs. (9) requires the assumption of an invertible function *g*, because of the subsequent stage of parameter estimation. The term 'pseudo-polynomial expressions' is used here because the parameters of any of the expressions in (9) can be computed as for a linear problem and/or for true polynomial expressions. Moreover, Eqs. (9) are transformed into the form of Eq. (1) during evolutionary search. Finally, the inclusion of exponential and logarithmic functions in the general expression of Eqs. (9) allows EPR to explore a large space of formulae where the analyst's understanding of the physical process warrants their inclusion. However, if such functions are not naturally describing the phenomenon being modelled EPR search would find exponent values for such inputs to be equal to zero.

As earlier said, parameters a_j are estimated by a least squares (LS) method integrated in the EPR procedure (Giustolisi and Savic, 2006). The LS guarantees a two ways relationship between the pseudo-polynomial structure and its coefficients. In addition to the usual LS search, the user can force the LS to search for structures that contain positive coefficients only (a_j>0). This is particularly useful in modelling systems where there is a high probability that the negative coefficient values (a_j<0) are selected to balance the particular realization of errors related to the finite training data set (Giustolisi et al., 2007).

Uncertainty Analysis in EPR

The models returned by EPR contain some constant values, each one determined for that single model. Those constant values (i.e. model parameters) are estimated by the LS approach, as already explained above. This implies that, for a given data set, a confidence interval can be computed to assess the uncertainty, i.e. reliability of the particular parameter estimate a_j. These uncertainties are computed here by using the asymptotic covariance method (Ljung, 1999). The use of this method is possible because all the models obtained by EPR are linear with respect to the unknown pseudo-polynomial coefficients (non-linearity is contained in each monomial model expression i.e. in the combinations of inputs). This fact is of particular relevance, since the uncertainty analysis of parameters here presented has been undertaken as if the models were linear, and this is made possible by the particular structure of the EPR models. Assuming that the residuals of the models are normally distributed with zero mean and variance λ_0, and that $\boldsymbol{\theta}_N$ is the estimate of the "true" parameter vector $\boldsymbol{\theta}_0$ in the N-dimensional data set, the covariance matrix $\mathbf{P}_N$ can be calculated as follows:

$$\mathbf{P}_N = \lambda_0 \left[\mathbf{XX} \times \mathbf{XX}^{\mathrm{T}} \right]^{-1} \tag{10}$$

where the j-th column of matrix **XX** for e.g. case 0 in Eqs. (9) is defined as follows:

$$\mathbf{XX}_j = \left(\mathbf{X}_1\right)^{\mathbf{ES}(j,1)} \cdot .. \cdot \left(\mathbf{X}_{\mathrm{k}}\right)^{\mathbf{ES}(j,\mathrm{k})} \cdot f\left(\left(\mathbf{X}_1\right)^{\mathbf{ES}(j,\mathrm{k}+1)}\right) \cdot .. \cdot f\left(\left(\mathbf{X}_{\mathrm{k}}\right)^{\mathbf{ES}(j,2\mathrm{k})}\right) \tag{11}$$

Variance λ_0 can be estimated as follows (Ljung,1999):

$$\lambda_0 \cong \lambda_N = \frac{1}{N-d}\left(\mathbf{Y} - \hat{\mathbf{Y}}\right) \times \left(\mathbf{Y} - \hat{\mathbf{Y}}\right)^T \tag{12}$$

where **Y** is the vector of target values and d=dim($\boldsymbol{\theta}$) is the dimension of the vector of parameters. Note that expression (12) returns an *unbiased* estimate of λ_0. Once the covariance matrix of $\boldsymbol{\theta}_N$ has been computed, it is possible to sample $\boldsymbol{\theta}$ by using e.g. the Latin Hypercube of size M constituted by the multivariate normal distribution with mean vector $\boldsymbol{\theta}_N$ and covariance matrix $\mathbf{P}_\mathrm{N}$. In EPR, the value of M is usually assumed equal to 50. Once the M estimates of $\boldsymbol{\theta}_0$ are available, it is possible to compute the M different model predictions at a given time step. The maximum and minimum predictions for each data point are computed in order to define a sort of uncertainty band that is the confidence interval for the prediction. An average value of the interval width (uncertainty band) is then evaluated and that value is used as a model uncertainty performance indicator. In addition, further control can be imposed on the a_j coefficient values as in Giustolisi and Savic (2006). This is related to the coefficients uncertainty during the search. Indeed, it may be argued that a low coefficient value with respect to the variance of estimates corresponds to the terms that begin to describe the noise rather than the underlying function of the phenomena being analysed. Therefore, the distribution of estimated pseudo-polynomial coefficients is used to eliminate those parameters whose value is not sufficiently larger than zero (Giustolisi and Savic, 2004; 2006). It is again assumed that the parameter variation follows the Gaussian probability density function N(a_{j0},$\mathbf{P}_N$). Hence, the following expression is used (Giustolisi and Savic, 2006):

$$\left|a_{j0}\right| - \gamma\sqrt{P_{\mathrm{jj}}} \cong \left|a_j\right| - \gamma\sqrt{P_{\mathrm{jj}}} \leq 0 \quad \Rightarrow a_j = 0 \tag{13}$$

where the square root of P_{jj} denotes the standard deviation of the estimated constant a_j (calculated from the diagonal elements of the covariance matrix $\mathbf{P}_N$) and γ is the standard score (from the Standard Normal Table). Eq. (13) states that if, for example, the modulus of the estimated constant a_j is lower than 3.2905P_{jj} (which corresponds to a confidence level of 99.9%), the corresponding constant value is assumed to be equal to zero.

