



Brigham Young University
BYU ScholarsArchive

International Congress on Environmental
Modelling and Software

4th International Congress on Environmental
Modelling and Software - Barcelona, Catalonia,
Spain - July 2008

Jul 1st, 12:00 AM

OptimumWater Allocation at Basin Scale Using Meta-Modeling

M. Shourian

S. Jamshid Mousavi

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

Shourian, M. and Mousavi, S. Jamshid, "OptimumWater Allocation at Basin Scale Using Meta-Modeling" (2008). *International Congress on Environmental Modelling and Software*. 15.
<https://scholarsarchive.byu.edu/iemssconference/2008/all/15>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Optimum Water Allocation at Basin Scale Using Meta-Modeling

M. Shourian^a and S. Jamshid Mousavi^b

^a *PhD. Candidate, Department of Civil and Environmental Engineering, Amirkabir
University of Technology (Tehran Polytechnic), Tehran, Iran, (shourian@aut.ac.ir)*

^b *Associate Professor, Department of Civil and Environmental Engineering, Amirkabir
University of Technology (Tehran Polytechnic), Tehran, Iran, (jmosavi@aut.ac.ir)*

Abstract: Evolutionary optimization algorithms, in many cases, suffer from a high computational cost due to the high-fidelity simulation models performed for objective function evaluations. Meta-modeling is one of the useful approaches to overcome this problem in which a surrogate model, running of which is much faster than the exact model, is used in lieu of the simulation model. To build a meta-model it is required to use a function-approximation procedure by which the expensive simulation model is approximated. Artificial Neural Networks (ANNs), as meta-models, have shown different applicability in various engineering design problems. However, training ANNs needs enough input-output data (design of experiments) each of which is obtained by running the expensive simulation model. A methodology is presented in this study in which the problems of design of experiments, function approximation and function optimization are sequentially solved in a feedback loop so that a much fewer number of experiments is required for the task of function approximation. The proposed approach adaptively utilizes the information obtained from function optimization, finds the regions where more data are needed, updates the training data set to fill the space and sequentially improves the accuracy of the meta-model. The performance of the proposed approach is analysed using a optimization problem on a benchmark multi-modal mathematical function and a real-world water allocation optimization problem at basin scale.

Keywords: River Basin; Decision Support Systems; Optimization, Meta-Modelling; Neural Networks, Design of Experiments.

1. INTRODUCTION

Water resources systems optimization problems at basin scale, containing environmental systems optimization, due to high number of the systems components and variables and discontinuous, nonsmooth, nonlinear or nonconvex functions are complex and hard to solve. There are various challenging faces to deal with in these optimization problems arising. Various highly-related components in water resources systems make the simulation-based or evolutionary optimization approach a promising and useful method in solving these problems. But, high computational costs associated with the use of high-fidelity river basin simulation models pose a serious impediment to the successful use of evolutionary algorithms (EAs) to optimization of water resources systems design and operation. In such complex problems, EAs typically require thousands of function evaluations to locate a near optimal solution. Hence, when computationally expensive simulation models, common in water resources systems analysis, are used for simulating system's performance, the use of EAs may be computationally prohibitive. One approach to reduce the computational expense is to approximate the simulation model by a less

expensive meta-model among which artificial neural networks (ANNs) have been gaining significant attention.

In field of using meta-models, Jin et al. [2002] presented a framework for coupling EAs and neural network-based surrogate models. This approach uses both the expensive and approximate models throughout the search, with an empirical criterion to decide the frequency at which each model should be used. Yan and Minsker [2006] proposed a dynamic modeling approach, called adaptive neural network GA in which ANNs networks are adaptively and automatically trained directly within a GA to replace a time-consuming groundwater remediation simulation model. A more detailed survey of the state-of-the-art in meta-modeling can be found in Simpson et al. [2002].

