Adaptive meta-modeling-based simulation optimization in basin-scale optimum water allocation: a comparative analysis of meta-models

Golnazalsadat Mirfenderesgi and S. Jamshid Mousavi

ABSTRACT

Incorporating river basin simulation models in heuristic optimization algorithms can help modelers address complex, basin-scale water resource problems. We have developed a hybrid optimization-simulation model by linking a stretching particle swarm optimization (SPSO) algorithm and the MODSIM river basin decision support system (DSS), and have used the SPSO-MODSIM model to optimize water allocation at basin scale. Due to high computational cost of the SPSO-MODSIM model, we have, subsequently, used four meta-model types of artificial neural networks (ANN), support vector machines (SVM), kriging and polynomial response functions, replacing the MODSIM DSS, in an adaptively learning meta-modeling approach. The performances of the meta-models are first compared in two Ackley and Dejong benchmark functions optimization problems, and the meta-models are then evaluated by solving the Atrak river basin water allocation optimization problem in Iran. The results demonstrate that independent of the meta-model type, the sequentially space-filling meta-modeling approach can improve the performance of meta-models in the course of optimization by adaptively locating the promising regions of the search space where more samples need to be generated. However, the ANN and SVM meta-models perform better than others in saving the number of costly, original objective function evaluations.

Key words | adaptive meta-modeling, function approximation, particle swarm optimization, simulation optimization, water allocation

Golnazalsadat Mirfenderesgi
Formerly with Department of Civil and
    Environmental Engineering,
Amirkabir University of Technology (Tehran
    Polytechnic),
Tehran,
Iran
Current address:
Department of Civil, Environmental and Geodetic
    Engineering,
The Ohio State University,
Columbus, OH,
USA

S. Jamshid Mousavi (corresponding author)
Department of Civil and Environmental
    Engineering,
Amirkabir University of Technology (Tehran
    Polytechnic),
Tehran,
Iran
E-mail: *jmosavi@aut.ac.ir*

INTRODUCTION

Modeling water resources at basin scale calls for different kinds of mathematical models, representing complex hydro-logic, legal-administrative framework, socio-economic and management processes taking place in a river basin. Most of the mathematical river basin management models, e.g. MODSIM (Labadie 1995), ACRES (Sigvaldson 1976), RIBA-SIM (Hydraulics 1991), CALSIM (Draper *et al.* 2004), MikeBasin (DHI Water & Environment 2006), River Ware (Zagona *et al.* 2001), HEC 5 (US Corps of Engineers; Bonner 1989) or its modified version HEC-ResSim and WEAP (SEI 1999), have been developed for simulation purposes. They typically make use of pre-defined reservoir operation rules or single-period optimization schemes to optimally allocate available water resources among competing users, so they are not able to identify optimal size and operation of the systems' components by using multi-period optimization. There have been some cases where large scale, multi-reservoir systems management have been approached by full dynamic, multi-period optimization algorithms with more degree of simplifications, e.g. lineari-zation (Kuczera & Diment 1988; Cai *et al.* 2002; Jenkins *et al.* 2004).

The purpose of modeling a river basin system could also go beyond the simulation of a multiple reservoir system

operation or even optimizing water allocations for multiple supply-demand systems. This is because a basin system consists of various components such as reservoirs, aquifers, pumping systems, hydroelectric power plants, diversions, and water transfer systems. Therefore, there are various processes or inter-relationships taking place within or among the components, e.g. erosion, sedimentation, runoff and flow routing, recharge, pollution and contamination transport, eutrophication, represented by highly non-linear, non-convex, discontinuous and non-algebraic equations solution of which demands more detailed, computationally intensive simulation models.

Simulation-optimization is an alternative approach for solving large-scale river basin optimization problems, in which a simulation model is linked to a global heuristic optimization algorithm (Shourian *et al.* 2008). One of the main advantages of the approach is that it does not require the variables, functions, relations and the related computer codes, simulating the system's processes, to be continuous, differentiable, algebraic and even completely accessible by users. These features provide the potential use of more detailed simulation models that can better represent the real processes being simulated. However, the main problem with this approach is the computational load needed to run the simulation model.

Although meta-heuristic, nature-inspired stochastic search methods, which use random elements to transfer one candidate solution into a hopefully better solution, are efficient optimization tools, they need a large number of function evaluations, and each requires the running of the mentioned complex simulation model for objective function evaluation. This makes the resulting simulation-optimization approach computationally intensive. Surrogate modeling, also known as meta-modeling (Blanning 1975), has evolved and is extensively used to produce computationally efficient surrogates of high-fidelity models (Sacks *et al.* 1989; Jones 2001; Razavi *et al.* 2012a). In this technique, a cheaper-to-run surrogate of the original model, running of which is much faster than the original simulation model, is used instead of the exact model. Artificial neural networks (ANN), support vector machines (SVM), kriging and polynomial response functions are some of the most commonly used approximation techniques in meta-modeling that have been applied to various engineering optimization problems.

Robinson & Keane (1999) presented a case for employing variable-fidelity analysis models and approximation techniques to improve the efficiency of evolutionary optimization for complex design tasks. This problem was revisited by El-Beltagy *et al.* (1999) who argued that the issue of balancing the concerns of optimization with that of design of experiments must be addressed. Farhang-Mehr & Azarm (2005) employed a sequential approach with adaptation to irregularities in the response behavior for Bayesian meta-modeling in engineering design simulations. Computational frameworks for integrating a class of single-point approximation models with evolutionary algorithms (EAs) were proposed by Nair & Keane (2001) for a domain-specific class of approximation models and for more general models by Ratle (2001), who integrated an EA with kriging models. EAs and global optimization algorithms were coupled with local search and quadratic response surface methods (Liang *et al.* (2000), stochastic response surface models (Regis & Shoemaker 2007), radial basis functions (RBFs) (Shoemaker *et al.* 2007) and ANN (Jin *et al.* 2002). Simpson *et al.* (2004) made a detailed survey on the state-of-the-art in meta-modeling.

Mousavi & Shourian (2010) presented the adaptive sequentially space filling (ASSF) meta-modeling approach in which the sub-problems of design of experiments, function approximation and function optimization in a surrogate optimization problem were sequentially solved in a feedback loop. Tsoukalas & Makropoulos (2015) compared the performance of state-of-the-art evolutionary and surrogate-based multi-objective optimization algorithms within the parameterization-simulation-optimization framework for robust multi-reservoir rules generation under hydrological uncertainty.

In spite of extensive work done in meta-modeling for decreasing the computational burden of global optimization techniques with computationally intensive function evaluations, the comparison of different meta-models, particularly in hydro-systems, is limited. Razavi *et al.* (2012b) listed 32 studies in water-resources-related problems and pointed out that the focus of few of them is on the comparison of different meta-models' performances. In this study, we aim to compare ANN, SVM, kriging and polynomial models in a surrogate optimization-based water allocation problem at the Atrak river basin in Iran. The surrogate models replace MODSIM decision support system (DSS) in the particle

swarm optimization (PSO) algorithm using the ASSF approach already developed by Mousavi & Shourian (2010).

The paper is structured as follows: first a brief overview of the ASSF approach, PSO algorithm and principle features of the four mentioned meta-modeling techniques are presented. Next, the results of application of the ASSF approach to two bench-mark function optimization problems with different meta-models are reported to assess the performance of the proposed tools. We then apply the meta-modeling techniques to a real world, basin-scale water allocation case study in north-east of Iran, and the results are presented and discussed followed by an overall evaluation of the considered meta-modeling techniques in the summary and conclusion section.

ASSF META-MODELING

A brief description of the ASSF approach, developed by Mousavi & Shourian (2010), is presented in this section. The three main features of the ASSF approach are design of experiments (DoE or function evaluation), function approximation and surrogate function optimization that are integrated in an inter-related scheme. Consider the following formulation of an optimization model, referred to as problem (1).

$$\min f(\underline{y})$$

Subject to:
$$h_i(\underline{x}) = 0, \ i = 1, \ \ldots, m \tag{1}$$
$$g_i(\underline{x}) \geq 0, \ j = 1, \ \ldots, n$$
$$y = sim(\underline{x})$$

where $f(\underline{y})$ is the objective function to be minimized, $\underline{x}$ is the vector of decision variables and $h_i(\underline{x})$ and $g_i(\underline{x})$ are, respectively, the equality and inequality constraints. In the above problem, the evaluation of objective function $f(\underline{y})$ requires the running of the simulation model $y = sim(\underline{x})$. In an important class of simulation optimization problems in the field of water resources engineering, a high-fidelity, computationally intensive simulation model (e.g. a river basin water allocation model, a river reservoir water quality model, a pressure-driven or head-driven water distribution model, a rainfall-runoff or

earth system hydrological model, a numerical hydrodynamic surface water, groundwater or sediment model) is used to evaluate $f(y = sim(\underline{x}))$. Note that the mathematical functions in the simulation model $sim(\underline{x})$ may not be algebraic as needed in typical gradient-based optimization techniques. Moreover, the function $f(\underline{y})$ may be multi-modal with respect to decision vector $\underline{x}$, and some of the mathematical functions of $h_i(\underline{x})$ and $g_i(\underline{x})$ could be non-smooth or can make the feasible space of the problem non-convex. Facing these difficulties, meta-heuristic and evolutionary optimization techniques are promising in solving these types of optimization problems. They can easily be linked with any simulation model without the need to have access to computer codes or details of the function $sim(\underline{x})$. Nevertheless, they typically need thousands of objective function, $f(\underline{y})$, evaluations before converging to a good or near optimum solution. Since each function evaluation needs the high-fidelity simulation model $sim(\underline{x})$ to run, problem (1) could become computationally very difficult to solve. To deal with this difficulty, a meta-model replacing the simulation model, $sim(\underline{x})$, may be used.