Single vs. Multi Objective GA based EPR

Although the original EPR methodology proved effective (Giustolisi and Savic, 2004; 2006), it was using the single-objective genetic algorithm (SOGA) (Goldberg, 1989) strategy for exploring the formulae space. In fact, this exploration was achieved by assuming first the maximum number of terms *m* in the pseudo-polynomial expressions shown in Eq. (1) and then sequentially exploring the formulae space having one, two, …, *m* terms. To speed up the convergence, the initial population of each EPR search was (optionally) seeded with the formulae obtained in the previous search (e.g. the population for formulae having *j* terms was seeded with the best formulae having *j*-1 terms). However, the SOGA-based EPR methodology has the following drawbacks: (1) its performance is exponentially decreasing with the increasing number of polynomial terms *m* (also because increasing *j* means more GA runs); (2) the results are often difficult to interpret. In fact, the set of models identified could be either ranked according to their fitness to data or according to their structural complexity. However, ranking models according to their structural complexity requires some subjective judgment, and consequently this process is often biased by the analyst's experience rather than being purely based on some mathematical criteria (Young et al., 1996) that in our case are the objectives; (3) when searching for the formulae with *j* terms, those having less terms belong to the space of formulae with *j* terms as a degenerative case. However these "degenerative formulae" could have a better accuracy than the previously found ones (i.e., for lower values of index *j*), but discarded because at run *j* there could be less parsimonious formulae that fit data better. To overcome these drawbacks, it is possible to use a multi-objective genetic algorithm (Goldberg, 1989) (MOGA) strategy in EPR. In fact, assuming *m* pseudo-polynomial terms and considering that all pseudo-polynomials that have less than *m* terms belong to the formulae space of *m* terms as a degenerative case, it is possible to explore the space of *m*-term formulae using the following two (conflicting) objectives: (1) maximization of the model accuracy and (2) minimization of the number of polynomial coefficients in the formulae. This problem can be solved using the MOGA approach based on the Pareto dominance criterion (Pareto, 1896). Using this criterion makes the EPR search faster because search for all models ($j=1,2,\ldots,m$) is performed simultaneously. Moreover, the models obtained in this way are already ranked according to the: (1) number of terms obtained (i.e. parsimony) and (2) accuracy achieved (i.e. model fitness to training data). Following this reasoning, further improvement to EPR is to use the MOGA strategy to optimise for the number of formulae inputs ($\mathbf{X}_i$ are the vectors) as well. Therefore, the enlarged objectives for the EPR search are as follows: (1) maximisation of the model accuracy, (2) minimisation of the number of polynomial coefficients and (3) minimisation of the number of inputs (e.g. the number of times each $\mathbf{X}_i$ appears in the model). Note that EPR can determine the Pareto front consisting of best formulae (maximum of *m* terms) considering both parsimony (number of constants and variables) and accuracy in a single formulae space exploration. This makes EPR results easily interpretable because the formulae are ranked according to the parsimony and accuracy objectives. Moreover, the overall Pareto front gives insight into the model selection phase. Finally, the GA used for the evolutionary stage of EPR is OPTIMOGA. Further details on OPTIMOGA can be found in Giustolisi et al. (2004).

EPR as a System Identification Tool

From a system identification point of view (Ljung, 1999), EPR is a non-linear global stepwise regression approach providing symbolic formulae for the models. The stepwise regression feature of EPR originates from the Draper and Smith (1998) method aimed at selecting attributes for linear models considering the objective of fitting a model to data. Thus, the space of "linear solutions" is explored by changing the model input according to a set of rules and by evaluating model agreement with data. EPR generalizes the original stepwise regression method by considering non-linear structures, which are pseudo-polynomials as above described. This means that the polynomial nature of the model assures a two ways relationship between each model structure and its parameters and, consequently, the parameter estimation phase is cast as a linear inverse problem. Furthermore, the exploration of the solution space is performed using an evolutionary computing approach. Therefore, from an optimization standpoint, EPR can be classified as a global search method, working on a combinatorial problem which often does not have a unique solution (i.e., search space 2 dimensions is defined as a multimodal surface). This approach results in the evolutionary exploration of the solution space constrained to the non-linear models (the linear model is as special case) having a pseudo-polynomial structure and assuring linearity with respect to parameter estimation.

In comparison to other data-driven technique EPR could also be seen as an attempt to overcome some reported drawbacks of genetic programming (Koza, 1992; Babovic and Keijzer, 2000), as described in Giustolisi and Savic (2006). From a regressive standpoint, EPR has the following beneficial features not found in other data-driven techniques:

- a small number of constants to be estimated (helps avoiding over-fitting problems, especially for small data sets);
- a linear parameters estimation (assuring the unique solution is found when the inverse problem is well-conditioned);
- an automatic model construction (avoiding the need to preselect the functional form and the number of parameters in the model) and;
- a transparent form of the regression characteristics makes model selection easier, i.e., the multi-objective feature allows selection not only based on fitting statistics.

Similarly, when compared to classical regression techniques, for example input-output artificial neural networks (ANNs), it is possible to emphasize the following EPR features:

- EPR can perform both linear and non-linear analyses in a single algorithm run, whereas ANNs are either linear or non-linear depending on the transfer function user selected for the hidden neurons.
- EPR does not require the assumption on the model structure (components of the input vector for each pseudo-polynomial term, the number of pseudo-polynomial terms, nor the function model structure). ANNs generally require the prior selection of the input vector, of the number of hidden neurons and of their unique transfer function.
- EPR uncertainty analysis is easy to perform due to model linearity with respect to parameter estimation. ANNs parameter (weight) estimation is an inverse non-linear problem, making it difficult to deal with.
- EPR provides a Pareto set of the best models trading off parsimony against model fit (to training data). ANNs provide one, best-fit model considering an objective function used for parameter estimation. However, ANN can be developed by means of a multi-objective strategy, as shown by Giustolisi and Simeone (2006).