In this paper, an adaptive sequentially space filling (ASSF) method is presented for training an ANN, with minimum possible design of experiments, which is used as the surrogate model in a function optimization procedure. Performance of the proposed methodology is demonstrated at first in a benchmark mathematical problem with many local minima and then the developed model is applied in solving the problem of optimum water allocation at upstream Sirvan river basin system in Iran as a real case study. In the considered case study, the trained ANN is used for replacing the MODSIM [Labadie, 1995] model, a generalized river basin DSS (decision support system), in order to achieve the best possible accuracy with the least experiments. The size of planned dams and water transfer systems in a river basin, as design variables, and the values of some operational variables regarding relative water allocation priorities are searched for by the particle swarm optimization (PSO) [Kennedy and Eberhart, 1995] algorithm as the optimization engine in which MODSIM or its approximation is used to simulate the river basin system operation. Results of three modeling tools including simulation-optimization without meta-modeling [Shourian et al., 2008], adaptive sequentially space filling (ASSF) and pure random data generating meta-modeling are analysed and compared in the mentioned problems.

2. ADAPTIVE SEQUENTIALLY SPACE FILLING META-MODELING

The formulation of an optimization model whose objective function evaluation requires performance of a simulation model may be represented as below, referred to as problem (1):

$$\text{Min } f(y)$$

(1)

Subject to:

$$h_i(x) = 0, \quad i = 1, \dots, m$$

$$g_j(x) \geq 0, \quad j = 1, \dots, n$$

$$y = \text{sim}(x)$$

In problem (1), $f(y)$ is the objective function to be minimized and $h_i(x)$ and $g_j(x)$ are the equality and inequality constraints of the problem. In order to evaluate $f(y)$, in many cases, there is a need to use a high-fidelity, computationally expensive simulation model. Moreover, the mathematical functions and relationships in the simulation model, $\text{sim}(x)$, may not be algebraic functions suitable to optimization by gradient-based techniques. There are also many cases in which the simulation model is a black box expensive function in the optimization procedure. In that case the modeler may not have access to computer codes and routines (functions) of the model. It should be noted that function $f(y)$ may be multimodal with respect to decision vector x and also some of functions $h_i(x)$ or $g_j(x)$ may be nonsmooth or the feasible space of the problem may be nonconvex. Meta-heuristic and evolutionary optimization techniques are promising in solving these types of optimization problems as they can be easily linked with any simulation model without the need to have access to computer codes or details of the function $\text{sim}(x)$. However, meta-heuristic algorithms may require thousands of objective function ($f(y)$) evaluations to converge to a good solution. Each evaluation needs running the high-fidelity simulation model $\text{sim}(x)$. This may make solving problem (1) using a meta-heuristic optimization

algorithm computationally prohibitive. To address this difficulty, the incorporation of meta-models has been suggested. The purpose of meta-models is to approximate the relationship between $f(y)$ and the vector of input decision variables x by using efficient mathematical models. Therefore, a function approximation technique may be used to solve problem (1) where the approximation function $\tilde{f}(y)$ is optimized instead of exact function $f(y)$. The optimization problem in this case referred to as problem (2) may be expressed as following:

$$\text{Min } \tilde{f}(y)$$

(2)

Subject to:

$$h_i(x) = 0, \quad i = 1, \dots, m$$

$$g_j(x) \geq 0, \quad j = 1, \dots, n$$

In problem (2), $\tilde{f}(y)$ is used in lieu of the exact function $f(y)$. Hence, to solve main optimization problem (1), we need to determine the surrogate function $\tilde{f}(y)$. To have an accurate approximation the following condition should be satisfied:

$$\left| \tilde{f} - f \right| \leq \varepsilon, \quad \forall x \in X$$

(3)

where ε is the accuracy parameter of the approximation and X is the search space of problem (1). In order to obtain the required precision for function approximation used in optimization problem (2), the only possible approach is to design experiments such as to fill the search space uniformly. These experiments are used to construct the meta-model utilized as the function approximator (e.g. in case of using ANN, it is the data set used for training the network). Therefore, one could say that there is a need to perform a number of experiments, D , which means that the function $f(y)$ should be evaluated D times. In other words, in order to get the approximate function $\tilde{f}(y)$ in problem (2), a set of experiments known as $S_D = \{(x_k, f(\text{sim}(x_k)))\}$, $k=1, \dots, D$, should be prepared, which requires evaluating function f for D times. This problem is referred to as design of experiments, or problem (3).

In a classical method of design of experiments, the set S_D is made by a random data generation process. This means that in order to cover the search space of the problem, X , with a required homogeneity, a uniform random data generation process is used to locate points of x_k and then for each point of x_k the function $f(\text{sim}(x))$ is evaluated. Two main characteristics are important to determine the size of S_D : nonlinearity of function f and the dimension of the input vector x . The higher nonlinearity of function f and also the larger number of decision variables, the more experiments are needed to be designed for construction of the meta-model.