Approximating the functional relationship between $f(\underline{y})$ and the vector of input decision variables $\underline{x}$ by using the smallest possible number of $(\underline{x}, f(\underline{y}))$ examples is an important subject in meta-modeling. Therefore, a function approximation technique is employed while solving problem (1) where instead of optimizing the original function $f(\underline{y})$, an approximate function $\tilde{f}(\underline{y})$ is optimized that is referred to as problem (2) (Mousavi & Shourian 2010):

$$Min \, \tilde{f}(\underline{y})$$

Subject to:
$$h_i(\underline{x}) = 0, \ i = 1, \ \ldots, m \tag{2}$$
$$g_i(\underline{x}) \geq 0, \ j = 1, \ \ldots, n$$

Because $\tilde{f}(\underline{y})$ is to be evaluated instead of evaluating the original function $f(\underline{y})$, the first issue in surrogate optimization is about how to determine the approximate, surrogate function $\tilde{f}(\underline{y})$ with the smallest possible number of $f(\underline{y})$ evaluations. Below is the condition for an adequately accurate function approximation:

$$\left| \tilde{f}(\underline{x}) - f(\underline{x}) \right| \leq \varepsilon, \quad \forall x \in X \tag{3}$$

where ε is the accuracy parameter, $\left| \tilde{f}(\underline{x}) - f(x) \right|$ is the approximation error and X is the search space of problem (1). A reasonable way to secure the required precision for the approximate function in the optimization problem (2) is to design sufficient number of experiments to fill the search space uniformly. These experiments are then implemented to construct a meta-model. Therefore, if D experiments are needed for constructing the meta-model, the function $f(\underline{y})$ has to be evaluated D times. In other words, to determine the approximate function $\tilde{f}(\underline{y})$ in problem (2), a set of experiments known as $S_d = \{(\underline{x}_k, f(sim(\underline{x}_k)))\}$, $k = 1, \ldots, D$, must be generated. This problem, dealing with design of experiments, is referred to as problem (3).

The ASSF approach starts with generating a relatively small number of experiments in the search space, just enough to construct a preliminary approximate surrogate model. These experiments are designed randomly, and the function $f(sim(\underline{x}))$ is evaluated for each of them by running of the original, exact simulation model (design of experiments and function evaluation, problem (3)). Two main factors affecting the initial number of experiments needed in the search space (size of S_d) are non-linearity of function f and the dimension of input vector $\underline{x}$ (Mousavi & Shourian 2010). The minimum initial number of data points is selected so as not to have an over-fitted function approximation model. A meta-model, such as ANN, is trained then by using the generated data. This stage is about approximating function f (function approximation sub-problem that is referred to as problem (2a)).

Subsequently, the minimum or optimum value of approximate function f will be determined by using an evolutionary optimization algorithm. This stage is in fact the approximate function optimization stage represented as problem (2). If the function approximation stage has been undertaken precisely, one can expect that the minimum value of the approximate function f, reached by solving problem (2), is almost the same as the true minimum value of the original function f. In order to check whether or not the optimum value of $\tilde{f}$ (problem (2)) can be considered as the solution of problem (1), Mousavi & Shourian (2010) suggested checking two criteria: (1) existence of a predefined required number of experiments in the neighborhood of the solution in the training set. This

criterion is used to check if the meta-model has learned well to approximate the region around the solution, and (2) approximation error at the solution $(f - \tilde{f})$ being smaller than the predefined threshold error ε. This criterion is set to evaluate the error of the online approximation in terms of optimization.

If the above criteria are met, the solution can be considered as an optimizer of both f and $\tilde{f}$; otherwise, it will be considered as a false solution, located at the region that is not filled by adequate data samples (experiments), and therefore the meta-model has not learned the actual (original) function in this region. The algorithm stores the solution and marks it so that the search mechanism never finds it again. The EA will converge then to another point for which the two criteria will be checked again, and the process is continued up to a predefined maximum number of iterations. Once the iteration exceeds the maximum number of iterations, a set of non-perfect solutions, considered as center points of the gap regions, are located, and additional experiments will be designed in the neighborhood of each center point. These new experiments are added to the training data set generated before and constructs the new, updated training set. The meta-model is re-trained then using the new training data set, and a new run of EA, in which the re-trained meta-model is used, is performed to find new solutions. It is highly likely that the new EA skips the false solutions, already found by the EA. This procedure continues until the EA does not converge to a false solution any more. The flow diagram of the proposed approach is in Figure 1. More details of the ASSF approach may be found in Mousavi & Shourian (2010). In our study, PSO and MODSIM are the evolutionary optimization algorithm and the simulation model, $f(sim(\underline{x}))$, respectively.

Stretching particle swarm optimization algorithm

PSO is a stochastic-based search algorithm introduced by Kennedy & Eberhart (1995). The algorithm, simulating the behavior of a bird flock, serves as an optimization algorithm for nonlinear problems. In PSO, each particle represents a single intersection of all search dimensions. The particles change their positions according to the objective function
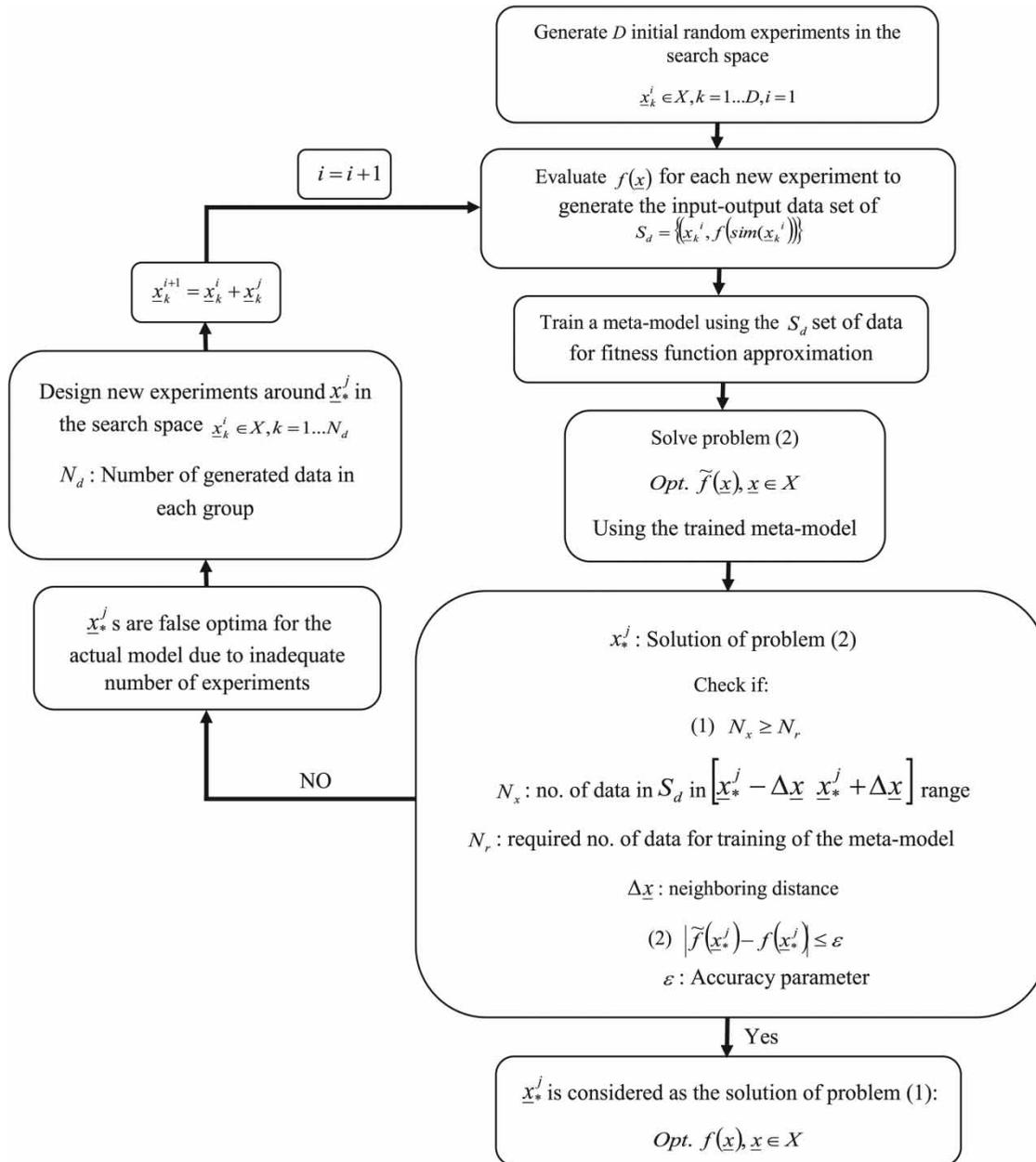
Figure 1 │ Different stages of the ASSF meta-modeling approach (Mousavi & Shourian 2010).

while sharing memories of their 'best' positions to adjust their velocities as well as their positions. In a D-dimensional search space, each particle i of the swarm is represented by a D-dimensional vector, $\underline{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. In each iteration, the particle remembers its previous best position, referred to as P_{best}, $\underline{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The velocity of each particle is represented by another D-dimensional

vector, $\underline{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The velocity of any particle is updated so that the particle moves toward its own previous best position, p_i, and the best position among all other particles positions, p_g.

Equations (4) and (5), respectively, represent the adjusted velocities and positions of particles in each iteration presented by Shi & Eberhart (1998) as a modification

to the original PSO formulas:

$$v_{id}^{n+1} = \chi(\omega v_{id}^n + c_1 r_1^n(p_{id}^n - x_{id}^n) + c_2 r_2^n(p_{gd}^n - x_{id}^n)) \qquad (4)$$

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1} \qquad (5)$$

where $d = 1, 2, \ldots, D$, $i = 1, 2, \ldots, N$, N is the size of the swarm, χ is a constriction factor that is used in constrained optimization problems in order to control the magnitude of the velocity, c_1, c_2 are two positive constants called cognitive and social parameters, respectively, r_1, r_2 are random numbers uniformly distributed in [0,1], $n = 1, 2, \ldots$ is the iteration number, and finally ω is the inertia weight that can be updated dynamically in every iteration by using Equation (6):

$$\omega(n) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{n_{\max}} \times n \qquad (6)$$

where $\omega(n)$ is the inertia weight at iteration n, $n_{\max}$ is the maximum number of iterations, and $\omega_{\max}$ and $\omega_{\min}$ are, respectively, the maximum and minimum inertia weights (Parsopoulos & Vrahatis 2002). The PSO algorithm starts with a set of randomly generated solutions. The swarm is then updated by using Equations (4) and (5) in each iteration. This process is continued until the stopping criteria are met.

