A Dynamic Meta-Model Approach to Genetic Algorithm Solution of a Risk-Based Groundwater Remediation Design Model

[1]Shengquan Yan and [2]Barbara Minsker

[1] Research assistant, Department of Civil and Environmental Engineering, University of Illinois, 205 North Mathews, MC-250, 4146 Newmark Civil Engineering Laboratory, Urbana, IL 61801-2352. (smyan@uiuc.edu)
[2] Associate professor, Department of Civil and Environmental Engineering, University of Illinois, 205 North Mathews, MC-250, 3230 Newmark Civil Engineering Laboratory, Urbana, IL 61801-2352. (minsker@uiuc.edu)

Abstract

Approximation ("meta") models have been used in coupled optimization and simulation models to improve computational efficiency. In most instances, multiple simulation runs have been done before the optimization, which are used to fit an approximate model that is then used for the optimization. In this study, we propose a dynamic meta-modeling approach, in which artificial neural networks (ANN) and support vector machines (SVM) are embedded into a genetic algorithm (GA) optimization framework to replace time-consuming flow and contaminant transport models. Data produced from early generations of the GA are sampled to train the ANN and SVM and the numerical models are periodically called to dynamically update the ANN and SVM. This allows the meta model to adapt to the area in which the GA is searching and provide more accuracy. Preliminary results show that a well trained ANN or SVM can achieve satisfactory accuracy. Different approaches to dynamic training will be presented at the conference.

Introduction

Groundwater management models often involve coupling complex simulation models with optimization models to achieve management goals. For the risked-based remediation design model studied in this paper (Smalley and Minsker, 2000), a flow model, contaminant transport model, chemical reaction model, and risk evaluation model are combined with optimization to identify optimal solutions. The combination of those different complex sub-models and the different model objectives make the traditional non-linear programming approach difficult for finding a global optimum. As an alternative, heuristic optimization techniques have prevailed in recent years because of their loose requirement on gradient information and their better global searching ability. For example, simulated annealing (Dougherty and Marryott, 1991); genetic algorithm (Smalley and Minser, 2000); and tabu search (Zheng, 1996) have been used to solve groundwater remediation design problems. Among those different methods, only genetic algorithms (GA) can easily incorporate noisy and multiple objectives into the risk-based model (Reed et al., 2002; Gopalakrishnan et al., 2003).

The limitation of using GA is its expensive computational requirements. Its fitness evaluation can involve the solving of time consuming contaminant transport and chemical reaction models. The GA optimization process needs to evaluate the fitness function thousands of times before it can converge to the global optimum. This is a great limitation when applying the GA optimization framework to larger remediation cases, even with the help of parallel computation.

To alleviate the computational burden, many less accurate but much more computationally efficient approximation methods have been explored. Cooper et al., 1998 used curve-fitting methods to approximate the response surface. Aly and Peralta, 1999; Rogers et al., 1995 used artificial neural network (ANN) and GA to optimize groundwater problems. While these methods can be helpful in GA optimization, they were all using a static response surface that was trained using a well designed off-line data set before the optimization began. This requires the off-line data set to be carefully selected so that it is representative of the problem domain, which can be a time-consuming process by itself. Furthermore, the approximated static response surface, albeit accurate at the beginning, may be less and less representative when the GA population converges to other local areas later in the run.

This paper proposes a unique dynamic approach to create and adjust the response surface within a GA optimization framework. The algorithm periodically samples data points from the GA population and uses that information to synchronize the response surface with the current GA population. The approximation models replace the PDE models to accelerate the fitness evaluations. As the GA converges to local areas, the response surface should become a more and more representative approximation to the decision surface of those local areas. Different approximation methods, including artificial neural network (ANN) and Support Vector Machine (SVM) are discussed in the paper. The initial results indicate a promising future.