Finding a way to solve problem (2) with the minimum number of experiments is one of the main objectives of the problem solving approaches, which this paper intends to address. A methodology is used to design experiments for data training in which problems (1), (2) and (3) are not viewed as separate problems; but each of them sequentially takes advantages of the results of the previous ones.

The procedure starts at first with generating a small number of input-output data in the search space, just enough to construct an approximate surrogate model. The experiments are designed randomly and the function $f(\text{sim}(x))$ is evaluated for each experiment by running the original simulation model (design of experiments and function evaluation, problem (3)). A meta-model, which is a multi-layer perceptron ANN in this paper, is trained using the data which can be used for approximating function f (function approximation that is referred to as problem (2a)). The minimum value of approximate function \tilde{f} is obtained using the PSO algorithm (approximate function optimization, problem (2)). This value will be near to the minimum value of the exact function f , if the approximation procedure is effective. To verify this, one should evaluate the error of the

approximation (ε) between \tilde{f} and f for the solution of problem (2). If ε is small enough, there is a high chance that the located minimum of \tilde{f} in problem (2) is a near-optimal solution for the main problem (1); Otherwise, the trained meta-model has not been able to learn and accurately approximate the exact function f . If so, the solutions obtained for problem (2) are not accepted as the solutions of problem (1) and thus more experiments should be designed. These new experiments are added to the former set of experiments, S_D , and the procedure is iterated until the whole search space of the problem and especially the regions with good solutions of the main problem (1) are covered with the required accuracy. The relationships between the different parts of the problem solving procedure are shown in Figure 1.

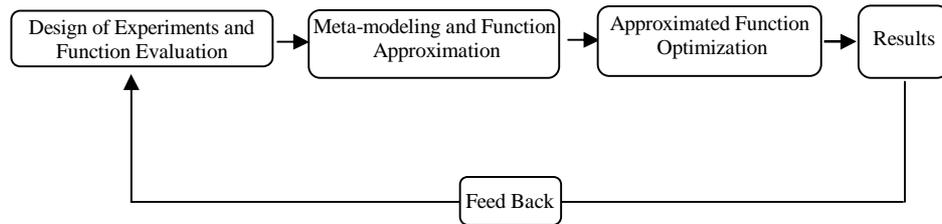


Figure 1. Relation between different parts of the proposed methodology for meta-modeling

As seen in Figure (1), the three sub-problems of (3), (2a) and (2) are solved in a feed-back loop such that solving each sub-problem is based on the information obtained from the other parts. The final procedure converges to find the solution of problem (1) using a meta-model with the minimum number of experiments.

Solving problem (2) finds an optimal solution of variables, x^j , for the surrogate function \tilde{f} . In order to accept the solution of the problem (2) as the answer of the problem (1) two important constraints must be satisfied. The first is that it is checked if there is a minimum pre-defined number of experiments in the training data set of S_D in neighborhood of x^j , as the constraint showing that the meta-model has found the optimal solution based on what it has learned from the search space of the problem. The second rule is that the exact objective value for x^j is calculated using the original simulation model. The error of approximation must be less than the required accuracy criterion. If these constraints are satisfied, x^j is accepted as a good optimizer for both \tilde{f} and f and the meta-model has been able to approximate well the promising regions of the search space. Otherwise, x^j is within a gap of the search space which is not well estimated and it is required to design a number of experiments in a covering range to be included in the training set for the next construction of the meta-model. In this case, x^j is just stored and $\tilde{f}(x^j)$ in problem (2) is temporarily penalized to let the search engine (PSO) escape from x^j and search for other solutions. The next section describes the PSO algorithm used as the search engine to solve problem (2) within the proposed methodology just explained above.

3. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

The PSO, originally introduced in terms of social and cognitive behavior by Kennedy and Eberhart [1995], has proven to be a powerful competitor to other evolutionary algorithms. The PSO algorithm simulates social behavior among individuals (particles) “flying” through a multi-dimensional search space, each particle representing a single intersection of all search dimensions. The particles evaluate their positions relative to a goal (fitness) at every iteration. Particles in a local neighborhood share memories of their “best” positions and then use those memories to adjust their own velocities and positions as shown in

equations (4) and (5) below. The PSO algorithm defines each particle as a potential solution to a problem in a D -dimensional space, with the i th particle represented as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. Each particle also remembers its previous best position, designated as p_{best} , $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, and its velocity, $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. In each generation, the velocity of each particle is updated, being pulled in the direction of its own previous best position (p_i) and the best of all positions (p_g) reached by all particles until the preceding generation.