One of the main problems encountered by optimization methods including the PSO algorithm is the problem of local minima, especially in multimodal functions. PSO, despite being an efficient method, also suffers from this problem. Function stretching is one of the techniques that can help omit undesirable local minima of the objective function. The idea behind function stretching is to perform a two-stage transformation of the original objective function. This can be applied immediately after a local minimum has been detected. The first transformation stage elevates the objective function and eliminates all the local minima that are located above the detected local minimum. The second stage stretches the neighborhood of the local minimum upwards by assigning higher function values to those points. Neither stage alters the local minima located below the detected minimum; thus, the location of the global minimum is left unaltered. More details on the stretching PSO (stretching particle swarm optimization (SPSO)) can be found in Parsopoulos & Vrahatis (2002) and Mousavi & Shourian (2010).

In this study, we have made use of the SPSO algorithm with the following function approximation techniques as meta-models in the ASSF approach.

FUNCTION APPROXIMATION TECHNIQUES

ANN

A neural network, which was first introduced by McCulloch & Pitts (1943), is a computational model that is loosely based on the neuron cell structure of the biological nervous system. Given a training set of data, the network can learn the data with a learning algorithm; once properly trained, the network provides a data-driven model that is capable of giving reasonable answers when presented with input vectors that have not been encountered during the training process.

In this study, we use feed-forward ANN to approximate the relation between decision variables and the resulting objective function of the system. Feed-forward networks (FFNs) are a subclass of layered networks in which there is no intra-layer connection whose main feature is that connections are allowed from a node in layer i only to nodes in layer $i + 1$ (Figure 2). Each of the hidden layer nodes (neurons) computes a weighted sum of inputs, passes the sum through the transfer (activation) function and presents the result to the next layer until the output layer is reached.
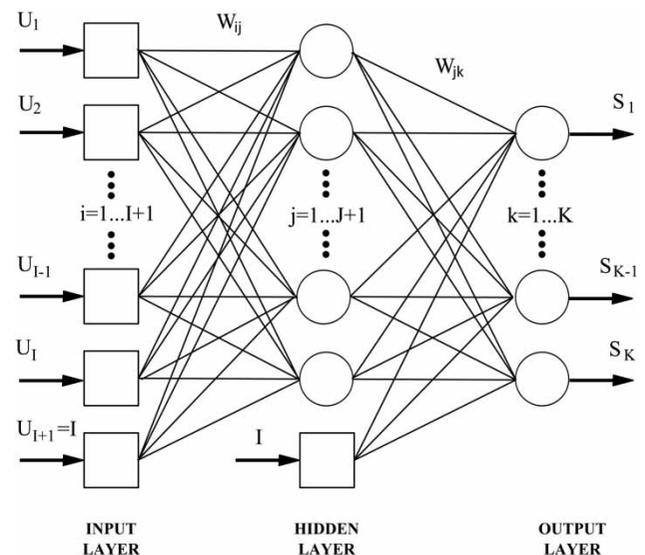

Figure 2 | Feed-forward neural network with one hidden layer.

We have used the back propagation (BP) with the Levenberg–Marquet training algorithm. More details on FFN and the BP algorithm may be found in different references (Hornik *et al.* 1989; Werbos 1994).

SVM

The SVM introduced by Vapnik (1998) is a robust and significant learning tool, which uses a learning bias derived from statistical learning theory. When SVM are employed for regression estimation, they are called support vector regression (SVR). The goal of learning process is to find a function $\tilde{f}(\underline{x})$ as an approximation of the value $f(\underline{x})$ with a minimum error based on available independent and identically distributed data as:

$$(x_1, y_1), \ldots, (x_m, y_m) \subseteq (X \subseteq R^n \times Y \subseteq R) \tag{7}$$

In SVR, an approximate function is determined by a small subset of training samples called support vectors. A specific loss function called *ε-insensitive* is developed to produce a sparseness property for SVR as follows (Vapnik 2000):

$$L(f, \tilde{f}(\underline{x}, \alpha)) = \left| f - \tilde{f}(\underline{x}, \alpha) \right|_\varepsilon$$
$$= \begin{cases} 0 & \text{if } \left| f - \tilde{f}(\underline{x}, \alpha) \right| \le \varepsilon \\ \left| f - \tilde{f}(\underline{x}, \alpha) \right| - \varepsilon & \text{otherwise} \end{cases} \tag{8}$$

where $\tilde{f}(\underline{x})$ is the approximate value of f, and the corresponding errors being less than *ε-boundary* (*ε*-tube) are not penalized (Figure 3). For linear function approximation, all the linear functions of input vector x have the following representation:

$$\tilde{f}(\underline{x}) = <w, \underline{x}> + \underline{b} \quad w, \underline{x} \in X \subset R^n, b \in R \tag{9}$$

where the angle bracket indicates the inner (or dot) product of two vectors in a Hilbert space. To find $\tilde{f}(\underline{x})$, it is necessary to minimize the regulated risk functional (R_{reg}) defined as follows:

$$R_{reg}[\tilde{f}] = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^m \left| f_i - \tilde{f}(\underline{x}_i)\right|_\varepsilon \text{ where } R_{emp}^\varepsilon[\tilde{f}]$$
$$= \frac{1}{m}\sum_{i=1}^m \left| f_i - \tilde{f}(\underline{x}_i)\right|_\varepsilon \tag{10}$$

where R_{emp} is the empirical error of training data that is defined in *ε-insensitive* loss function framework. Coefficient C in Equation (10) is a complexity indicator of function $\tilde{f}$.

Because real-world applications may require more expressive hypothesis spaces than linear functions, target functions cannot be explained just as a simple linear combination of the given attributes. Therefore, the considered hypothesis set could be functions of the below type:

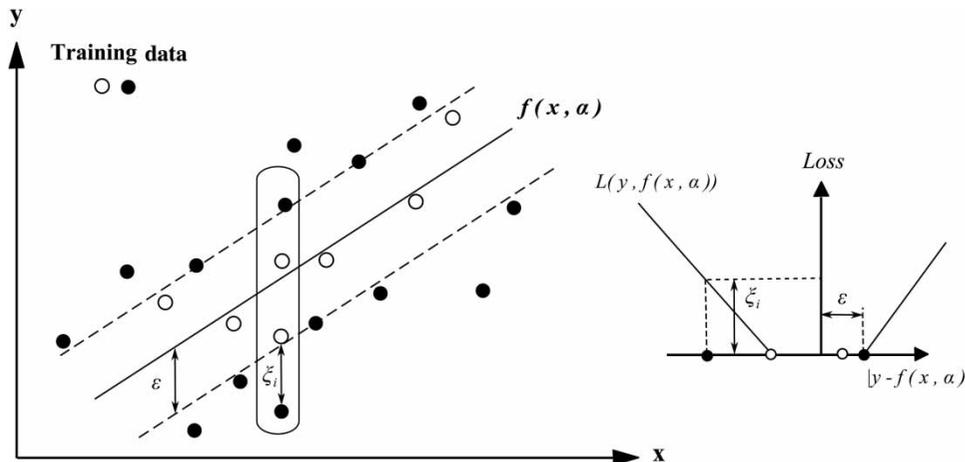$$\tilde{f}(\underline{x}) = <W, \varphi(x)> + \underline{b} \quad W \in F, b \in R \tag{11}$$

Figure 3 | *ε-insensitive* loss function in SVM.

where $\varphi: X \to F$ is a non-linear mapping from input space to some feature space. Finally, the following equation will be reached:

$$\tilde{f}(\underline{x}) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) \langle \varphi(\underline{x}_i), \varphi(\underline{x}) \rangle + \underline{b} \text{ where}$$

$$W = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) \varphi(\underline{x}_i) \tag{12}$$

If the inner product $<\varphi(x_i), \varphi(x)>$ is computed in feature space as a function of the original input points in a direct way, the explicit mapping of data will not be required. This implicit computation method is called a kernel method. A kernel function for all $x, z \in Z$ can be expressed as follows:

$$K(x, z) = <\varphi(x), \varphi(z)> \tag{13}$$

Linear kernel functions, polynomials, sigmoid functions and RBFs are some commonly used kernels. By substituting the inner product $<\varphi(x_i), \varphi(x)>$ with a suitable kernel function, the approximation function will be as follows (Vapnik 2000):

$$\tilde{f}(\underline{x}) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) K(\underline{x}_i, \underline{x}) + \underline{b} \tag{14}$$

More details on SVR may be found in Schölkopf & Smola (2002) and Cristianini & Shawe-Taylor (2000).

