Optimal Groundwater Remediation Design Using Trust Region Based Meta-models within a Genetic Algorithm

[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

Computational cost is a critical issue for large-scale water resource optimization problems that often involve time-consuming simulation models. Less accurate approximation ("meta") models can be used to improve computational efficiency. We propose a novel trust-region-based metamodel framework, in which hierarchically trained metamodels are embedded into a genetic algorithm (GA) optimization framework to replace time-consuming numerical models. Numerical solutions produced from early generations of the GA, along with solutions dynamically sampled from later generations, are used to retrain the metamodels and correct the GA's converging route. A bootstrap sampling technique is used to cluster the collected numerical solutions into hierarchical training regions and then multiple metamodels are trained based on these clustered regions. The hierarchically trained metamodels are then used to approximate the numerical models. A trust region testing strategy selects the most appropriate metamodels for prediction. This allows the local regions (particularly those near the optimal solution) to be approximated by smoother and smaller metamodels with higher accuracy. This can speed up GA's convergence when the population moves into local regions. The technique was tested with artificial neural networks (ANNs) and support vector machines (SVMs) on a field-scale groundwater remediation case in a distributed network computation environment. Our preliminary results show that the adaptive meta-model GA (AMGA) with the trust region based training technique converges with higher accuracy with the same computation effort.

Introduction

Groundwater management models often involve coupling complex chemical simulation models with optimization techniques to achieve management goals. These models usually involve solution of computationally intensive partial differential equations (PDEs). These complex nonlinear models can be difficult to solve with

traditional optimization approaches, which may not find a global optimum. Genetic algorithms are well suited for solving such problems, but can require substantial computational resources when each objective function (fitness) evaluation involves solving time-consuming PDE models. This is because the GA optimization process needs to evaluate the fitness function thousands of times before it can converge to the global optimum. This is a significant limitation when applying the GAs to more complex 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) and Rogers et al. (1995) used artificial neural network (ANN) and GA to optimize groundwater problems. The approximated static response surface, albeit accurate at the beginning, may be less and less representative when the GA population converges to local regions later in the run. As a result, the GA may prematurely converge to regions containing local minima. One approach that addresses this difficulty involves adaptive response surface methods. Brooker (1998) proposed a framework for generating and managing a sequence of surrogates to the objective function. Jin et al. (2002) used Evolutionary Algorithms (EAs) with ANNs that adapt to sampling model errors to accelerate aerodynamic design problems. In the water resource field, Yan and Minsker (2004) proposed a dynamic neural network framework for optimizing groundwater management problems. This approach used sampling techniques to update the ANNs to achieve good prediction accuracies.

This paper extends the work of Yan and Minsker by proposing a novel trust-region based meta-model framework. Instead of creating a global approximation model, the method hierarchically trains a global model and a series of local models to replace the numerical models. The meta-models trained on local regions are trusted only in these regions and trust region testing is used to select appropriate meta-models for prediction. A bootstrap sampling algorithm is used to cluster the sampled PDE solutions into hierarchical regions for training these models. An overview of the algorithm is presented in this paper. The approach is tested on a field-scale groundwater remediation case using both artificial neural networks (ANNs) and support vector machines (SVMs) as the approximation models.

Trust Region Based Adaptive Meta-model GA Framework

Figure 1 shows the flowchart for the trust region based adaptive meta-model GA (TRAMGA), which starts like all simple genetic algorithms. First an initial population of trial designs is generated randomly. Second, the designs are evaluated to determine their fitness with respect to the objectives and constraints of the optimization problem. In this study, the fitness evaluations were distributed to multiple host computers on a standard office network and cluster systems, allowing them to run them simultaneously (TRAMGA does not require such a setup, however). After the fitness of each design is evaluated, TRAMGA uses GA operators to create new trial designs for the next iteration ("generation"). This process continues until the population converges to the optimal solution. In the first few generations, TRAMGA is in the phase of preparing training sets for the meta-models and the designs are evaluated by simulation models. Once a sufficient number of simulation model

evaluations have been completed to train the meta-models, most of the fitness evaluations are bypassed using meta-model evaluations based on a trust region test, coupled with a memory cache of previously evaluated simulations. At each generation, only a few designs are evaluated using the simulation model ("sampled") according to a sampling policy. These designs are evaluated by the simulation models, whose results are used to update the cache and build the retraining sets. From time to time, meta-models may be retrained and if sufficient new data are available, new hierarchical meta-models are created by sampling the training points into trust regions, which in turn improves their approximation accuracy. These components are described in more detail in the following sections.