The original PSO formulae were modified by Shi and Eberhart [1998] with the introduction of an inertia parameter, ω , that was shown empirically to improve the overall performance of PSO as follows:

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

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

where $d = 1, 2, \dots, D$; $i = 1, 2, \dots, N$, and N is the size of the swarm; χ is a constriction factor which is used in constrained optimization problems in order to control the magnitude of the velocity (in unconstrained optimization problems it is usually set equal to 1.0); ω is called *inertia weight*; c_1, c_2 are two positive constants, called *cognitive* and *social* parameters respectively; r_1, r_2 are random numbers uniformly distributed in $[0,1]$; and $n = 1, 2, \dots$, determines the iteration number.

4. FUNCTION OPTIMIZATION USING ADAPTIVE SEQUENTIALLY SPACE FILLING META-MODELING METHOD

4.1 Levy # 5 Function Optimization Problem

At first, to demonstrate the robustness of the proposed methodology, a well-known mathematical function is considered to locate its global minimum. The function is a notorious two dimensional test function, called the *Levy No. 5* [Levy et al., 1981] as follows:

$$f(x) = \sum_{i=1}^5 [i \cos((i-1)x_1 + i)] \sum_{j=1}^5 [j \cos((j+1)x_2 + j)] + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2 \quad (6)$$

where $-10 \leq x_i \leq 10$, and $i = 1, 2$. There are about 760 local minima and one global minimum with function value $f(x^*) = -176.1375$, at the point $x^* = (-1.3068, -1.4248)$. The large number of local minimizers makes it difficult for any method to locate the global minimizer. In Figure 2, the original plot of the *Levy No. 5* function, within the cube $[-10, 10]^2$, is shown.

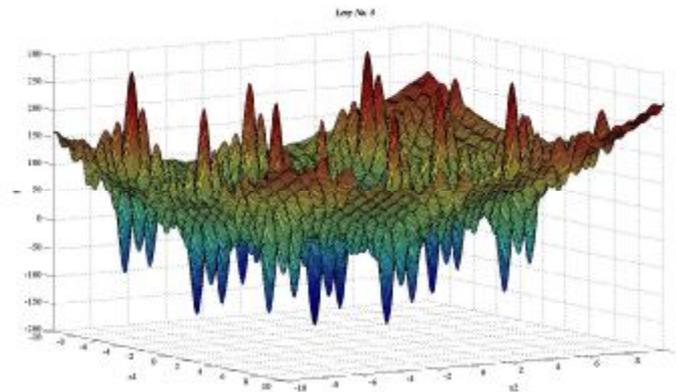


Figure 2. Plot of the *Levy No. 5* function

Results of using the ASSF meta-modeling for this bench-mark problem when compared to using meta-modeling with random design of experiments (DoE) process is represented in Table 1. Also, the solution obtained by the PSO algorithm using the exact function is presented. In this table, PSO-LEVY~ANN stands for optimization of the approximated function of Levy with the ANN.

Table 1. results of the PSO-LEVY and PSO-LEVY~ANN models

Item	PSO-LEVY	PSO-LEVY~ANN	
		ASSF DoE	Random DoE
Minimum exact objective function f^*	-176.1375	-175.9248	-2.5892
No. of training data	-	2'170	20'000
x_1^*	-1.3068	-1.3228	-1.8829
x_2^*	-1.4248	-1.4233	-1.316

As seen in Table 1, the ASSF meta-modeling method is able to locate a near optimal solution just using a number of 2'170 experiments for training the ANN while the network trained by 20'000 random data is not able to obtain a good solution. Also, it is observed that the PSO algorithm is able to locate the global solution of the problem.

4.2 Basin Scale Water Allocation Optimization Problem

Optimum water allocation planning in the upstream Sirvan river basin system in Iran has been considered as the case study problem in this paper. Excess surface water resources in the basin and growing water uses in neighboring basins have made important the subject of water transfers and water allocation. In the proposed development plan for the basin, construction of dam storage and water transfer projects has been considered. Deciding on the size of the project and how water resources should be allocated over time and space considering coordinated operation of the system components are some of the challenging issues in management of the basin. They could be dealt with by formulating a large scale optimization model for the Sirvan river basin system.