Kriging

Kriging is an interpolation method that predicts unknown values of a random function or random process. The theoretical basis for the method was developed by Matheron (1963) based on the Master's thesis of Krige (1951), a South African mining engineer. A kriging prediction is a weighted linear combination of all output values already observed. The weights depend on the distances between the new input and the old ones. A kriging model $f(\underline{x})$ is a combination of a global model and an additive localized approximation:

$$f(\underline{x}) = \tilde{f}(\underline{x}) + Z(\underline{x}) \tag{15}$$

where $\underline{x} \in R^n$ is a design variable, $\tilde{f}(\underline{x})$ is a known function of $\underline{x}$ as a global model of the original function, and $Z(\underline{x})$ is a Gaussian random function with zero mean and non-zero variance representing a local deviation from the global model. The regression model $\tilde{f}(\underline{x})$ appearing in Equation (19) can be written as:

$$\tilde{f}(\underline{x}) = \sum_{i=1}^{p} \beta_i f_i(\underline{x}) \tag{16}$$

where $f_i: R^n \to R$, $i = 1, \ldots, p$ are polynomial terms (typically of order 1 or 2), and, in many cases, they are reduced to constants. The coefficients β_i, $i = 1, \ldots, p$ are regression parameters. The covariance of $Z(\underline{x})$ is expressed by:

$$Cov[Z(\underline{x}^{(j)}), Z(\underline{x}^{(k)})] = \sigma^2 R[R(\theta, \underline{x}^{(j)}, \underline{x}^{(k)})], \quad j, k = 1, \ldots, N_s, \tag{17}$$

where N_s is the number of sample points, σ^2 is the so-called process variance, R is the correlation matrix, and $R(\theta, \underline{x}^{(j)}, \underline{x}^{(k)})$ is the correlation function between any two of the N_s samples $\underline{x}^{(j)}$ and $\underline{x}^{(k)}$ with unknown parameters θ. Note that R is an $N_s \times N_s$ symmetric matrix with diagonal elements equal to 1, and the form of the correlation function $R(\theta, \underline{x}^{(j)}, \underline{x}^{(k)})$ can be chosen by the user. Among the variety of correlation functions proposed in the literature, Gaussian correlation function is most frequently used. It is defined as follows:

$$R(\theta, \underline{x}^{(j)}, \underline{x}^{(k)}) = \exp\left(-\sum_{i=1}^{n} \theta_i \left| \underline{x}_i^{(j)} - \underline{x}_i^{(k)} \right|^2\right) \tag{18}$$

where $\theta = (\theta_1, \ldots, \theta_p)^T$ is the vector of correlation parameters. The kriging prediction of the fitness function value at any point $\underline{x}$ is given by the equation:

$$\tilde{f}(\underline{x}) = f^T(\underline{x})\hat{\beta} + r^T(x)R^{-1}(\underline{y} - F\hat{\beta}) \tag{19}$$

where $\hat{\beta}$ is the estimated value of β, $r^T(\underline{x}) = [R(\underline{x}, \underline{x}^{(1)}), \ldots, R(\underline{x}, \underline{x}^{(N_s)})]$ is the correlation vector evaluated at $\underline{x}$, y is the vector of responses to the sample locations $\{\underline{x}^{(1)}, \ldots, \underline{x}^{(N_s)}\}$, and F denotes the following matrix:

$$F = \begin{bmatrix} f^T(\underline{x}^{(1)}) \\ \cdot \\ \cdot \\ \cdot \\ f^T(\underline{x}^{(N_s)}) \end{bmatrix} \tag{20}$$

In this study, the kriging models are constructed by means of the DACE Toolbox. This environment provides both the kriging predictions and the related mean squared error (MSE) estimations (Lophaven *et al.* 2002) given by:

$$MSE[\tilde{f}(\underline{x})] = \sigma^2[1 + u^T(\underline{x})(F^T R^{-1} F)^{-1} u(\underline{x}) - r^T(\underline{x})R^{-1}r(\underline{x})]$$

$$\text{where} \quad u^T(\underline{x}) = F^T R^{-1} r(\underline{x}) - f(\underline{x}) \tag{21}$$

Polynomial response surface models

Response surface modeling techniques were originally developed by Box & Wilson (1951) to analyze the results of physical experiments to create empirically based models of the observed response values. Response surface models can be written in the following form:

$$\tilde{f}(\underline{x}) = f(\underline{x}) + \varepsilon \tag{22}$$

where $f(\underline{x})$ is the unknown function of interest, $\tilde{f}(\underline{x})$ is the polynomial approximation of $\underline{x}$, and ε is a random error that is assumed to be an independent, normally distributed variable with zero mean and variance σ^2. The polynomial function, $\tilde{f}(\underline{x})$, used to approximate $f(\underline{x})$ is typically a low order polynomial, which is assumed to be either linear, Equation (23), or quadratic, Equation (24):

$$\tilde{f} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i \tag{23}$$

$$\tilde{f} = \beta_0 + \sum_{i=1}^{k} \beta_i \underline{x}_i + \sum_{i=1}^{k} \beta_{ii} \underline{x}_i^2 + \sum_{i=1} \sum_{j<1} \beta_{ij} \underline{x}_i \underline{x}_j \tag{24}$$

where β_0, β_i, β_{ii} and β_{ij}, the parameters of the polynomials in Equations (23) and (24), are determined using the least-squares regression method.

Experimental setup

The ASSF approach is applied in three optimization problems including two benchmark functions optimization and an optimal water allocation problem at basin scale. We consider five different cases for the applied approach. In the first four cases, one of the aforementioned meta-modeling techniques including ANN, SVM, kriging and polynomial are incorporated to approximate the objective function considering the existing data points and in the last case the optimization algorithm (SPSO) is performed without considering the meta-modeling approach. The performance of the defined cases are then analyzed and compared based on two main criteria: (1) number of evaluations of the original function (NFE), which is the total number of evaluations of the original function in the algorithm including function evaluations used in DoE, during the optimization and those at the end of optimization to calculate final population; this gives indication of the required computation burden in this study; and (2) accuracy is an important indicator of a meta-model's performance. The accuracy metrics should reflect the deviation of the meta-model output $\tilde{f}$ from the output of the exact simulation model f. The MSE

$$MSE = \sum_{i=1}^{n} (f_i - \tilde{f}_i)^2 / n \tag{25}$$

provides a general evaluation of the overall prediction accuracy. We have used the MSE index in training, testing and validation stages.

The parameter values of meta-models, e.g. the regularization parameters in SVR and kriging, could have considerable impact on their predictive ability. A simple, yet effective way for selecting parameter values is to use k-fold cross-validation scheme. In k-fold cross validation, the data is first partitioned into k equally (or nearly equally) sized segment folds. Subsequently, k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation, and the remaining (k-1)-folds are used for learning. We have used a 10-fold cross validation scheme for selecting the best parameter values for each of the four meta-models. Under this scheme, the sample data are divided into 10 equally sized subsets and 10 iterations of training and validation tasks are then performed. In every iteration, the meta-models are trained with different parameter settings, and the parameter settings are evaluated with the validation set. The grid search is used to explore the entire parameter space, and the optimal parameter set with a minimum average

error on the validation set is found for each of the meta-models.

RESULTS

Benchmark optimization problems

The proposed methods are first tested in optimizing two well-known benchmark functions that have widely been used for evaluating the performance of meta-heuristic optimization algorithms and function approximation techniques while doing surrogate optimization. In fact, it is important to know how the performances of the considered meta-modeling techniques are affected by moving from a simpler two-dimensional (2D) optimization problem to a problem whose dimension is comparable with the real engineering design problem under study. Note that because of the need to calibrate and tune the parameter sets of any optimization or meta-modeling approaches, these types of tests are much more difficult to do on real engineering problems with computationally intensive simulation models, whereas they can easily be applied to benchmark optimization problems with fast objective function evaluations.

The first function is the Dejong function which serves as a good example of a continuous, convex, uni-model function. The second function is the Ackley function, known as a complex, multimodal function with a large number of local minima. The formulas of the Dejong and Ackley functions are defined below:

$$f(\underline{x}) = \sum_{i=1}^{n} \underline{x}_i^2 \quad -5.12 < \underline{x}_i < 5.12 \tag{26}$$

$$f(\underline{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}\underline{x}_i^2}\right)$$
$$- \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\Pi\underline{x}_i\right) + 20 + e \tag{27}$$

where n is the dimension of the functions. The global minimum of both functions is located at $\underline{x}^* = [0]$, $f(\underline{x}^*) = [0]$. Figure 4 illustrates the 2D Dejong (Dejong-2D) and 2D Ackley (Ackley-2D) functions:

The selected benchmark problems to be solved by the ASSF approach are Dejong-2D, Dejong-8D, Ackley-2D and Ackley-8D function optimization problems. The PSO swarm sizes were 10 and 30 and the PSO maximum iterations were 100 and 300 for 2D and 8D problems, respectively. More than one solution was determined by the ASSF approach because the best solution with the best objective value reached by the approach may not be the one having the best original objective function. Therefore, the results reported in Table 1 presents the best solutions among 10 runs of the SPSO algorithm. In this table, the approximate optimum function ($\tilde{f}(\underline{x}^*)$), the original fitness function ($f(\underline{x}^*)$) and the related error have been reported.
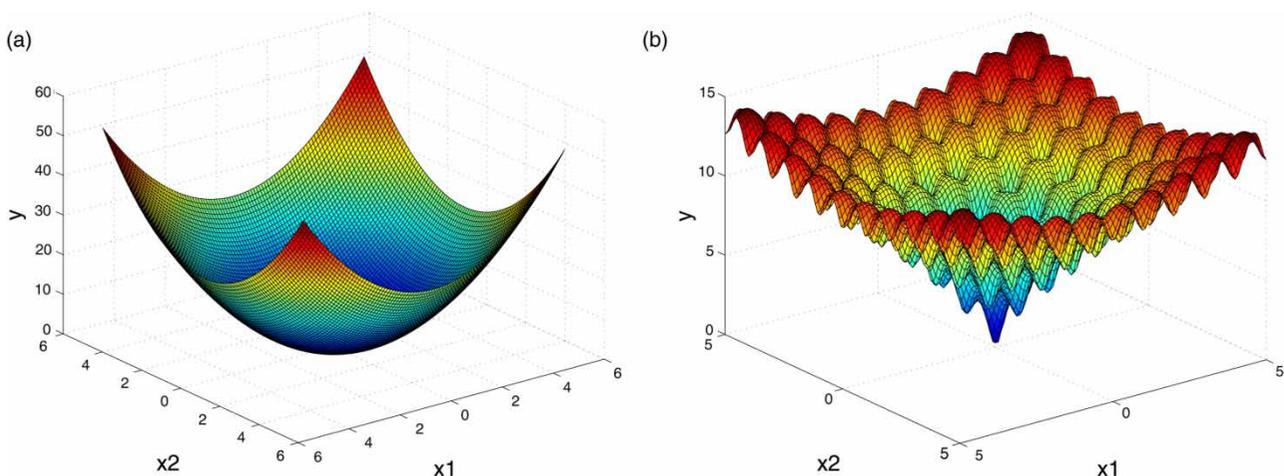

Figure 4 | (a) 2D Dejong function, (b) 2D Ackley function.