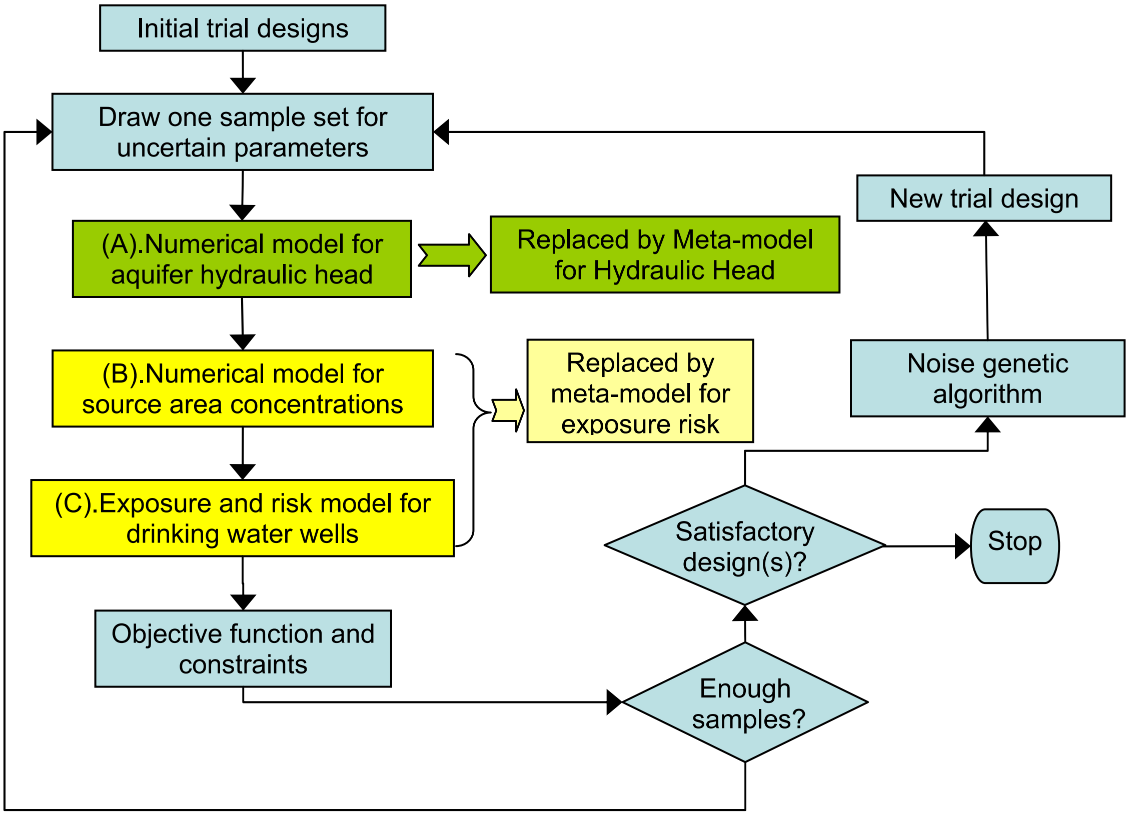
Risk-based GA Optimization Framework

The risk-based GA optimization framework developed by Smalley et al. (2000) is depicted in Figure 1, along with notations for where the meta-models would replace the simulation models in this paper. First, the initial GA trial designs are generated randomly. Second, the Monte-Carlo simulation samples uncertain parameters (hydraulic conductivities and human health risk assessment parameters). Third, the numerical groundwater flow model (component A, MODFLOW) is called to simulate the flows in the source area for the specified set of parameters. Fourth, the chemical fate and transport model (component B, RT3D) is called to estimate the contaminant plume spreading over time. The exposure and risk assessment model (component C) estimates the human health risk at drinking water wells downgradient from the contaminant source. The risk information is incorporated into the GA to produce penalties if any of the risk standards is violated. The GA then uses the selection, crossover, and mutation to create new trial designs for the next iteration.

The objective function, given by Equation 1, represents the total cost of a remediation design, where the total cost C_{TOT} consists of three components – C_{REM}, which is the capital and operating costs for the wells; C_{MON}, which is the cost of on-site monitoring; and C_{SYST}, which includes additional capital and operating costs for the ex-sit treatment system. The details are presented by Smalley et al. (2000). The goal of the optimization framework is to find the least-cost solution without violating the pumping rate and hydraulic head limits and risk criteria. The GA repeatedly detects possible locations to install pumping wells, which are discrete variables, and possible pumping rates for the remediation process until the global or near-global optimum is found.

$$MinC_{TOT} = C_{REM} + C_{MON} + C_{SYST} \qquad (1)$$

Although the GA can handle this typical mixed-integer problem, it is computationally expensive if all the numerical sub-models are fully realized throughout the GA optimization process. MODFLOW, the groundwater flow model, and RT3D, the chemical transport and fate model, can be time-consuming for complex sites. For a large test case that has a finely divided grid, each evaluation of the two models require more than 2 hours. During our research, two meta-models based ANN or SVM are embedded into the framework to replace the numerical models. The first one replaces MODFLOW (Component A). The second one replaces the combination of RT3D and risk evaluation model (component B plus component C). If the two meta-models are adequately trained and updated, the optimization process can be greatly accelerated without excessively compromising accuracy.