These features will be discussed in more detail in the following case study.

Case Study

The aim of the case study is to demonstrate capabilities of EPR as analysis tool. The case study should demonstrate that EPR is particularly helpful as a decision support tool for data modelling and analysis. Thus, EPR is tested on a case study aimed at determining the dynamic relationship between rainfall depth and water table depth for a shallow unconfined aquifer located in southeast Italy.

The shallow unconfined aquifer system of Brindisi is located in the northern part of the Salento Peninsula in Apulia, southeast Italy (Fig. 1). It serves as an opportune subject for investigation because it is a relatively simple hydrogeological structure that occupies a small area (about 200–300 km^2) and comprises a shallow aquifer that is supplied only by direct rainfall, an ideal arrangement for scrutinizing the relationship between groundwater levels and rainfall (Giustolisi et al., 2008). For further details on Geological and hydrogeological framework of the groundwater see Giustolisi et al. (2008).

Phreatic Level and Raingauge Station

In order to study the relationship between groundwater levels and rainfall amounts, the data that have been used are measured phreatic levels from the gauging station located near Brindisi and rainfall data from the Brindisi raingauge station. Both of these stations are operated by the National Hydrographical Service of Italy. Observations from the phreatic level gauging station are available for a relatively long period extending from 1952 to 1996, see Figure 2(b), while rainfall data have been recorded at Brindisi for an even longer duration, since the end of the 19th century. Despite such an ample rainfall record, the authors have chosen to use only data corresponding to the observational period of groundwater levels. Figure 1 indicates the location of the well from which the groundwater level data used in the application of the multi-objective EPR were sampled. The available data collections are: (a) a rainfall time series and (b) a groundwater level time series. Each series incorporates 528 data points in which the rainfall series consists of monthly cumulative depths measured in cm and the groundwater series comprises average monthly values of the depth of the water's free surface in the well (measured from the mouth of the well, which is located at 35.92 m above sea level). Both the rainfall and groundwater data series cover a 44-year period: from January 1953 to December 1996. Figures 2(a) and (b) presents the time plots of rainfall and groundwater levels, respectively. The study area is distinguished by a typically Mediterranean climate, experiencing one dry period and one wet period each year. Preliminary studies reveal that replenishment of the groundwater system typically occurs in the first three months of the year while the rainy autumn months do not contribute to recharge since the water that infiltrates is needed to restore the water content associated with the field capacity of the soil. The greatest variations of the pluviometric regime occur in March and April, reaching a minimum during summer when evapotranspiration is more intense. Recharge is more vigorous passing from autumn towards spring for low intensity rainfall events. The contribution to infiltration of high intensity and short duration rainfall events is low as a consequence of soil permeability and of runoff and evapotranspiration processes. A long-term analysis reveals that, during the 40-year historical record considered, a remarkable decline in phreatic levels has occurred.

Preliminary Modeling Aspects

Among the equations in (9), the structure named ‘case 2’ was chosen using *m*=3. Then, the family of models that will be explored is

$$\begin{aligned} \hat{y} &= a_0 + \sum_{j=1}^{m=3} a_j A_j \cdot \ln\left(B_j\right) \\ A_j &= x_t^{ES(j,1)} \cdot \mathrm{K} \cdot x_{t-nb}^{ES(j,nb+1)} \cdot y_{t-1}^{ES(j,nb+2)} \cdot \mathrm{K} \cdot y_{t-na}^{ES(j,na+nb+1)} \\ B_j &= x_t^{ES(j,na+nb+2)} \cdot \mathrm{K} \cdot x_{t-nb}^{ES(j,na+2nb+2)} \cdot y_{t-1}^{ES(j,na+2nb+3)} \cdot \mathrm{K} \cdot y_{t-na}^{ES(j,2(na+nb+1))} \end{aligned} \tag{14}$$

where $\hat{y}$ is the estimated output of the system/process; a_0, a_1, a_2, a_3, are the model constant values; x_{t-l} and y_{t-i}, are the model inputs and outputs, respectively. They are selected by the process considering the user-specified maximum number of inputs *nb* (the number of past inputs is *nb*-1) and outputs *na*. They are process-selected as the exponent of “0” means that a particular input/output is eliminated. Thus, although the general configuration of the structures is defined by the user (i.e., inputs, exponents and maximum expression length), EPR can return simplified structures according to the strategy it pursues.

The space of candidate formulas can be explored by EPR according to two main strategies: (1) an SO search and (2) an MO approach. Although the effectiveness of the SO approach in environmental modeling has been demonstrated (Giustolisi et al., 2007), it presents some drawbacks. The MO approach outperforms that of SO, since it explores the space of candidate formulas by assuming just the maximum number of constants (a_j). This case study focuses on the MO strategy (Giustolisi et al., 2008).

The objectives assumed for this search are three: (1) the number of constants (a_j), (2) the total number of inputs ($\mathbf{X}_k$) represented in each formula and (3) the models’ fitness to data.