**Table 1** | Results of the function minimization problems using the ASSF approach for different meta-models

	Optimum point located	Approximate obj. function ($\tilde{f}(x^*)$)	Original obj. function ($f(x^*)$)	MSE	Number of evaluations of the original function (NFE)
Dejong-2D					
SPSO	[0 0]	–	0	–	1,000
SVM	[0 0]	−0.01	0	0.01	341
ANN	[0 0]	0.0419	0	−0.0419	627
Kriging	[0 0]	-8.84×10^{-7}	0	-8.84×10^{-7}	346
Polynomial	[0 0]	-4.34×10^{-15}	0	4.34×10^{-15}	306
Dejong-8D					
SPSO	[00000000]	–	0	–	9,000
SVM	[00000000]	−0.0408	0	0.0408	1,159
ANN	[00000000]	8.14×10^{-4}	0	−0.0114	2,021
Kriging	[−0.0636−0.029−0.0596−0.1245 0.0166 0.0023 0.006−0.0270]	-5.2683×10^{-4}	0.8854	0.8859	2,346
Polynomial	[00000000]	-2.329×10^{-3}	0	-2.329×10^{-3}	1,201
Ackley-2D					
SPSO	[0 0]	–	0	–	1,000
SVM	[0 0]	0.1672	0	−0.1672	256
ANN	[0 0]	0.159	0	−0.159	494
Kriging	[0 0]	0.0896	0	−0.0896	667
Polynomial	[0 0]	0.9987	0	−0.998	2,893
Ackley-8D					
SPSO	[00000000]	–	0	–	9,000
SVM	[00000000]	0.00275	0	0.00275	1,956
ANN	[00000000]	0.2025	0	−0.2025	6,215
Kriging	[−0.0175 0.0779−0.0046−0.006 0.05 0.0776−0.0125 0.1071]	0.07981	0.07869	−0.012	8,345
Polynomial	[−0.2548−0.2732−0.0562−0.4618 0.1307−0.1127 0.3673−0.1004]	4.7988	2.7613	−2.0376	9,689

Figures 5 and 6 compare the predictions (meta-model's outputs) vs. the actual values (targets) of the objective function for Dejong-8D and Ackley-8D benchmark problems, respectively (figures of 2D functions are not shown due to the lack of space).

Dejong function is a well-behaved, convex, uni-model function. For the case of Dejong-2D function, the ASSF approach with all the meta-models has successfully located the optimum point ($x^* = [0]_{1 \times 2}$) with minimum and maximum NFE associating with polynomial and ANN meta-models, respectively. However, SVM and kriging have maintained almost the same NFE, which means that these two meta-models have performed almost the same. Polynomial, SVM and kriging meta-models

have been able to find the global minimum with NFE less than 350 function evaluations, whereas ANN has needed more function evaluations. The reason why ANN has required a larger NFE to optimize such a simple function is consistent with the fact that larger sets of design points are typically required to train neural networks (Yan & Minsker 2006; Zou et al. 2007), and at the same time the adaptively updating mechanism of the ASSF approach plays a less significant role in low-dimensional search spaces.

For the case of Dejong-8D function, the SVM, ANN and polynomial meta-models have successfully located the optimum point ($x^* = [0]_{1 \times 8}$) with minimum NFE and 4.08% error percentage belonging to SVM (Table 1). For the
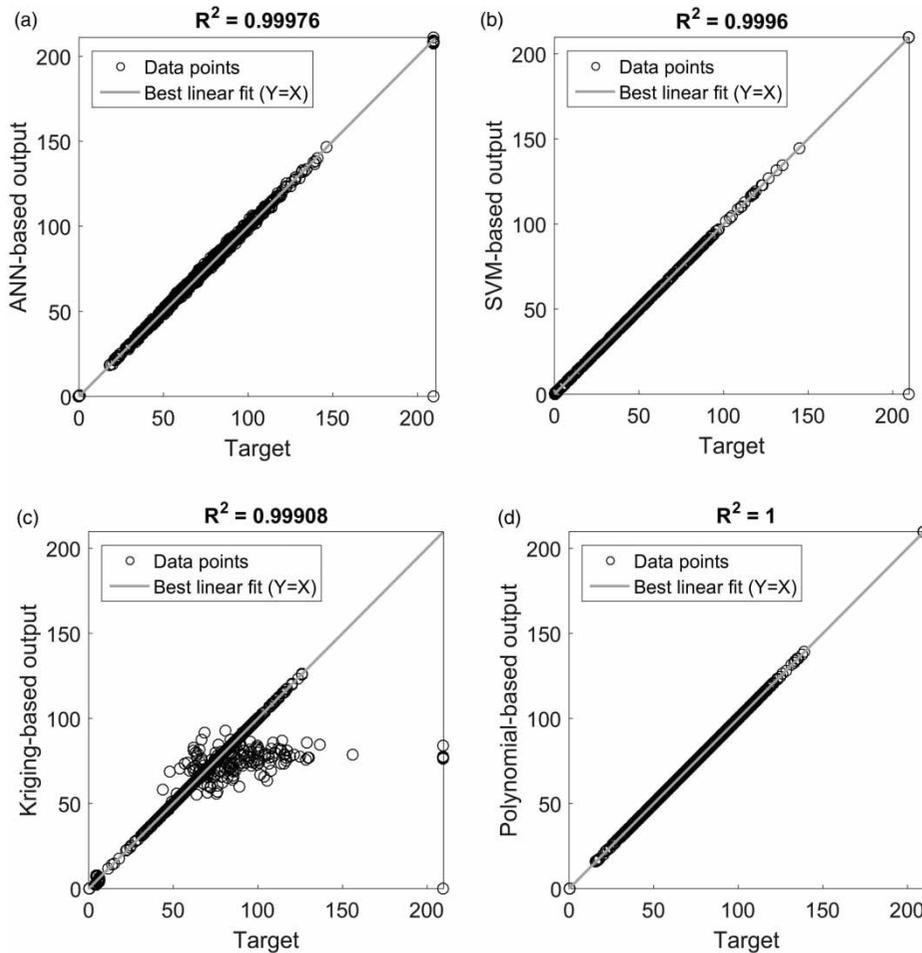
Figure 5 | (a) ANN, (b) SVM, (c) kriging, (d) polynomial-based predictions (outputs) vs. actual values (targets) of the objective function (Dejong-8D) with the obtained R^2 equal to 0.99976, 0.9996, 0.99908 and 1, respectively.

smooth function of Dejong, simple learning techniques such as polynomial response functions are enough, and more complicated techniques of ANN and kriging, requiring more data points to be properly trained, are not necessary.

The Ackley function is a more complex, non-smooth function containing numerous local minima with no detectible trend toward the global region of attraction. For the case of Ackley-2D function, the polynomial meta-model has required a larger NFE to reach the optimum point ($\underline{x}^* = [0]_{1\times2}$), and SVM has outperformed the others with the smallest NFE equal to 256 followed by ANN with 494. Therefore, SVM and ANN surrogates have performed better for the more complex function optimization problem. The polynomial meta-model has failed to reach the optimum point due to the limitation in the random access memory (RAM) of the computer used.

Atrak river basin water allocation problem

SPSO-MODSIM model

To test the efficacy of the ASSF approach and different meta-models in a real problem, we consider the optimum water allocation problem at the Atrak river basin in northeast Iran (54° to 54°04′ E; 38° 17′ to 38° 57′ N). Elevation ranges from −22 m.a.s.l. at the Caspian Sea southern coast to 2,903 m.a.s.l. at the highest point in the easternmost parts of the basin. The Atrak river is one of the longest rivers of the country with a total length of about 25,627 km and the drainage area of 520 km² inside Iran and a small part within the territory of Turkmenistan (Sheikh 2014). Figure 7 is a schematic map of the river basin showing the locations
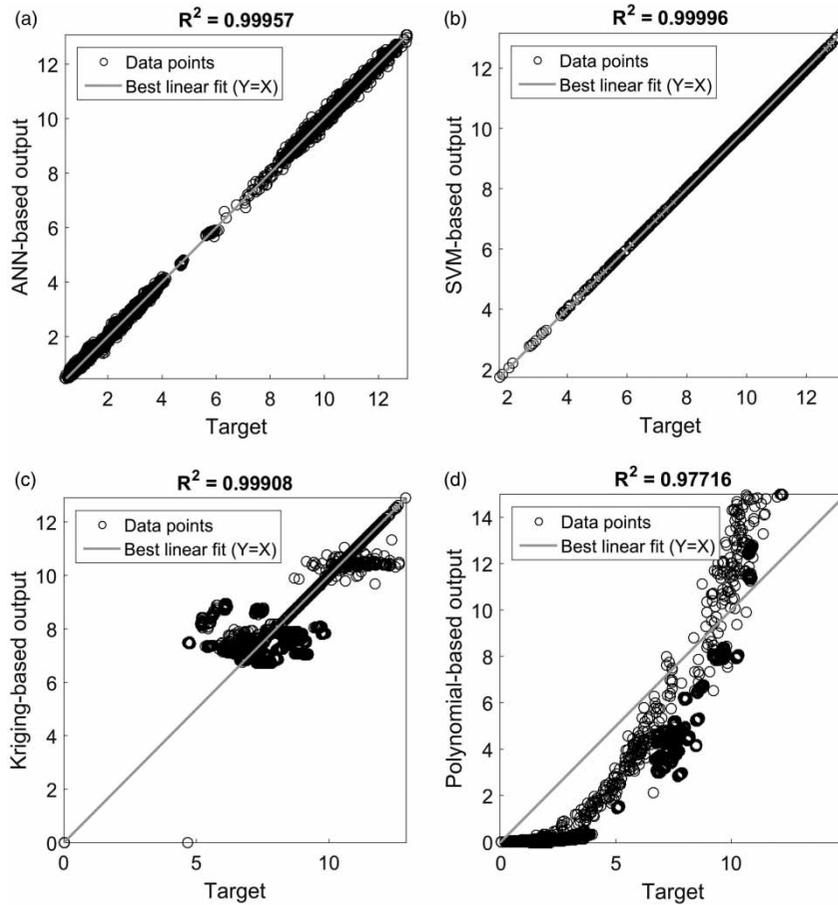
Figure 6 | (a) ANN, (b) SVM, (c) kriging, (d) polynomial-based predictions (outputs) vs. actual values (targets) of the objective function (Ackley-8D) with the obtained R^2 equal to 0.99957, 0.99996, 0.99908 and 0.97716, respectively.

of reservoirs. There are 12 reservoirs among which Ghordanlu, Darband, Garmab, Amand, Chaily, Chandir and Sumbar have not been constructed yet and, therefore, it is necessary to decide their capacity.