Initial trial designs
Draw one sample set for uncertain parameters
(A).Numerical model for aquifer hydraulic head
Replaced by Meta-model for Hydraulic Head
(B).Numerical model for source area concentrations
Replaced by meta-model for exposure risk
(C).Exposure and risk model for drinking water wells
Objective function and constraints
Enough samples?
Satisfactory design(s)?
Stop
Noise genetic algorithm
New trial design

Firgure 1. GA optimization framework

Meta-model Methodology

Artificial Neural Network (ANN). ANNs have been successfully applied to automatic driving, post code recognition and weather forecasting (Russel, 1995). Their widespread application in different disciplines can be attributed to their ability to approximate almost any linear or non-linear problems. An ANN has many neuron-like units. Each unit accepts input and gives output according its activation function. The interconnections among the neurons in an ANN have weights associated with each connection, which compose a large parameter set. By gradually tuning the weights associated with each interconnection, a supervised learning algorithm can learn the mapping relationship between the inputs and the outputs sampled from a training set. Then the trained ANN is used to make a prediction.

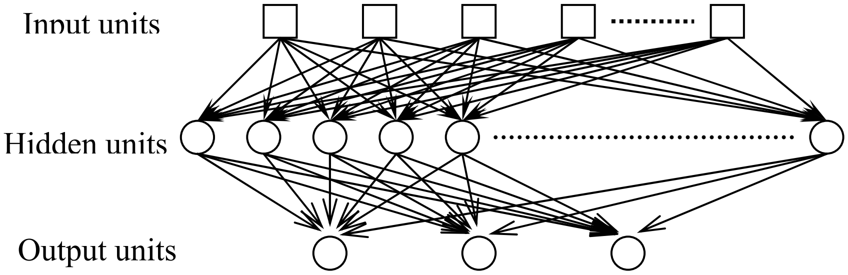


Figure 2. ANN architecture

We adopted a two layer, feed forward neural network for our meta-models. As shown in Figure 2, the input data are directly sent to the hidden layer. After the hidden layer transformation, the output layer has the final predicted results. The activation function of each neural unit is the sigmoid function, which is convenient for derivative calculation for the training algorithm.

Both the back propagation and conjugate gradient learning algorithms are tested in our study. The back propagation learning is the traditional gradient descent algorithm. The error produced by the output layer is propagated back to the input layer. The weights associated with the links are then adjusted according to the error derivatives. After a number of training phases, the error is minimized on the training data set.

The back propagation algorithm is easy to implement. However, its convergence is slow and it is easily trapped in local minima. Conjugate gradient algorithm provides an alternative approach. Rather than strictly following the steepest descent direction, as the back propagation algorithm does, conjugate training follows a shifted decreasing direction by taking information from previous steps into account (Fletcher, 1964). A conjugate search generally converges better than the back propagation algorithm due to its more effective searching approach.

Although ANN can approximate any training set, over fitting must be avoided. An over fit ANN just memorizes the mapping relationship of the inputs and outputs but loses generalization ability (i.e., the ability to predict outputs for other data sets). This can be avoided by reducing the complexity of the ANN structure (number of hidden layers and their neurons) or by stopping the training phase when the testing error begins to decrease. Generally, if the training set is much larger than the total number of adjustable weights in the ANN, over fitting is less likely, which is true in our case.

Radius Basis Function (RBF) network is also considered in this research. The RBF network uses a radial function - typically a Gaussian function - as the activation function for transforming inputs. Clustering of the data set, using algorithms such as K-means clustering, is the first step in creating an RBF network. Clustering identifies the center point and radius (i.e., mean and standard deviation) of the Gaussian function in each unit of the RBF network. Each point in the data set belongs to one and only one cluster. We expect that an RBF network may perform particularly well if the GA converges to many local areas in later generations.

Support Vector Machine (SVM). In many respects, SVM shows competing or even better performance than ANN (Schölkopf, 1999). The basic principle of SVM is demonstrated by a simple linear separable classification problem (Figure 3), where the white nodes and black nodes represent the positive (+1) and negative points (-1) in a hyperspace. The objective is to find a hyperplane $\mathbf{w} \bullet \mathbf{x} + b = 0$ separating the two classes of points. The SVM algorithm simply looks for the separating hyperplane with the largest margin (the perpendicular distance between the two dashed lines), which is the distance that the hyperplane shifts perpendicularly without losing its separation ability. Intuitively, the circled points are much more important than other points because they are confining the margin and the moving of any of them will change the hyperplane normal vector **w**. Those circled points are called "support vectors".