The modelling phase was carried out according to the following assumptions:

- both time series were split into two subsets: the first made up of 300 samples, called the training set, and the second made up of 228 samples, called the test set. The test set is considered in a latter phase, when EPR has already generated the set of best models. In this phase it is important to test the generalisation capabilities of the models, i.e. to assess how these models perform when fed with an input data set different from that used to identify them (unseen data).
- The set of variables considered as candidate input to the models are: H_{t-1} (*na*=1) as past value of the groundwater head, and P_t, P_{t-1}, P_{t-2}, P_{t-3} (*nb*=4) as actual/past values of the rainfall depths. Subscripts denote the measurement time: for instance *t*-1 indicates the groundwater level observed one month

before the present (*t*) or model computed when the *k*-month ahead prediction is performed. These candidate inputs to the models have been selected according to the aquifer response to the rainfall perturbations (3-month delayed) reported in Ricchetti and Polemio (1996). Past output $H_{t\text{-}1}$ have been incorporated to reflect the persistence of piezometric head variation.

- The possible model structures, see Eq. (14), are assumed to be pseudo-polynomial considering the natural logarithm as a candidate function to model the phenomena.
- The polynomial expressions consist of three terms at most, excluding the bias term (if selected by the procedure).
- Each pseudo-monomial term is the product of the methodology-selected inputs to the power of the exponents selected by EPR in the pre-specified set { -1; 0; 1}. The exponent 0 allows the procedure to deselect the unnecessary inputs and the exponents {-1; 1} introduce an inverse-linear and linear effects to the input, respectively. Finally, the natural logarithm function introduces a smoothing effect to the input.
- The LS estimate of the constant a_j is constrained to positive values according to the approach by Lawson and Hanson (1974).
- Data are not scaled (e.g. in the range [0 1]).
- The optimization parameters are: 1000 generations, initial population size 20 elements, probability of crossover 0.4 and probability of mutation 0.1.
- The number of potential candidate solutions among which EPR searches is 2.06×10^{14}.

The main fitness indicator considered in this paper is the Coefficient of Determination (CoD).

The set of non-dominated models identified by EPR defines a global scenario of possible model structures which is presented to the analyst who must then select the best candidate for the problem at hand. This final selection is guided by an analysis of the similarities and differences among formulae and through consideration of the trade-off between structural complexity and fitness level attained). Therefore, the user can identify those terms/inputs that are common among the models and assess which terms/inputs are discarded by the methodology when the structural complexity decreases as recently proposed by Giustolisi (2006) for support vector machines; by Giustolisi and Simeone (2006) for artificial neural networks and Giustolisi et al. (2008) for EPR itself. Moreover, this analysis permits identification of terms that appear in one model only and such terms are likely to be weakly related to the physical phenomenon, but rather to the specific error realization contained in data.

Results

In this section, the entire set of non-dominated EPR models is presented keeping in mind that the goal is to furnish a decision support strategy (Giustolisi, 2006; Giustolisi and Simeone, 2006; Giustolisi et al.; 2008) and not strictly a model suitable for a unique case study. EPR identified 17 non-dominated models, described by the Eqs. (15) to (31), completing 1,000 generation runs in about 4 minutes using a notebook computer with a Pentium Intel M 1.10 GHz processor.

$$H_t = 31.43 \tag{15}$$

$$H_t = 0.92H_{t-1}+2.46 \tag{16}$$

$$H_t = 0.93H_{t-1}+7.02P_{t-1}+2.13 \tag{17}$$

$$H_t = 381.27H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+73.27 \tag{18}$$

$$H_t = 0.91H_{t-1}+5.95P_{t-1}+4.66P_{t-2}+2.49 \tag{19}$$

$$H_t = 0.91H_{t-1}+239.69P_{t-1}P_{t-2}+2.69 \tag{20}$$

$$H_t = 0.90H_{t-1}+5.93P_{t-1}+179.08P_{t-2}P_{t-3}+3.08 \tag{21}$$

$$H_t = 377.05H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+236.52P_{t-1}P_{t-2}+72.72 \tag{22}$$

$$H_t = 381.95H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+6.03P_{t-1}+209.30P_tP_{t-3}+73.17 \tag{23}$$

$$H_t = 377.56H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+77.48P_{t-1}P_{t-2}\ln\left(P_{t-2}^{-1}\right)+209.30P_tP_{t-3}+72.76 \tag{24}$$

$$H_t = 376.31H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+193.51P_{t-1}P_{t-2}+177.61P_tP_{t-3}+72.60 \tag{25}$$

$$H_t = 378.71H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+2437.06H_{t-1}P_{t-1}P_{t-2}\ln\left(P_{t-2}^{-1}\right)+177.61P_tP_{t-3}+72.89 \tag{26}$$

$$H_t = 376.68H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+64.53P_{t-1}P_{t-2}\ln\left(P_{t-2}^{-1}\right)+176.45P_tP_{t-3}+72.62 \tag{27}$$

$$H_t = 378.24H_{t-1}^{-1}\ln\left(H_{t-1}^{-1}\right)+63.82P_{t-1}P_{t-2}\ln\left(P_{t-2}^{-1}\right)+57.78P_tP_{t-3}\ln\left(P_{t-3}^{-1}\right)+72.79 \tag{28}$$

$$H_t = 377.46 H_{t-1}^{-1} \ln\left(H_{t-1}^{-1}\right) + 63.57 P_{t-1} P_{t-2} \ln\left(P_{t-2}^{-1}\right) + 1.86 H_{t-1} P_t P_{t-3} \ln\left(P_{t-3}^{-1}\right) + 72.70 \tag{29}$$

$$H_t = 378.41 H_{t-1}^{-1} \ln\left(H_{t-1}^{-1}\right) + 2001.35 H_{t-1} P_{t-1} P_{t-2} \ln\left(P_{t-2}^{-1}\right) + 1.86 H_{t-1} P_t P_{t-3} \ln\left(P_{t-3}^{-1}\right) + 72.81 \tag{30}$$

$$H_t = 377.73 H_{t-1}^{-1} \ln\left(H_{t-1}^{-1}\right) + 2131.35 H_{t-1} P_{t-1} P_{t-2} \ln\left(P_{t-2}^{-1}\right) + 0.53 H_{t-1} P_t P_{t-3} \ln\left(H_{t-1} P_{t-2}^{-1} P_{t-3}^{-1}\right) + 72.73 \tag{31}$$