There are 9 reservoirs in the basin. Among the reservoirs, Zhaveh, Palangan and Gerdalan (S9-1) reservoirs are not constructed yet and they have to be sized. Also, relative priorities of target reservoir storages are operational variables through which the network flow program in MODSIM decides whether water stored in reservoirs in a time period should be released to meet water demands of that time period or it should be kept in the reservoirs for future uses. These priorities indicate the significance of storing in or releasing water from reservoirs in every time step in comparison with the priorities of the water demands. Therefore, there are two main sets of decision variables: (i) design variables, which include the sizes of planed dams and water transfer systems and (ii) operational variables, which include the relative priorities of target reservoir storage volumes. This optimization model in which MODSIM is linked with the PSO algorithm, PSO-MODSIM, has the form of problem (1) in which optimization the main function of f (the net benefit of the system) requires the performance of the expensive simulation model of MODSIM. Details of integrating MODSIM and PSO and structure of the PSO-MODSIM model and its application to optimal water allocation at basin scale could be found in Shourian et al. [2008].

High computational cost of the PSO-MODSIM model is the main prohibitive factor for its application to real problems. The procedure needs to call and perform a MODSIM simulation for each searching particle, which results in the total of 3000 to 4000 times of MODSIM runs with a PSO model using 20 particles. For instance, the computational time required by an up-to-date PC for solving the optimization model of the upstream Sirvan river basin over a 46 years planning horizon exceeds 20 hours. Therefore, using a meta-model to replace the high-fidelity MODSIM DSS is followed to reduce the computational cost of PSO-MODSIM model. An ANN is used as the surrogate model for substituting MODSIM in the PSO-MODSIM model. Error of training the ANN is one of the most affecting parameters in performance of the PSO-MODSIM~ANN model (~ indicates

approximation of MODSIM by ANN). Because of using an ANN as the meta-model in the PSO algorithm, it is desired to minimize training error of the ANN. Obviously, it is more beneficial to reach the required accuracy for the surrogate model with the minimum possible number of experiments used for training.

Each pair of $(x_k, f(sim(x_k)))$ (x_k is the vector of variables) in the training set S_D requires MODSIM to be executed. The sample training data set may be randomly generated. In this way, it is not guaranteed that the search space is covered appropriately unless the space is filled with a large number of generated data sets. Therefore, adaptive sequentially space filling meta-modeling presented in the second section may be employed to obtain the required surrogate model with the minimum number of experiments.

In the considered problem, capacities of Zhavveh, Gerdalan and Palangan reservoirs as design variables and priorities of the Gavshan, Zhavveh, Palangan, Azad and Gerdalan reservoirs are considered as the operational decision variables, resulting in an 8-dimensional search space. Firstly, the PSO-MODSIM model was performed and its solution was stored for comparison purposes. Then, ANN model was constructed to replace MODSIM using 100 random exemplars of the decision variables and their associated objective values for which MODSIM has been run. Subsequently, the PSO-MODSIM~ANN model was employed in which the trained ANN is called instead of MODSIM. Although the ANN model had been trained well in terms of both training and validating errors, the PSO-MODSIM~ANN model found solutions around which there were no experiments in the training set. As a result, the solutions were taken as gap representatives at which function \tilde{f} is not adequately correct and therefore more experiments are needed to fill the gaps. By using the proposed ASSF technique and increasing the number of exemplars generated around the gaps found by the PSO to 1890, the PSO-MODSIM~ANN located a near optimal solution in comparison with the answer found by the PSO-MODSIM model. The new solution, whose exact objective value (using the original MODSIM simulation model) was very close to that reported by PSO-MODSIM~ANN, was also very close to the best objective value of the PSO-MODSIM model. As described before, once the PSO algorithm finds a solution it is checked if it is a gap representative. If so, its objective value is changed temporarily such that the PSO does not come back to this solution and then the search process is continued to locate another solution. Gradually, the number of solutions in gaps decreases and finally the PSO converges to a solution which satisfies both of the described constraints to be accepted as the optimum solution for the problem. This shows that the search space has been reasonably covered and there is no need to generate extra experiments.