The SVM training algorithm uses a Lagrangian formulation to maximize the margin. While this illustrative example is for classification of data, SVM can also be used to fit a nonlinear function to data as in this research. For nonlinear function approximation, SVM finds a kernel function that maps the inseparable input data to a higher dimensional hyperspace. Through this mapping, the data become more separable, allowing separating hyperplanes to be found to classify the data into similar groups (as illustrated in Figure 4). Support vectors are then found for each class, as described previously. The support vectors are then used to approximate the function represented by the data.

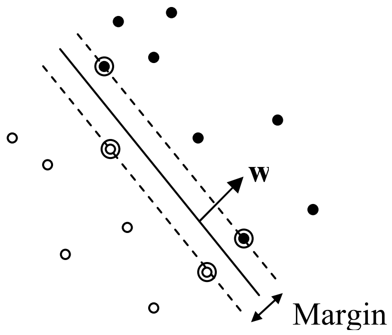


Figure 3. Maximizing the margin of separating hyperplanes; the circled points are support vectors.

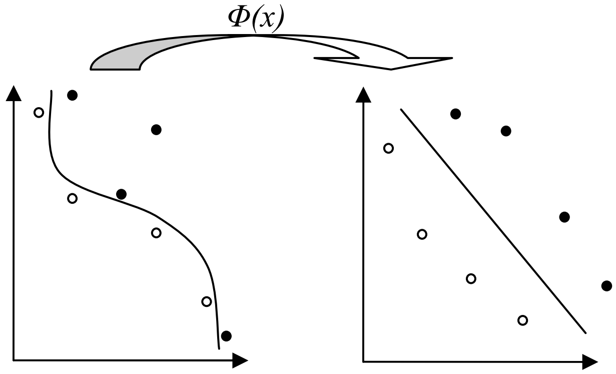


Figure 4. Mapping from a inseparable space to a separable space

In this study, we used an SVM optimization package from the University of Dortmund (Ruping, 2000). Initial runs showed that the anova kernel (defined by Burgers, 1998) within the package has a better prediction result, so this kernel function was used for this study.

Case Study and Results

Case Study. A simple case study is used in the research (Figure 5). The aquifer plane is discretized into a 16 x 8 mesh grid. Due to regional flow and dispersion, the contaminant plume spreads slowly to the down gradient area. Three pumping wells are the candidates for pumping treatment. The decision variables are their locations in the mesh grid and the pumping rates. Monitoring wells are installed at the boundaries and several locations in the source area. The contaminant concentration readings are used to predict human health risk associated with drinking the water.