Note that the set of 17 models found by EPR range from the simple model representation of the average value (Eq. 15), through to the linear models of Eqs. (16), (17) and (19), and then on to more complex configurations. Table 1 reports the number of $\mathbf{X}_i$ used in the formulae (in the last three columns), the number of a_j and the fitness indicator, CoD, computed on training set.

The performance for models generalization is evaluated in terms of CoD computed on the data test set and summarized in Table 1, also. A bootstrap procedure (Efron, 1979) was applied for the CoD of the test set in order to improve the robustness of its estimation. For this purpose, the data were re-sampled 1,000 times; Table 1 provides the CoD values averaged on the 1,000 samples obtained for each class of prediction (1-, 2-, 4-, 6-months and simulation). Simulation stands for prediction performed without using groundwater level measurements at all in the model.

Furthermore, Table 1 reports the persistency model (H_t=H_{t-1}) predictions on test set (in the first row), which are an useful of models' performance. The average width of the uncertainty (as defined in the paragraph "Uncertainty Analysis in EPR") band of the 1-month ahead prediction due to parameter estimation is reported in the column after CoD performance on test set.

Discussion

EPR identified 17 non-dominated models with differing structural complexity and performance. For those models on-line predictions of the groundwater head at different time horizons are presented in Table 1. Those predictions related to the test set (unseen data) were never used for model construction. Note that, although the choice of the prediction horizons is motivated by management needs, a longer planning horizon, e.g. 6-infinity months (infinity means simulation) is reasonable for predicting the behaviour of the aquifer which in turn influences the management policies that can be adopted. On the other hand, shorter prediction horizons of 1-4 months can be useful for the adoption of emergency policies, for instance related to an anomalous dry period or excessive pumping. However, in a decision support strategy on model selection, it is important not only to compare the performance of each model, but to consider their structural variations and the contiguities of inputs and pseudo-monomial terms.

Then, we can observe that all models encompass the term $H_{t\text{-}1}$, which relates to the persistency of groundwater head variations. Eq. (16) reports the model found by EPR (H_t=0.92$H_{t\text{-}1}$+2.46), which is most similar to the complete persistency model (i.e., H_t=$H_{t\text{-}1}$). It is also important to note that the models (16), (17) and (19) are linear while other solutions are increasingly non-linear, i.e., ranging from one non-linear pseudo-monomial term to all of them being non-linear in a single formula. This demonstrates that the multi-objective strategy together with the particular EPR architecture (based on a sum of pseudo-monomials), allows linear and non-linear analysis of data in a single run of the EPR tool.

It could be also observed that the linear models generally perform well, if only slightly inferior to more complex non-linear models when short-time predictions (1-month and 2-months) are considered. However, non-linear complex models are better performing for long-term predictions and simulation. This indicates that the underlying physical phenomenon is non-linear because by increasing the prediction horizon (until simulation, which does not use $H_{t\text{-}1}$ measurements for prediction) the stochastic processes influencing short-time predictions disappear.

Furthermore, it is worth noting that with increased model complexity there are some pseudo-monomial terms appearing in a number of models. For example, $H_{t\text{-}1}\ln(H_{t\text{-}1})^{-1}$ is always present in non-linear models and the terms $P_{t\text{-}1}P_{t\text{-}2}$ and $P_tP_{t\text{-}3}$ appear in most non-linear models. Those terms, slightly modified using the natural logarithm function, as in $P_{t\text{-}1}P_{t\text{-}2}\ \ln(P_{t\text{-}2})^{-1}$ and $P_tP_{t\text{-}3}\ln(P_{t\text{-}3})^{-1}$, positively influence model performance, especially for long prediction horizons. The actual value of rainfall (P_t) is found to perform well for long-time predictions while the delayed rainfall, i.e., *nk*=1 ($P_{t\text{-}1}$), is present in the linear models.

Considering model uncertainty, it is important to note that the average width of the uncertainty band for 1-month ahead prediction on test set is within the range from 7 to 11cm and it does not depend on the model complexity (Table 1). This happens because all models are linear with respect to parameters. Furthermore, they contain a small number of constants, thus allowing parameter estimation on a generally well-conditioned linear inverse problem. However, this type of uncertainty analysis should not be confused with model stability (as a dynamical system), which is not guaranteed for non-linear models and in particular those with long prediction horizons.

Figures 3 and 4 report the model predictions using Eq. (28) on test set (unseen data), which is here selected for demonstration purposes. Note that CoD related to that model results for simulation is estimated to be 0.686 (CoD of 1 is a perfectly fitting model) on the test set (see Table 1) and CoD for 6-months ahead prediction is 0.778. Such accuracy can be considered acceptable for stakeholders interested in planning the aquifer operation for next season or even longer term.