In the proposed development plan of the basin, construction of new storage dams and water transfer projects is being studied. To simulate different complex institutional, hydrologic and socio-economic processes taking place in a river basin system, a comprehensive DSS may be used as a simulator engine. We have selected MODSIM, a generalized river basin simulation model developed by John W. Labadie at Colorado State University (Labadie 1995), for this purpose. MODSIM is a valuable tool capable of analyzing the operation of complex river basin systems as a network consisting of nodes and links. To do the task of water allocation optimally, MODSIM sequentially solves the following one–period, linear optimization problem in each time step over

the planning horizon using an efficient minimum cost network flow program (NFP):

$$min \sum_{l \in A} c_l q_l$$

Subject to:

$$\sum_{j \in Q_i} q_j - \sum_{k \in I_i} q_k = 0; \qquad (28)$$

for all $i \in N$

$$l_l \leq q_l \leq u_l; \quad \text{for all} \quad l \in A$$

In the above formula, l is the set of all arcs or links in the network, N is the set of all nodes, Q_i is the set of links originating at node i (i.e. outflow links), I_i is the set of links terminating at node i (i.e. inflow links), q_l is the integer-valued flow rate in link l, c_l is the cost, weighting factor or priority per unit flow rate in each link l, l_l is the lower bound on the flow in link l, and u_l is the upper bound on the flow in link l. Further, the mass balance
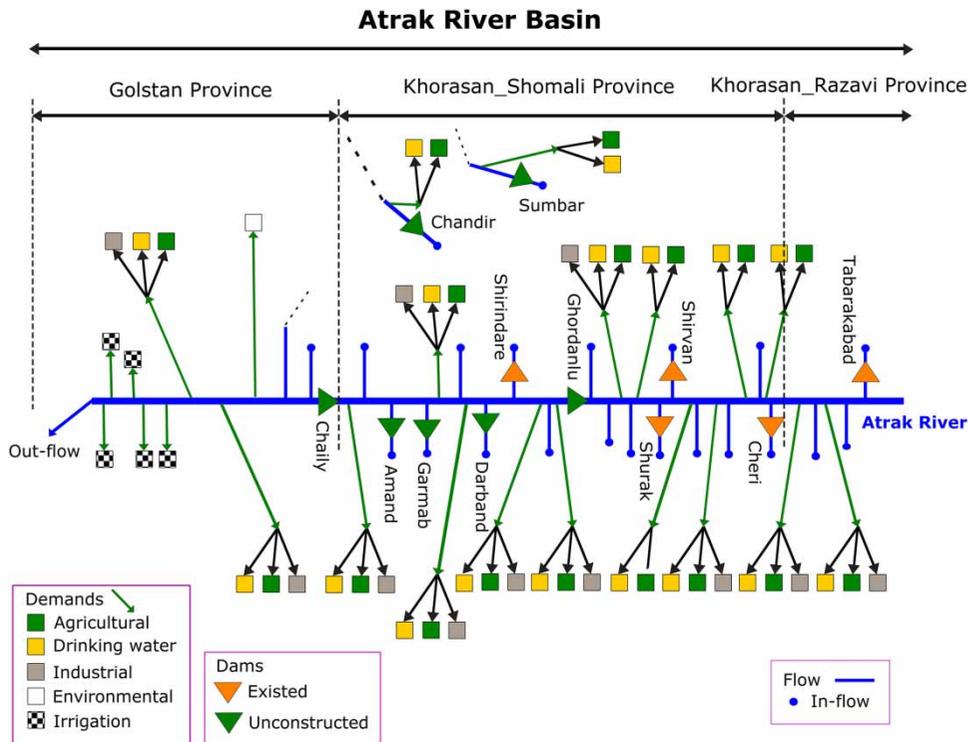
Figure 7 │ Schematic representation of the Atrak river basin.

equation must be satisfied at every node of the model. It is possible in MODSIM to define a cost coefficient for each connection link per unit of water transferred. Assigning a negative coefficient shows the benefit and a positive coefficient represent the cost of water transferred to the ending node of the link.

Despite all MODSIM's remarkable capabilities, it is not a full-dynamic, multi-period optimization model. Consequently, MODSIM, by itself, would not be able to determine an optimum design or operation policies. On the other hand, one of the features of MODSIM's latest versions is the ability of preparing customized codes in the VB.NET or C.NET (used in this study) languages compiled with MODSIM through the .NET Framework. Therefore, it is possible to link MODSIM with the PSO algorithm and use it as an efficient simulation optimization approach for solving a variety of large scale, basin-wide water resource problems (Shourian *et al.* 2008, 2009). Taking advantages of MODSIM's custom coding features, one can embed MODSIM in the SPSO algorithm and build an efficient simulation optimization tool to solve the Atrak river basin water allocation optimization problem. Additionally, because MODSIM is relatively a high-fidelity simulation model with several NPFs to be solved in a large,

complex river basin, the resulting SPSO-MODSIM model is computationally intensive; thus it can be considered as an appropriate application of the ASSF approach.

Determining the sizes of water resource development projects and deciding how to allocate available water resources among different demand nodes over time and space while considering coordinated operation of the system components are important to a basin management. These issues in the Atrak river basin system, which is under development by constructing a number of dams and irrigation projects as well as already-constructed reservoirs being operated, can be dealt with by formulating the problem as a large-scale, simulation optimization model.

Relative priority numbers assigned to target reservoirs' storage levels can be taken as operational variables, according to which the NFP employed in MODSIM determines whether to release water stored in a reservoir in each time period to meet that period's water demand or to keep it in the reservoir for future uses. Therefore, there are two main types of decision variables including design variables, consisting of the capacities of unconstructed reservoirs or water transfer components, and operational variables, i.e. relative priority

Table 2 │ Minimum and maximum values of design variables

Dam	Darband	Garmab	Ghordanlu
Maximum	10 mcm	10 mcm	50 mcm
Minimum	4 mcm	5 mcm	5 mcm

numbers of target reservoirs' storage levels of both unconstructed and constructed dams.

In the case of Atrak river basin, we have considered a six-dimensional problem with the capacities of Darband, Garmab and Ghordanlu reservoirs as design decision variables, and the priorities of these three dams as operational variables. Other reservoirs have smaller potential capacities, so their capacities are assumed to be known. Considering the geologic, topographic and hydrologic conditions, the ranges of capacities are 5 million cubic meters (mcm) to 10 mcm for Darband and Garmab reservoirs and between 5 mcm to 50 mcm for Ghordanlu reservoir (Table 2). Municipal and agricultural water demands have been estimated based on the projection of future population growth and agricultural land and crop patterns at year 2025. Other hydrologic input data including natural and sub-basin river flows, evaporation from and precipitation in to lakes have been gathered and introduced to the model over a historical 20-year period (1976 to 1996) of available records to account for hydrologic variability.

The optimization model would suggest what capacities the reservoirs should have considering hydrologic and economic factors. However, due to lack of information on economic estimations related to costs of dam constructions and benefits of supplying water to demand nodes, we simply formulate the objective function as a weighted sum of unconstructed reservoir capacities and the amounts of water shortages at the demand nodes in Golestan and Khorasan-Shomali Provinces as follows:

$$cost = w_1 \times$$

$$\left(\sum_{j=1}^{nDEM_{Kh-Shomali}} Q_{supplyKh-Shomali} - TARGET_{Kh-Shomali} \right)^2 + w_2$$

$$\times \left(\sum_{j=1}^{nDEM_{Golestan}} Q_{supplyGolestan} - TARGET_{Golestan} \right)^2$$

$$+ w_3 \times (DamCap_{Darband}) + w_4 \times (DamCap_{Garmab})$$

$$+ w_5 \times (DamCap_{Ghordanlu}) \tag{29}$$

where:

$Q_{supplyKh-Shomali}$: water volume supplied to demand nodes in Khorasan-Shomali Province.

$Q_{supplyGolestan}$: water volume supplied to demand nodes in Golestan Province.

$TARGET_{Kh-Shomali}$: target water demand volume of Khorasan-Shomali Province.

$TARGET_{Golestan}$: target water demand volume of Golestan Province.

Although $TARGETKh\_Shomali$ and $TARGET_{Golestan}$ are just two input parameters in the above function, the optimization model results and, therefore, the shares of water allocations to each of the provinces are quite sensitive to these parameters' values. On the other hand, the competition between the two upstream (Khorasan-Shomali) and downstream (Golestan) provinces, as the main water consumers of the system, is a source of conflict over these parameter values. Each of the provincial parties does their best to convince the members of a national, third-party committee, among whom the second author is included, to recognize their water requirements are as large as possible. In fact, each province may tend to report its own water demand as larger than reality in order to attract more water to the province's demand sites. Specifically, there are a number of newly developed irrigation lands in Khorasan-Shomali Province whose water supply is severely criticised by the downstream Golestan provincial administration. In other words, due to severe water shortages in the downstream Golestan Province, stakeholders there are insisting on the cessation of any development projects in the upstream Khorasan-Shomali Province including the irrigation schemes as well as any under-study dam reservoirs.

Facing such a critical dispute, we, as a member of a national, third party committee, ran MODSIM model over a 20-year historical period under a water demand condition in which none of the elements of irrigation and surface reservoir development projects were put into operation. Under this condition, the volumes of water supplied to different demand nodes of the two provinces, i.e. Khorasan-Shomali and Golestan, were determined as 756.5 mcm and 195 mcm, respectively. These amounts were then set as $TARGET_{Kh\_Shomali}$ and $TARGET_{Golestan}$ input parameters to the model in subsequent runs under an increased water demand condition due to new irrigation development projects as well as new dam reservoirs

to be optimally sized. In this condition, the total average annual demands of the two provinces were equal to 1,202.3 and 531.6 mcm, respectively. However, a higher priority was given to meeting the portion of water demand of each province that had been established historically (first-priority demands) than that given to the part established due to more recently developed projects (second-priority demands).