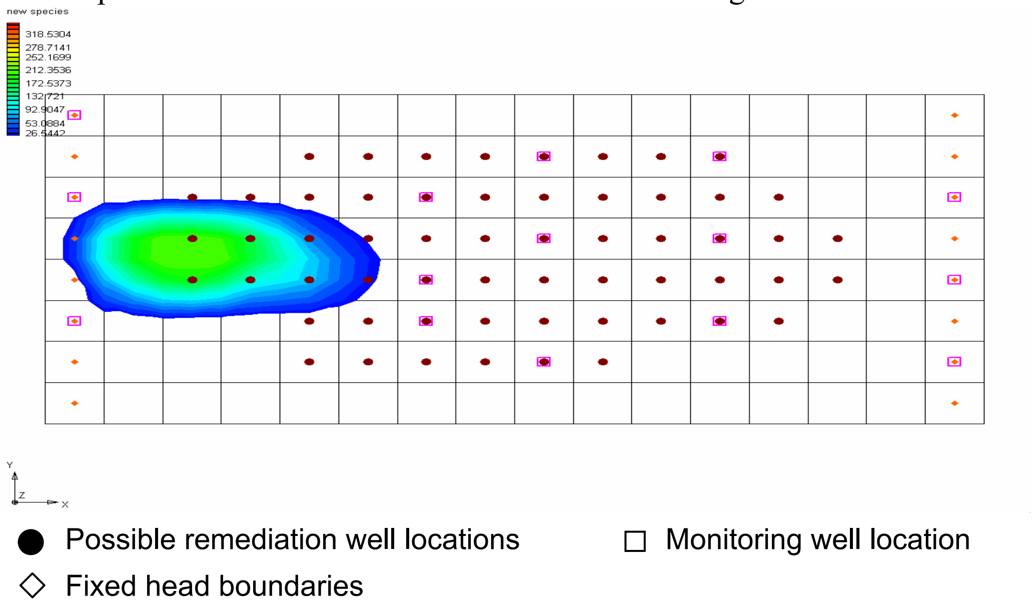


Figure 5. Aquifer case study

Model Parameters and Structures . The GA has a population of 100 individuals, which was determined using the population sizing approach developed by Reed et al. (2000) in previous work. The fitness evaluation of each individual requires about 1~2 minutes using the numerical models. The algorithm converges after about 98 generations. In our test case, in the first 10 generations, the numerical models are called to generate the training data set. The numerical models are called again in the 11th generation to generate the testing data set. The GA is then suspended to wait for the meta model training. The trained meta models are then tested on the test data set before they are fed to the GA process for future generations.

Both the ANN meta model and the SVM meta model have two sub-models. The first model uses the pumping well locations and the pumping rates as input to produce output hydraulic heads at the pumping wells. The pumping well locations, the pumping rates and the hydraulic heads at those wells are then used to feed a second sub-model, which predicts the maximum log risk at all the monitoring wells.

Log risk was used to avoid scaling problems associated with extremely small numbers.

The groundwater flow sub-model of ANN has 9 units in the input layer, 23 units in the hidden layer and 3 units in the output layer. The risk evaluation sub-model of ANN has 12 units in the input layer, 20 units in the hidden layer and 1 unit in the output layer. SVM uses anova kernel as the transformation mapping function. Unlike the ANNs, the SVM can't have multiple outputs. So there are three independent SVMs in the flow sub-model, one for each pumping well. As with the ANN risk sub-model, The SVM risk sub-model has 12 inputs and 1 output. All of the model structures were determined by an incremental search. Starting from a small number of units in the ANN or the low range of the parameters in the SVM, we gradually increased the number of units in the ANN or adjusted the parameters of SVM. We then chose the meta models whose training error was reduced as much as possible without increasing the testing error (overfitting).

Inputs and outputs of the meta-models are scaled to the range *[-0.9, 0.9]*. Mean Square Error (MSE) is used to as evaluate the predictive accuracy. All the MSE results presented here are based on the scaled data. Cross validation is monitored during the training phases to avoid over training.

Results. Our results are shown in the following figures. The horizontal axes represent the individual number in the 11th population generated by the GA (the testing set). Each individual corresponds to a candidate treatment solution. Figures 6, 7 and 8 show the original model head prediction and the ANN head prediction for each pumping wells. For the three pumping wells, the ANN training algorithm attains an MSE equal to 0.0015, and the ANN prediction on the testing set gives an MSE equal to 0.0025. This is equivalent to an average head error of only 1%. The SVM training algorithm stops when the training MSE is less than 0.0001. The SVM prediction gives an MSE on the testing set of 0.0008. The average head error is only 0.36%. Figures 9, 10 and 11 show the SVM head prediction for the three pumping wells.

The SVM prediction is better than the ANN prediction. However, both of them have low error on the testing set and thus are qualified for substitution in the groundwater flow models in the following generations. SVM needs less time for the training phase than ANN (1 minute for SVM vs. about 8 minutes for the gradient descent training on ANN and about 5 minutes for the conjugate training on ANN, for runs done on a Pentium III 1.7G PC), so SVM would be preferred for dynamic training during a GA run.

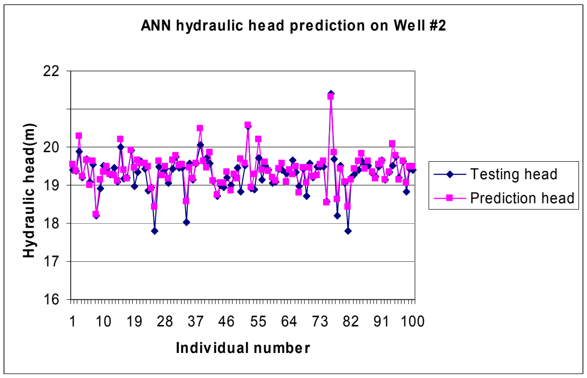


Figure 6.