Finally, the test set reported a period of drought lasting about 8 years starting from day #420 until about day #520. This window of data is useful for analysing predictions at different prediction horizons. For example, Figure 3 shows that the one to two-month ahead predictions follow the groundwater level variations quite closely while the peaks are missed for the increased prediction horizon (4-months). Figure 4 demonstrates, especially for the window of data [420; 520], that 6-month ahead prediction and simulation are useful in order to forecast the groundwater level variations for both seasonal and yearly time horizons.

Conclusions

The use of EPR as decision support strategy for data analysis and modelling is described. In particular, the procedure was applied to capture the dynamic relationship between groundwater heads (output) and rainfall depths (input). This application of EPR suggested the possible advantages of evolutionary multi-objective modelling in general. These are: (1) the expressions cover a wide range of solutions which represent the best models for different structural complexities, (2) several important aspects in the analysis of the an environmental system are considered as evident in the analysis of the Pareto front of Eqs. and (3) the algorithm is more computationally efficient compared with the multiple single-objective runs for separately analyzing fitness and complexity. These features allow the user to select from among a robust group of models, since a comprehensive set of possible structures can be developed for each purpose. Even if a single model is ultimately settled upon, a wide range of models can be helpful for understanding which terms/inputs are physically meaningful and which can comfortably be eschewed for the sake of model parsimony, while simultaneously striving for a degree of generality.

References

Babovic, V., and Keijzer, M. 2000. Genetic programming as a model induction engine. Journal of Hydroinformatics, 1: 35–61.

Draper, N.R., and Smith, H. 1998. Applied Regression Analysis. John Wiley and Sons, New York, USA.

Efron, B. 1979. Bootstrap Methods. Another Look at the Jackknife. The Ann. of Statist., *7*: 1-26.

Giustolisi, O., Doglioni, A., Savic, D.A., and di Pierro, F. 2008. An Evolutionary Multi-Objective Strategy for the Effective Management of Groundwater Resources. Water Resour. Res., 44, W01403.

Giustolisi, O. 2006. Using multi-objective genetic algorithm for SVM construction. Journal of Hydroinformatics, 2: 125-139.

Giustolisi, O., and Savic, D.A. 2006. A Symbolic Data-driven Technique Based on Evolutionary Polynomial Regression. Journal of Hydroinformatics, 3: 207-222.

Giustolisi O., and Savic, D.A. 2004. A Novel Genetic Programming Strategy: Evolutionary Polynomial Regression. Proceedings of Hydroinformatics 2004, Liong, Phoon & Babovic (eds), World Scientific Publishing Company, Vol. 1, pp. 787-794.

Giustolisi, O., and Simeone, V. 2006. Multi-Objective strategy in artificial neural network Construction. Hydrological Sciences Journal, 3: 502-523.

Giustolisi, O., Doglioni A., Savic, D.A., and Webb, B.W. 2007. A multi-model approach to analysis of environmental phenomena. Environmental Modelling & Software, 5: 674-682.

Giustolisi, O., Doglioni, A., Laucelli, D., and Savic, D.A. 2004. A proposal for an effective multiobjective non-dominated genetic algorithm: the OPTimised Multi-Objective Genetic Algorithm, OPTIMOGA. Report 2004/07, School of Engineering Computer Science and Mathematics, Centre for Water Systems, University of Exeter, UK.

Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, London, UK.

Golub, G.H. & Van Loan, C.F. 1993. Matrix Computations. The Johns Hopkins University Press, London, UK.

Holland, J. 1975. Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor, Michigan, USA.

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

Lawson, C.L., and Hanson, R.J. 1974. Solving Least Squares Problems. Prentice-Hall Inc, Englewood Cliffs, New Jersey, USA.

Ljung, L. 1999. System Identification: Theory for the User (second edition). Prentice-Hall Inc., Englewood Cliffs, New Jersey, USA.

Pareto, V. 1896. Cours D'Economie Politique. Rouge and Cic, Vol. I and II, Lausanne, Switzerland.

Ricchetti, E., and Polemio, M. 1996. L'acquifero superficiale del territorio di Brindisi: dati geoidrologici diretti e immagini radar da satellite (The shallow aquifer of Brindisi: geohydrological direct data and satellite radar images). Memorie della Società Geologica Italiana, 51: 1059-1074.

Soule, T., and Heckendorn, R.B. 2002. An analysis of the cause of the code growth in genetic programming. Genetic Programming and Evolvable Machines, 3: 283-309.

Young, P., Parkinson, S, and Lees, M. 1996. Simplicity out of complexity in environmental modelling: Occam's razor revisited. Journal of Applied Statistics, 2-3: 165-210.