By the above-described mechanism, the SPSO outer objective function (Equation (29)) drives the search toward the solutions that best meet the two objectives of minimizing the capacities of new dam projects as well as the shortages in supplying water to first-priority demands, and the one-period, minimum-cost objective functions of the MODSIM's NFPs will guide the model to deliver any excess water to second-priority water demands. This procedure is, in fact, a methodological setting of the first-in-time first-in-right water allocation mechanism combined with the riparian allocation mechanism in our modeling exercise. By this approach, the national, third-party committee was able to moderately reduce some disagreements over actual water demands between the two main provinces in the Atrak river basin.

The SPSO-MODSIM model was then used to solve the 8D problem presented above. The values of weighs w_1, w_2, w_4, w_4 and w_5 were, respectively, set to 1, 1, 100, 100 and 1 by using a trial and error based procedure. The number of particles and the maximum number of iterations were set as 30 and 3,000, respectively. The stopping criteria were either reaching the maximum number of iterations or no improvement of the SPSO best objective function value

Table 3 | Results of the SPSO-MODSIM model for the Atrak river basin problem

Optimized variables	Optimized values
Darband reservoir capacity (mcm)	5
Garmab reservoir capacity (mcm)	5
Ghordanlu reservoir capacity (mcm)	40
Darband target storage level's priority number	1
Garmab target storage level's priority number	3
Ghordanlu target storage level's priority number	10
Objective function	137,837

over 200 successive iterations. Other SPSO parameters were $w_{\min} = 0.1$, $w_{\max} = 1.2$, $C_1 = 1.75$ and $C_2 = 1.25$. The upper and lower bounds of priority numbers in MODSIM were, respectively, equal to 10 and 1. Table 2 presents the minimum and maximum values of the design variables introduced to the model. Note that the system used was a PC 3.33 GHz, Intel(R) Core(TM)2 Duo CPU, with 10 GB RAM, for which the regular time needed for each function evaluation (MODSIM run) was about 20 seconds.

Table 3 reports the best solution obtained by the SPSO-MODSIM model with a total number of function evaluations of 30,000 because of the dominance of the second stopping criterion. Due to stochastic nature of PSO algorithm, the results reported in this section (Tables 3 and 4) present the best solutions among 10 runs of the SPSO algorithm.

Figure 8 compares the monthly supply and demand values for the two competing principal provinces, resulted from the PSO-MODSIM model.

Table 4 | Comparative analysis of different meta-models employed in the ASSF approach for solving the Atrak river basin water allocation problem

| Method	SPSO-MODSIM	SPSO-MODSIM ∼ ANN	SPSO-MODSIM ∼ SVM	SPSO-MODSIM ∼ KRIGING	SPSO-MODSIM ∼ POLYNOMIAL
Approximate best obj. func. value	–	137,844	137,830	137,836	137,577
Original obj. func. value	137,837	137,837	137,837	137,837	137,837
MSE	–	-5.08×10^{-5}	5.37×10^{-5}	1.05×10^{-5}	0.00188
Number of original obj. function evaluations	30,000	482	477	1,500	1,667
Darband reservoir capacity (mcm)	5	5	5	5	5
Garmab reservoir capacity (mcm)	5	5	5	5	5
Gordanlu reservoir capacity (mcm)	40	40	40	40	40
Darband target storage level's priority number	1	1	3	6	4
Garmab target storage level's priority number	3	3	4	8	6
Ghordanlu target storage level's priority number	10	5	3	4	1
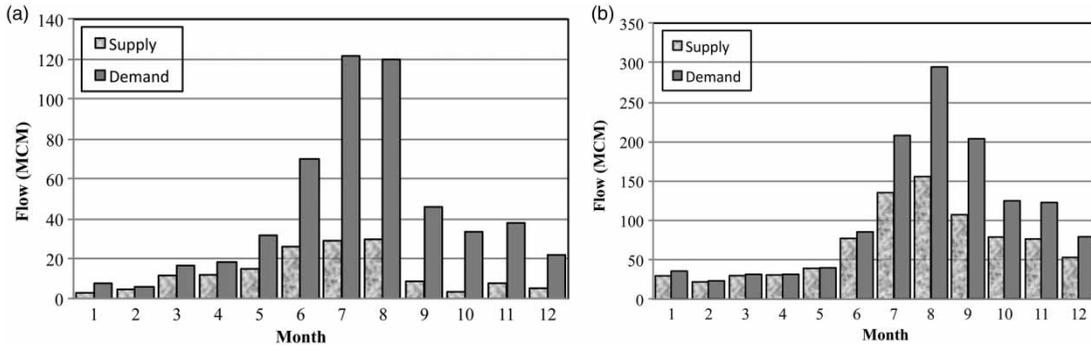
Figure 8 │ Monthly supply and demand values for (a) Golestan Province and (b) Khorasan-Shomali Province.
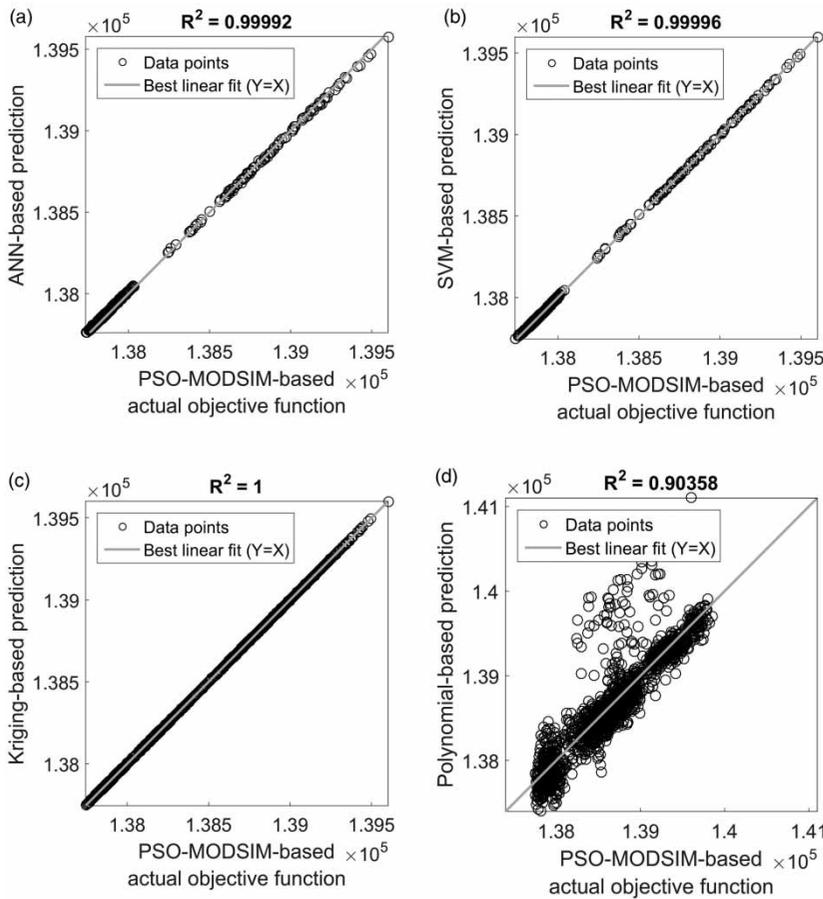

Figure 9 │ (a) ANN, (b) SVM, (c) kriging, (d) polynomial-based predictions at the last iteration of the ASSF approach (*y*-axis) vs. SPSO-MODSIM-based actual objective function values (*x*-axis).

Analysis of meta-models in Atrak river basin problem

For the case of Atrak water allocation problem, all the meta-models have started with 200 initial random samples as training data. Figure 9 illustrates the predicted vs. actual

objective function values resulted from different meta-models in the last iteration.

According to Table 4, all meta-models with fewer original function evaluations have found solutions similar to that of SPSO-MODSIM with the best objective function

value equal to 137,837. Similar to the Ackley-8D problem, SVM and ANN outperformed the other two meta-models with respect to NFE required, whereas kriging and polynomial have needed a larger NFE to reach the solution. Although the polynomial meta-model with higher MSE value has not been as accurate as the other models, it is a relatively simple, efficient technique needing much less learning time.

SUMMARY AND CONCLUSIONS

Incorporating a river basin simulation model in a heuristic optimization algorithm can help modelers address basin-scale water resource management problems more efficiently via an integrated simulation-optimization approach. The stretching PSO (SPSO) and the MODSIM river basin DSS were linked, and the SPSO-MODSIM tool was used to solve a large-scale, complex water allocation problem at Atrak river basin in Iran with a major dispute over water among upstream and downstream stakeholders. We estimated, by MODSIM, the amount of water supplied historically to two competing provincial water consumers over a long period as the first-priority water demands, and the demands in excess of that part, which were related to more recently established water resource development projects, were given a less priority. The SPSO objective function drives the search toward the solutions that best meet the two objectives of minimizing the capacities of new dam construction projects as well as the shortages in supplying water to first-priority demands, and the one-period, minimum-cost objective function of the MODSIM NFPs guide the solutions to deliver any excess water to second-priority water demands.

Because of high computational cost of the SPSO-MODSIM model, we, subsequently, considered the use of the adaptive sequentially space filing (ASSF) meta-modeling approach, where MODSIM was replaced by four types of meta-models, i.e. ANN, SVM, kriging and polynomial response functions. To test how well these approximation techniques perform, the performances of the techniques were first compared through two benchmark function optimization problems. The meta-models were then evaluated by solving the real-world Atrak river basin water

allocation optimization problem in Iran. The results of the ASSF approach with different meta-models were compared to those of the exact SPSO-MODSIM model as a basis to test if the proposed meta-models can replace the computationally intensive SPSO-MODSIM model.