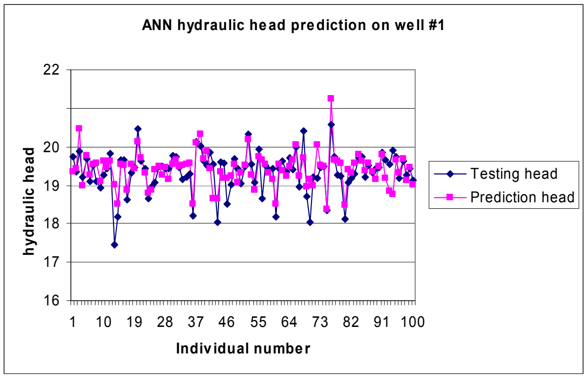


Figure 7.

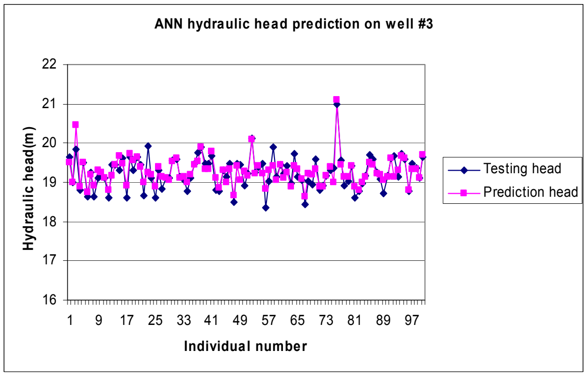


Figure 8.

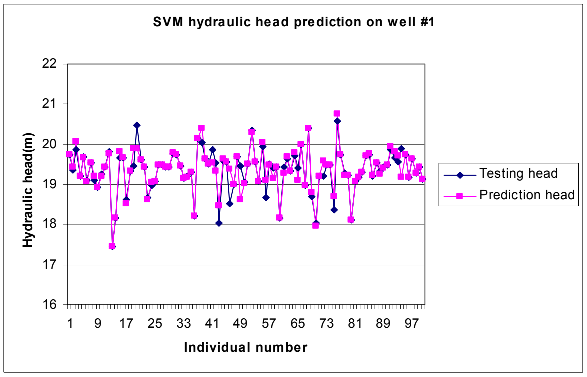


Figure 9.

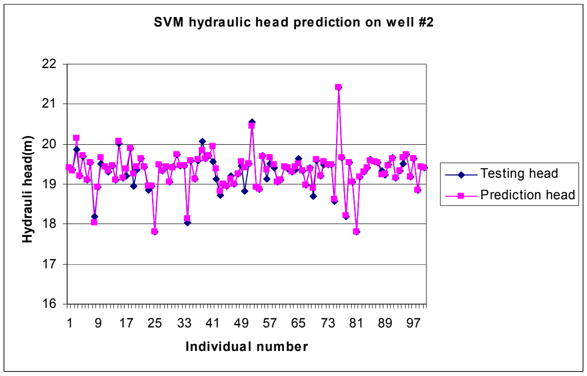


Figure 10.

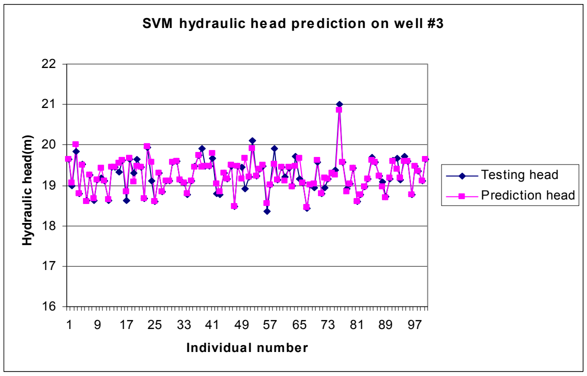


Figure 11.

For the risk meta models, the ANN training algorithm stopped at MSE equal to 0.0006, and the prediction of the ANN on the testing set gives an MSE equal to 0.0012. The average log risk error is 3.7%. Figure 12 shows the fit of the ANN risk prediction. The SVM training algorithm stops when the training MSE is less than 0.0001. The SVM prediction gives an MSE equal to 0.0042 with an average log risk error of 5.7%. Figure 13 shows the fit of the SVM risk prediction.

The ANN risk prediction is better than the SVM for these experiments. Further research need to be done on other test cases to make a more reliable conclusion. Both approaches are sufficiently accurate to qualify for the application in the GA framework. As with the groundwater flow models, the SVM training is much quicker than the ANN training.