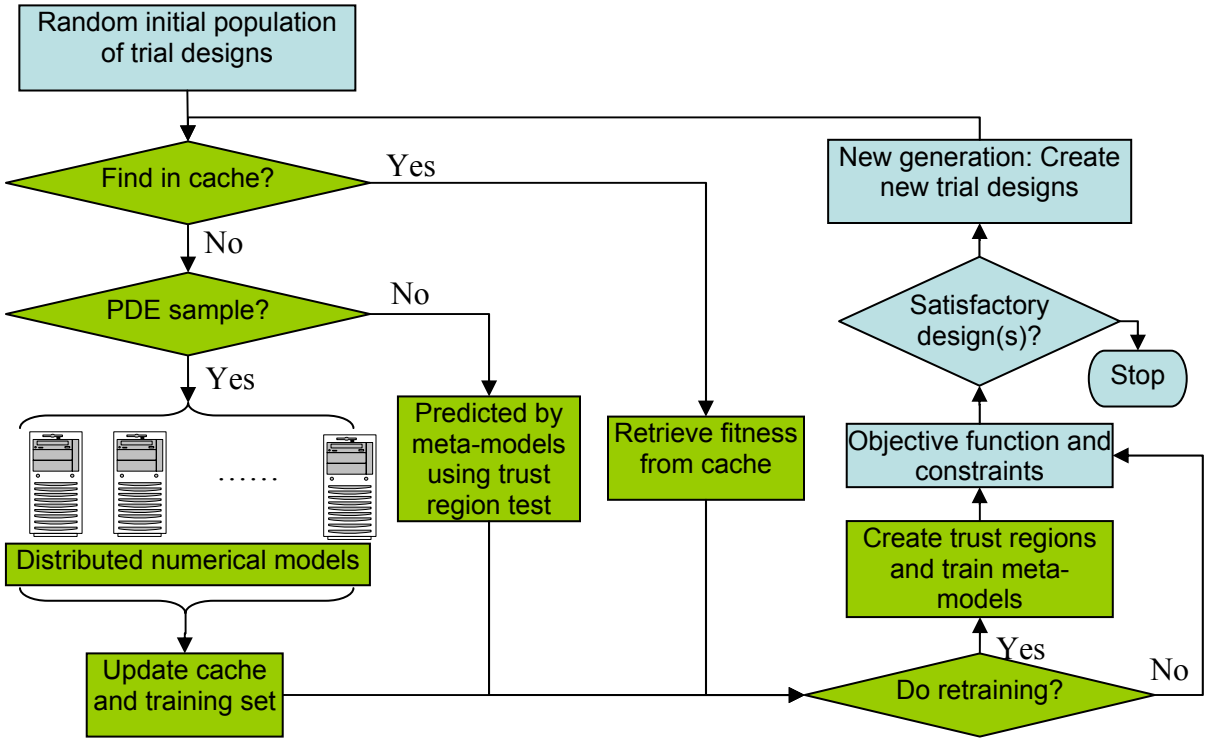

Figure 1. TRAMGA optimization framework

Trust Region Based Meta-Model Methodology

Artificial Neural Network (ANN). ANNs have been successfully applied to numerous applications, including Rogers, 1995, and Aly, 1999 in the water resources field. Their widespread application in different disciplines can be attributed to their ability to approximate almost any linear or non-linear problem. An ANN has many neuron-like units. Each unit accepts input and gives output according to 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.

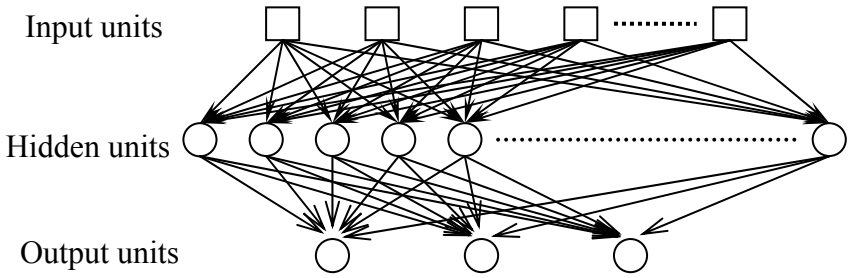

Figure 2. ANN architecture

We adopt 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 gives the final predicted results. The activation function of each neural unit in the hidden layer is the sigmoid function, which is convenient for derivative calculation for the training algorithm. The training algorithm is Levenberg-Marquardt backpropagation method, which is much faster than steep descent backpropagation and has affordable memory requirements. Each training episode has a training set and a smaller validation set. The training iterations terminate when the Mean Square Error (MSE) on the validation set stops decreasing, which avoids overfitting.