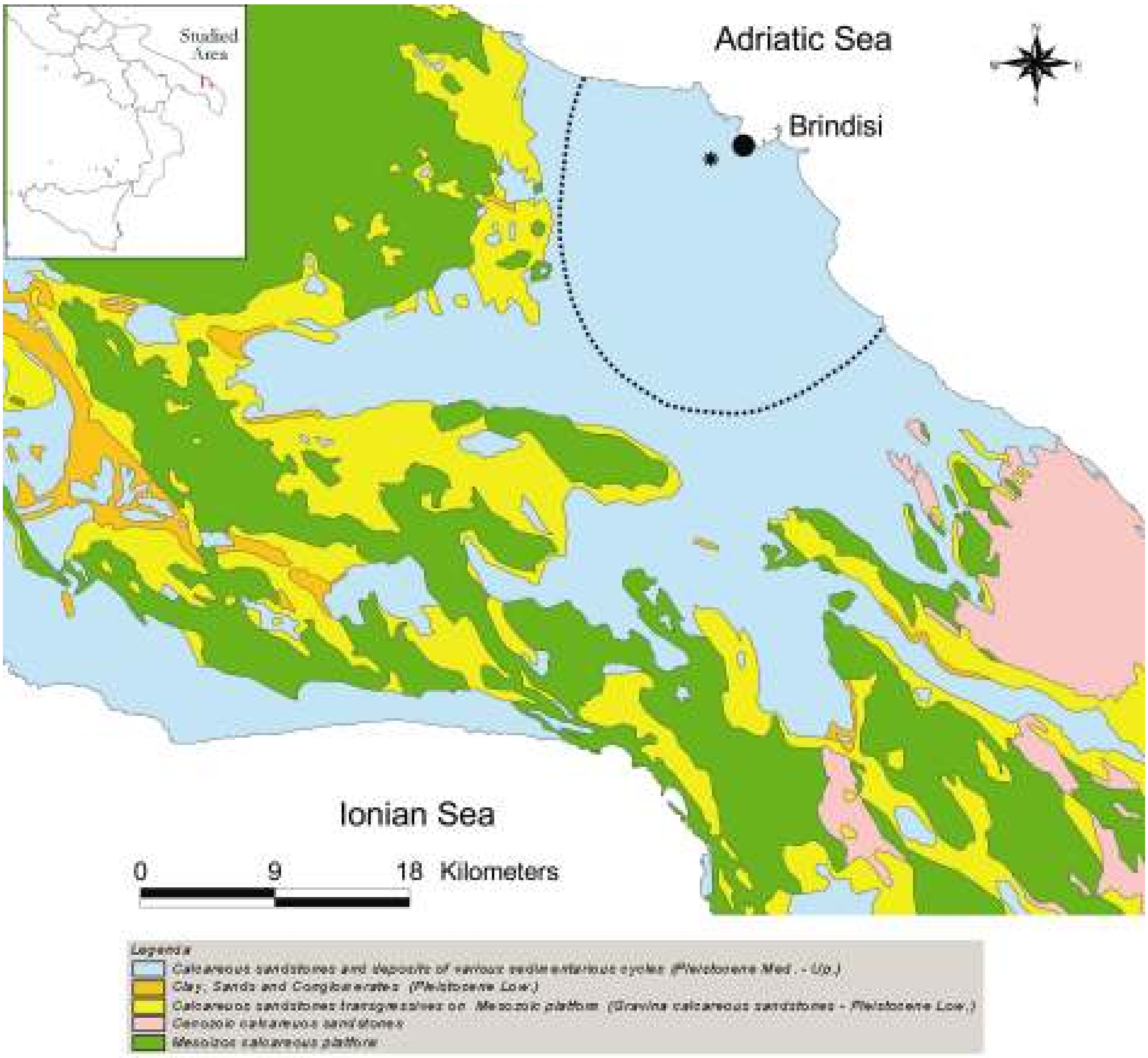
Studied Area
Adriatic Sea
Brindisi
Ionian Sea
0
9
18 Kilometers
Legenda
Calcareous sandstones and deposits of various sedimentarious cycles (Pleistocene Med. - Up.)
Clay, Sands and Conglomerates (Pleistocene Low.)
Calcareous sandstones transgressives on Mesozoic platform (Gravina calcareous sandstones - Pleistocene Low.)
Cenozoic calcareous sandstones
Mesozoic calcareous platform

Figure 1. Location of the study aquifer [∗ sampling well; • raingauge station].

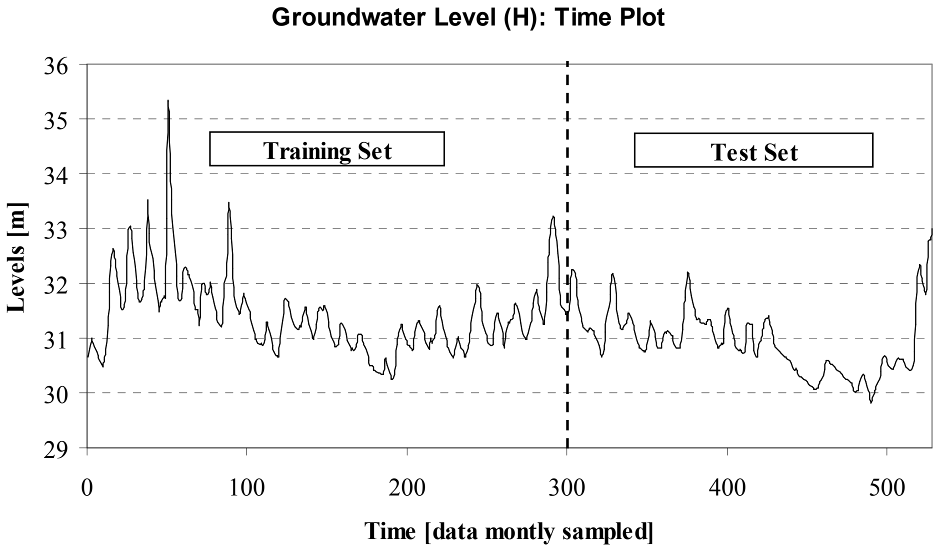

Figure 2. (a) Rainfall time plot and (b) groundwater level time plot of available data.