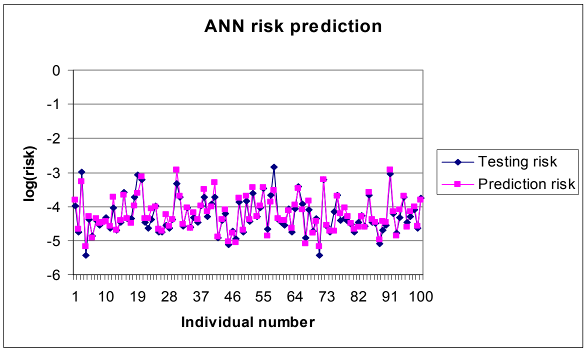


Figure 12.

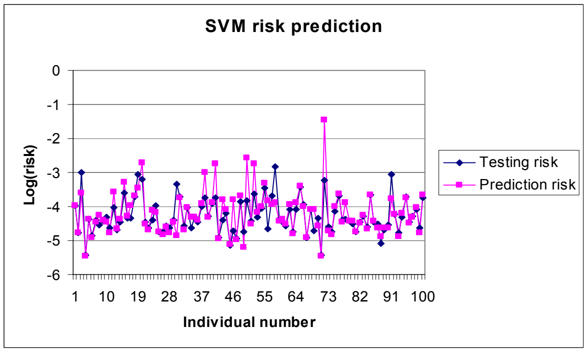


Figure 13.

To evaluate the need for dynamic training of the meta-model as the GA progresses in later generations, the above meta-models were next tested in making predictions for the 91st generation. Because the meta-models are all based on a training set from the first 10 generations, their response surfaces should be less accurate by the 91st generation, when the GA may be converging to some local areas. Figures 14 and 15 show the hydraulic head predictions at pumping well #1. The prediction MSE increased to 0.0051 (for ANN) and 0.0057 (for SVM). The average head errors are 1.9% and 2.2%. The other two wells have a similar result. If a new sampling set from 80-90 generations is used to retrain the meta-models, the MSE drops back to 0.001 for both ANN and SVM. This phenomenon clearly demonstrates the need for retraining the meta-models periodically to keep the model accuracy from going down. The same procedures repeated on the risk prediction gave similar results (Figure 16, 17). The MSE increased to 0.007 (with average log risk of 9.4%) for ANN and 0.036 (with average log risk of 24%) for SVM. The meta models have reduced generalization ability by the 91st generation and retraining the models can improve performance.

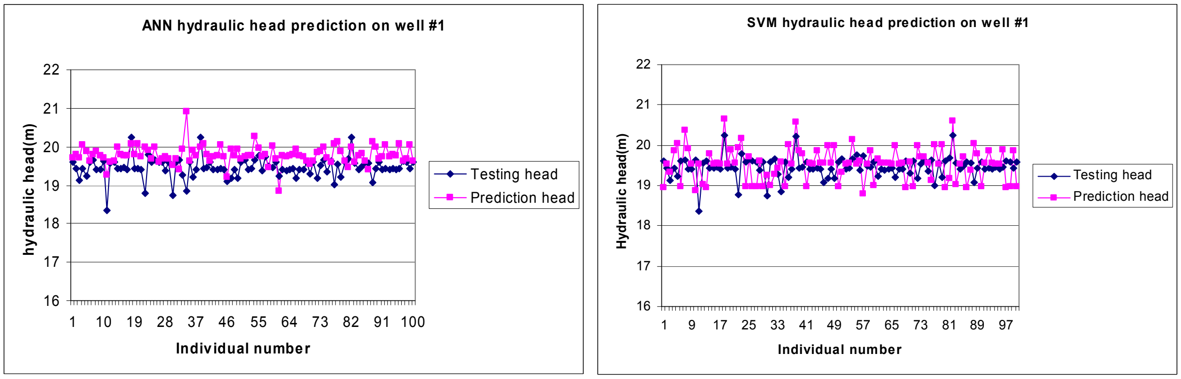


Figure 14, 15. Models trained on the 1~10 generations predict well #1 head on the 91st generation