In order to survey the robustness of the ASSF methodology, the PSO-MODSIM~ANN model was performed again, but using a pure random data generation process to prepare the training set. This means that using an initial set of 100 data points the ANN was trained and the PSO-MODSIM~ANN model run was performed. The solution was erroneous and thus a new set of experiments was designed randomly, instead of using the ASSF method, and added to the former set of training data. This procedure was continued until the difference between values of \tilde{f} and f for the best solution found by the PSO-MODSIM~ANN model met the required accuracy criterion. In this case, it was seen that by designing 20000 random experiments the algorithm was able to find a good solution. Table 2 reports the solutions found by the proposed models and the number of data used in training the meta-models.

Table 2: results of the PSO-MODSIM and PSO-MODSIM~ANN models

Item	PSO-MODSIM	PSO-MODSIM~ANN	
		ASSF DoE	Random DoE
Minimum exact objective function f^*	808	811	822
No. of training data	-	1890	20000
Zhavveh capacity (MCM)	247.85	297.97	304.53
Gerdalan capacity (MCM)	253.8	253.8	247.8
Palangan capacity (MCM)	80	80	80

Gavshan priority	24	24	16
Zhaveh priority	25	24	25
Palangan priority	25	25	24
Azad priority	25	25	25
Gerdalan priority	30	30	30

It is seen random design of experiments for meta-modeling fails to find solutions as good as the ASSF method. The process of random data generation was stopped after generating 20000 experiments as the reported solution is close to the best solution found by the PSO-MODSIM model. However, the ASSF method has been able to locate a good solution (less than 1% difference in terms of objective function value) after only 1890 experiments which is about 9.45% of the number of experiments generated randomly. The value of the objective function reported in Table 2 has been calculated by MODSIM simulating the solution obtained by the approximate model. The approximate objective values evaluated by ANN are equal to 813.03 and 794.1 for sequential and random DoE models, respectively.

5. CONCLUSIONS

In this paper, a methodology for minimizing the required number of experiments for constructing a meta-model, especially in high-dimensional optimization problems, is presented. Three sub-problems of function evaluation, function approximation and function optimization are linked with each other and take advantage from each part's results in order to locate the regions in the search space of the optimization problem which needs to design more experiments. The proposed methodology is applied to a benchmark mathematical optimization problem and then to a basin scale optimum water allocation problem. The results show that the ASSF (adaptive sequentially space filling) model performs significantly better than random data generation-based meta-modeling. The ASSF meta-modeling is able to find solutions which are very close or even the same as the ones of the exact simulation-optimization model with less than 10% of experiments required in random generating based meta-modeling. The generalization aspect of the proposed methodology to be applied in any problem of evolutionary optimization with approximate fitness functions makes it an effective tool for solving large scale water systems optimization problems which contain environmental problems as a within subject.

REFERENCES

- Jin, Y., Olhofer, M. and Sendhoff, B., A Framework for Evolutionary Optimization with Approximate Fitness Functions, *IEEE Transactions on Evolutionary Computation*, 2002.
- Kennedy, J. and Eberhart, R. C., Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks, IV, 1942-1948*, IEEE Service Center, Piscataway, NJ, 1995.
- Labadie, J., MODSIM: River Basin Network Flow Model for Conjunctive Stream-aquifer Management, *Program User Manual and Documentation, Colorado State University*, 1995.
- Levy, A., Montalvo, A., Gomez, S. and Galderon, A., Topics in Global Optimization, *Springer-Verlag*, New York, 1981.
- Shi, Y. and Eberhart, R.C., A modified Particle Swarm Optimizer, *Proceedings of the 1998 IEEE Conference on Evolutionary Computation*, AK, Anchorage, 1998.
- Shourian, M., Mousavi, S. J. and Tahershamsi, A., Basin-wide Water Allocation Planning by Integrating PSO Algorithm and MODSIM, *Water Resources Management, In press*, available at <http://www.springerlink.com>, 2008.
- Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch, P. N. and Yang, R.-J., Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion, *Proceedings of the Third ISSMO/AIAA Internet Conference on Approximations in Optimization*, October 14-25, 2002.

Yan, S., and Minsker B., Optimal groundwater remediation design using an Adaptive Neural Network Genetic Algorithm, *J. Water Resources Research*, 42, W05407, doi: 10.1029/2005WR004303.