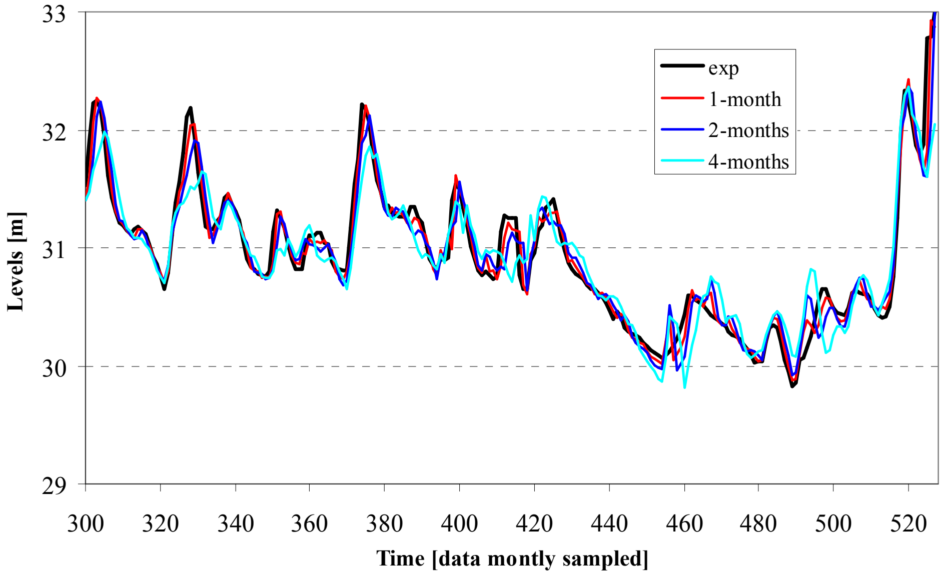

Figure 3. Groundwater head prediction at 1, 2, 4 months ahead computed on the test set using the model of Eq. (28).

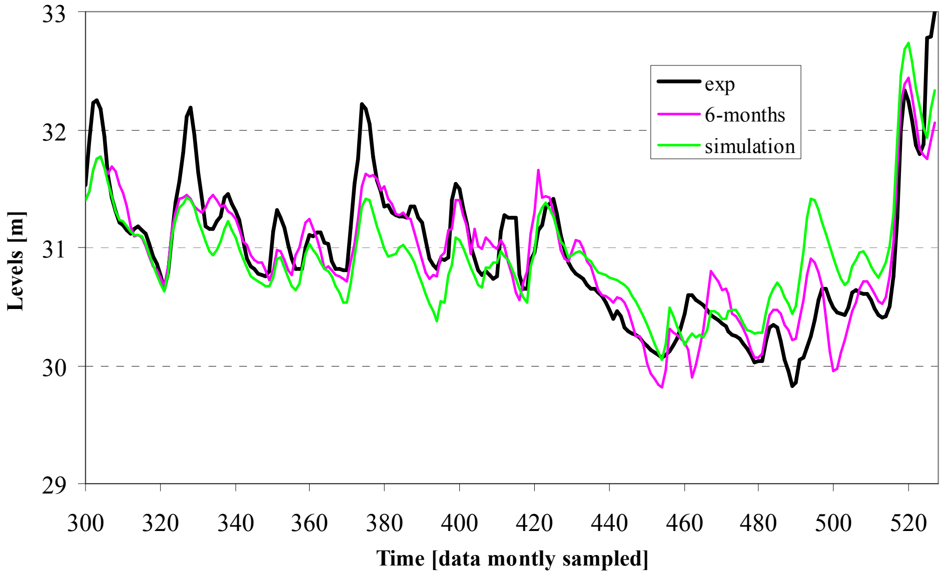

Figure 4. Groundwater head prediction at 6 months ahead and simulation computed on the test set using the model of Eq. (28).

Table 1. EPR models: predictions and uncertainty analysis on test set; values on training set of the multi-objective analysis. The bold character identifies the row corresponding to the selected Eq. (28).

	Test set (Unseen data)						Training set		
	CoD 1-month	CoD 2-months	CoD 4-months	CoD 6-months	Simulation	Uncertainty 1-month	CoD training	# inputs	# terms
H_t=H_{t-1}	0.929	0.782	0.437	0.239	-0.718	/	/	/	/
Eq. (15)	-0.718	-0.706	-0.711	-0.700	-0.700	0	0	0	1
Eq. (16)	0.921	0.768	0.444	0.254	-0.757	0.072	0.85	1	2
Eq. (17)	0.947	0.879	0.727	0.646	0.171	0.098	0.844	1	1
Eq. (18)	0.928	0.782	0.472	0.287	-0.334	0.078	0.876	2	3
Eq. (19)	0.951	0.882	0.768	0.707	0.314	0.109	0.855	2	2
Eq. (20)	0.948	0.878	0.737	0.663	0.274	0.079	0.886	3	4
Eq. (21)	0.951	0.881	0.756	0.696	0.355	0.098	0.883	3	3
Eq. (22)	0.951	0.89	0.777	0.731	0.318	0.074	0.892	4	4
Eq. (23)	0.958	0.905	0.802	0.756	0.537	0.101	0.886	4	3
Eq. (24)	0.955	0.899	0.791	0.75	0.385	0.073	0.895	5	4
Eq. (25)	0.96	0.904	0.808	0.774	0.592	0.096	0.888	5	3
Eq. (26)	0.955	0.895	0.79	0.751	0.393	0.082	0.896	6	4
Eq. (27)	0.961	0.909	0.816	0.779	0.636	0.094	0.888	6	3
Eq. (28)	**0.961**	**0.904**	**0.813**	**0.778**	**0.686**	**0.085**	**0.899**	**7**	**4**
Eq. (29)	0.96	0.908	0.815	0.779	0.685	0.086	0.9	8	4
Eq. (30)	0.961	0.906	0.812	0.777	0.685	0.097	0.9	9	4
Eq. (31)	0.96	0.906	0.81	0.77	0.632	0.093	0.9	10	4