In the benchmark problems, all the meta-models were able to achieve accurate solutions while saving the number of function evaluations (NFE) for both benchmark functions of Ackley and Dejong. The exact global optimum solution was found by all the meta-models for the 2D problems; however, kriging was able to locate just a good solution, not the global optimum, for the 8D problems. Additionally, polynomial and SVM meta-models performed better than others by locating the global optimum of the Dejong functions (2D and 8D) with fewer NFEs, whereas for more complex Ackley functions (2D and 8D), the SVM and ANN meta-models were better than the others.

For the Atrak river basin problem, all the meta-models were able to locate the best solution reached by the exact, computationally intensive SPSO-MODSIM model with significantly fewer NFEs compared with the exact model. This is mainly because of the fact that the ASSF approach sequentially improves the performance of any types of meta-models in the course of optimization by adaptively locating the promising regions of the search space where more samples need to be generated for meta-models training. However, ANN and SVM performed better than other meta-models in saving the number of costly, original function evaluations. The performances of SVM and ANN were almost the same, but SVM was slightly better.

The results show that simpler meta-models such as polynomial, which does not require a sophisticated training algorithm, could perform well enough for less-complex functions. However, the meta-models with more specific training algorithms such as ANN and SVM are needed for solving more complex problems with larger decision variables and highly non-linear response functions. Moreover, ANN requires a large number of design points to be trained even though they are learning a relatively simple function (Yan & Minsker 2006; Zou et al. 2007). Therefore, the adaptively updating mechanism of the ASSF approach is less significant when it comes to simple, low-dimensional function optimization problems. That is why ANN needed about 62% and 22% of exact

NFE for Dejong-2D and Dejong-8D functions, respectively, compared with 30.6 and 13.4% obtained by the polynomial meta-model.

Another point to have in mind is that ANN meta-models require the selection of appropriate network structures with a large number of parameters (weights and biases) to be adjusted (Maier & Dandy 2000), which needs more expertise. SVM and kriging also have regularization parameters with considerable impact on their performance that need to be determined by the users prior to the training process. It is worth noting that one reason for the performance of kriging not being as good as other meta-models, as mentioned by Razavi et al. (2012a), could be the fact that the DACE toolbox involves a global search method that is not efficient in improving the solution in the main region of attraction, and mostly focuses on surpassing the misleading valleys over the non-informative regions. Moreover, for the Ackley-8D problem, the polynomial meta-model failed to reach the optimum point due to the limitation in the RAM of the PC used. This difficulty can, however, be solved by accessing PCs with higher RAM capacities although larger NFE would be required.

Finally, more research is needed to explore the potential advantages of using other sampling techniques, such as Latin hypercube, and other meta-models including RBF.

ACKNOWLEDGEMENTS

REFERENCES

Blanning, R. W. 1975 The construction and implementation of metamodels. Simulation 24, 177–184.

Bonner, V. 1989 HEC-5: Simulation of flood control and conservation systems (for microcomputers). Model-Simulation, Hydrologic Engineering Center, Davis, CA, USA.

Box, G. E. & Wilson, K. 1951 On the experimental attainment of optimum conditions. J. R. Stat. Soc. Series. B. Stat. Methodol. 13, 1–45.

Cai, X., McKinney, D. C. & Lasdon, L. S. 2002 A framework for sustainability analysis in water resources management and application to the Syr Darya Basin. Water Resour. Res. 38, 21-1–21-14.

Cristianini, N. & Shawe-Taylor, J. 2000 An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge, UK.

DHI Water, Environment 2006 MIKE Basin Simulation Model: A Versatile Decision Support Tool for Integrated Water Resources Management and Planning. Horshelm, Denmark, January 23, 2006.

Draper, A. J., Munévar, A., Arora, S. K., Reyes, E., Parker, N. L., Chung, F. I. & Peterson, L. E. 2004 Calsim: generalized model for reservoir system analysis. J. Water Resour. Plan. Manage.-ASCE 130, 480–489.

El-Beltagy, M. A., Nair, P. B. & Keane, A. J. 1999 Metamodeling Techniques for Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations. Gecco-99, Orlando, USA.

Farhang-Mehr, A. & Azarm, S. 2005 Bayesian meta-modelling of engineering design simulations: a sequential approach with adaptation to irregularities in the response behaviour. Int. J. Numer. Methods Eng. 62, 2104–2126.

Hornik, K., Stinchcombe, M. & White, H. 1989 Multilayer feedforward networks are universal approximators. Neural Netw. 2, 359–366.

Hydraulics, D. 1991 RIBASIM River Basin Simulation. Project completion report to Water Resources Commission, Taipei, Taiwan.

Jenkins, M. W., Lund, J. R., Howitt, R. E., Draper, A. J., Msangi, S. M., Tanaka, S. K., Ritzema, R. S. & Marques, G. F. 2004 Optimization of California's water supply system: results and insights. J. Water Resour. Plan. Manage.-ASCE 130, 271–280.

Jin, Y. C., Olhofer, M. & Sendhoff, B. 2002 A framework for evolutionary optimization with approximate fitness functions. IEEE Trans. Evol. Comput. 6, 481–494.

Jones, D. R. 2001 A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. 21, 345–383.

Kennedy, J. & Eberhart, R. 1995 Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942–1948.

Krige, D. 1951 A statistical approach to some mine valuation and allied problems on the Witwatersrand. M.Sc. in Engineering, University of the Witwatersrand.

Kuczera, G. & Diment, G. 1988 General water supply system simulation model: WASP. J. Water Resour. Plan. Manage.-ASCE 114, 365–382.

Labadie, J. 1995 MODSIM: River basin network flow model for conjunctive stream-aquifer management, Fort Collins.

Liang, K. H., Yao, X. & Newton, C. 2000 Evolutionary search of approximated N-dimensional landscapes. Int. J. Knowledge Based Intel. 4, 172–183.

Lophaven, S. N., Nielsen, H. B. & Søndergaard, J. 2002 DACE-A Matlab Kriging toolbox, version 2.0, Kgs. Lyngby, Denmark.

Maier, H. R. & Dandy, G. C. 2000 Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environ. Modell. Softw.* **15**, 101–124.

Matheron, G. 1963 Principles of geostatistics. *Econ. Geol.* **58**, 1246–1266.

McCulloch, W. S. & Pitts, W. 1943 A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133.

Mousavi, S. J. & Shourian, M. 2010 Adaptive sequentially space-filling metamodeling applied in optimal water quantity allocation at basin scale. *Water Resour. Res.* **46**, W03520.

Nair, P. B. & Keane, A. J. 2001 Passive vibration suppression of flexible space structures via optimal geometric redesign. *AIAA J.* **39**, 1338–1346.

Parsopoulos, K. E. & Vrahatis, M. N. 2002 Recent approaches to global optimization problems through particle swarm optimization. *Nat. Comput.* **1**, 235–306.

Ratle, A. 2001 Kriging as a surrogate fitness landscape in evolutionary optimization. *Artif. Intell. Eng. Design Anal. Manufactur.* **15**, 37–49.

Razavi, S., Tolson, B. A. & Burn, D. H. 2012a Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environ. Modell. Softw.* **34**, 67–86.

Razavi, S., Tolson, B. A. & Burn, D. H. 2012b Review of surrogate modeling in water resources. *Water Resour. Res.* **48**, W07401, Doi:10.1029/2011WR011527.

Regis, R. G. & Shoemaker, C. A. 2007 Parallel radial basis function methods for the global optimization of expensive functions. *Eur. J. Oper. Res.* **182**, 514–535.

Robinson, G. M. & Keane, A. J. 1999 A case for multi-level optimisation in aeronautical design. *Aeronaut. J.* **103**, 481–485.

Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. P. 1989 Design and analysis of computer experiments. *Stat. Sci.* **4**, 409–423.

Schölkopf, B. & Smola, A. J. 2002 *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, USA.

SEI 1999 *WEAP: Water Evaluation and Planning*. Tellus Institute, Boston.

Sheikh, V. 2014 Analysis of hydroclimatic trends in the Atrak River basin, North Khorasan, Iran (1975–2008). *Int. J. Environ. Res.* **2**, 233–246.

Shi, Y. & Eberhart, R. 1998 A modified particle swarm optimizer. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence*. IEEE, Anchorage, AK, USA, pp. 69–73.

Shoemaker, C. A., Regis, R. G. & Fleming, R. C. 2007 Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation. *Hydrol. Sci. J.* **52**, 450–465.

Shourian, M., Mousavi, S. J. & Tahershamsi, A. 2008 Basin-wide water resources planning by integrating PSO algorithm and MODSIM. *Water Resour. Manag.* **22**, 1347–1366.

Shourian, M., Mousavi, S., Menhaj, M. & Jabbari, E. 2009 Neural-network-based simulation-optimization model for water allocation planning at basin scale. *J. Hydroinform.* **10**, 331–343.

Sigvaldson, O. 1976 A simulation model for operating a multipurpose multireservoir system. *Water Resour. Res.* **12**, 263–278.

Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch, P.N. & Yang, R. J. 2004 Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Struct. Multidiscip. Optim.* **27**, 302–313.

Tsoukalas, I. & Makropoulos, C. 2015 Multiobjective optimisation on a budget: exploring surrogate modelling for robust multi-reservoir rules generation under hydrological uncertainty. *Environ. Modell. Softw* **69**, 396–413.

Vapnik, V. 1998 *Statistical Learning Theory*. Wiley, New York.

Vapnik, V. 2000 *The Nature of Statistical Learning Theory*. Springer, New York, USA.

Werbos, P. J. 1994 *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. John Wiley & Sons, New York.

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

Zagona, E. A., Fulp, T. J., Shane, R., Magee, T. & Goranflo, H. M. 2001 Riverware: a generalized tool for complex reservoir system modeling1. *JAWRA J. Am. Water Resour. As.* **37**, 913–929.

Zou, R., Lung, W. S. & Wu, J. 2007 An adaptive neural network embedded genetic algorithm approach for inverse water quality modeling. *Water Resour. Res.* **43**, W08427.