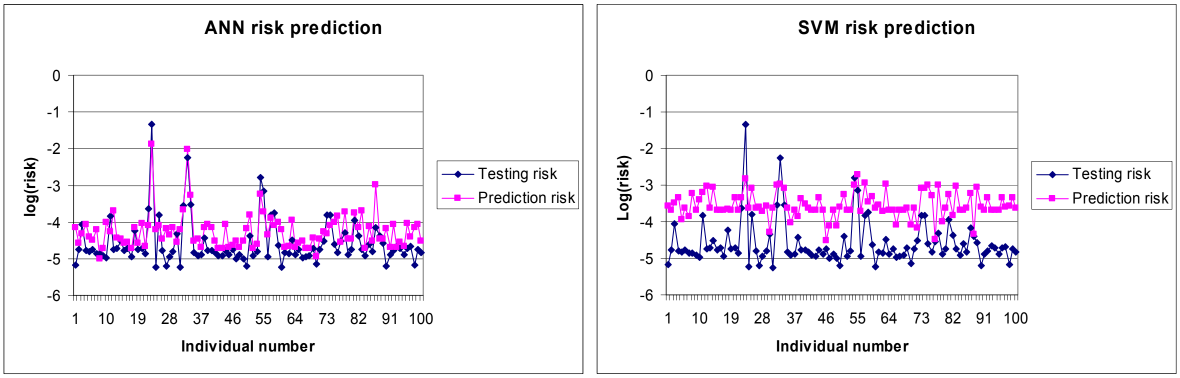


Figure 16, 17. Meta models trained on the 1~10 generations predict risk on the 91st generation

Conclusion

Our current research shows that the online sampling data from the early generations of a GA run, when fed to train SVM or ANN meta models, are able to capture the features of the response surfaces produced by the numerical models. The SVM hydraulic head prediction is slightly better than ANN, while ANN wins on the risk prediction. However, the meta models must be retrained or updated to fit the GA population in later generations in order to avoid loss of accuracy. Both meta models require trivial time when used for prediction but SVM training is substantially faster than ANN; hence, the SVM is better suited for online adaptive training. When the meta models trained by these data are used to replace the numerical models, the models can significantly accelerate the optimization computation time without losing too much accuracy. Work is ongoing to investigate approaches to online updating and to determine how frequently updating should be performed.

Acknowledgement

This material is based upon work supported by, or in part by, the U.S Army Research Office under grand number DAAD19-001-1-0025

References

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

Burges, Christopher J.C., *A tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery 2, 121-167, 1998.

Cooper, G., R. C. Peralta, and J. J. Kaluarachchi, *Optimizing separate phase light hydrocarbon recovery from contaminated unconfined aquifers*, Adv. Water Resour., 21(5), 339-350, 1998

Cotes, C. and V. Vapnik, *Support vector networks*, Machine Learning, 20, 273-279, 1995

Dougherty, D.E., and R.A. Marryott, *Optimal groundwater management 1. Simulated Annealing*, Water Resour. Res., 27(10), 2493-2508, 1991.

Fletcher, R. and C. M. Reeves, Function minimisation by conjugate gradients, *Comp. J.* **7**-149, 1964.

Gopalakrishnan, G., B. S. Minsker, and D. Goldberg, *Optimal sampling in a noisy genetic algorithm for risk-based remediation design*, J. of Hydroinformatics, 2003.

Reed., P. M., B. S. Minsker, and D. E. Goldberg., *Designing a Competent Simple Genetic Algorithm for Search and Optimization*, Water Resour. Res., 36(12), 3757-3761, 2000.

Reed, P., B. S. Minsker, and D. E. Goldberg, "*Simplifying Multiobjective Optimization: An Automated Design Methodology for the Nondominated Sorted Genetic Algorithm-II*." Water Resour. Res., submitted, 2002.

Rogers, Leah L., Farid U. Dowla, and Virginia M. Johnson, *Optimal Field-Scale Groundwater Remediation Using Nerual Networks and the Genetic Algorithm*, Environ. Sci. Technol. 29, 1145-1155, 1995

Ruping, Stefan, *mySVM – Manual*, 2000

Russel, Stuart, Peter Norvig, *Airtifical Intelligence-A Modern Approach*, Prentice Hall, 1995.

Schölkopf, B., *Support vector learning*, Tutorial given at DAGM'99, 1999

Smalley, J. Bryan, Barbara S. Minsker, and David E. Goldberg, *Risk-based in situ bioremediation design using genetic algorithm*, Water Resour. Res., 36(10), 3043-3051, 2000

Zheng, C. and P.P. Wang, *Parameter structure identification using tabu search and simulated annealing*, Advances in Water Resources, vol. 19, no. 4, 215-224., 1996