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 in a perpendicular direction without losing its separation ability. Intuitively, the circled points are much more important than other points because they are confining the margin and moving 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 used to approximate the function represented by the data.

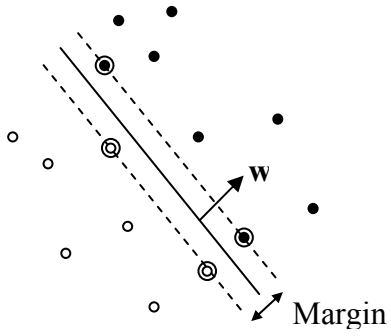

Figure 3. Maximizing the margin of separating hyperplanes; the circled points are support vectors.

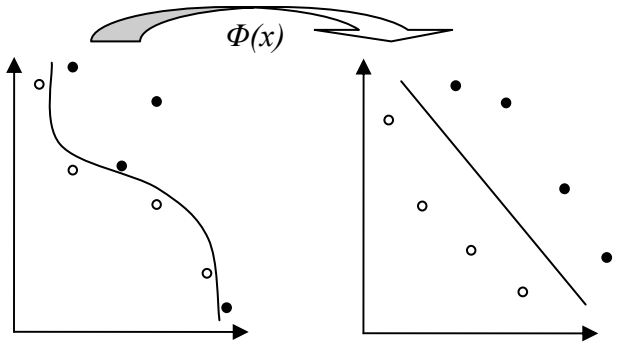

Figure 4. Mapping from a inseparable space to a separable space

In this study, we used LIBSVM, an SVM optimization package obtained from researchers at the National Taiwan University (Chang and Lin, 2001). Initial runs showed that the radius basis function kernel within the package has a good prediction result, so this kernel function was used for this study.

Trust Region Algorithm. In a dynamic environment, a framework that can dynamically create approximation models or adjust the existing ones is required. Here we propose a trust region based framework for this purpose. We first give an overview of the classical trust region algorithm. Then we show how the trust region based framework can be adapted to GAs. Finally we present a bootstrap multiple sampling algorithm for creating the trust regions and the approximation models.

Classical Trust Region Optimization Algorithm. The classical trust region algorithm is primarily used for non-linear optimization to improve global convergence. In most non-linear optimization methods, a local approximation model is made in the vicinity of the current solution. The approximation model is then used to identify the best search direction and the size of the search step to be taken in that direction. The trust region algorithm adaptively adjusts the searching step lengths and search directions to take best advantage of the approximation model's prediction but without moving beyond the region in which the local model can be trusted. The idea is to define a trust region so that the adjustment of the searching steps is confined within the trust region.

In practice, the trust region size should be updated from step to step: if the approximation model worked very well on predicting the improvement of the objective function, then the size should be enlarged. Otherwise if the approximation model worked poorly, the size should be decreased to improve the fidelity of the quadric approximation.

The classical trust region algorithm was designed to solve optimization problems that can be approximated by quadric functions. Alexandrov et al (1998) proposed a trust region framework for managing more general approximation models. In the framework, a series of approximation models with different fidelities can co-exist. In addition to adjust the trust region size, the framework may switch the approximation model so that a model with lower computational cost but acceptable fidelity is chosen for prediction.

Trust Region Based Adaptive Meta-Model GA. The idea of doing prediction with models of different fidelities can be applied to GAs as well. In GAs, there are no such predetermined searching steps to adjust. However, if we assume the approximation models have similar costs, we can switch to the best available high fidelity model to get better accuracy using the trust region testing approach.

When a meta-model is trained on a given training set, we call the region covered by the training set the trust region of the trained meta-model. The size of the region can be represented by the radius of the smallest ball containing the training points. If there are multiple approximation models trained around a point but with different trust regions, the fidelities of the approximation models can be estimated by their trust region sizes (assuming the models are trained with similar number of training points). This is because smaller trust regions have smoother response surfaces and consequently lower prediction errors from the locally trained approximation models. When a set of models are used for prediction, the models whose trust regions cover the prediction point are trusted. However, the models with higher fidelity are more likely to provide higher accuracy. Therefore, the final prediction result can be the weighted prediction average of the trusted models, or the most trusted local model can take the full credit. This approach can presumably produce better accuracy in local regions because locally trained models are usually superior to a single approximation model that covers the whole global area.

In a trust region framework, the locally and globally trained ANNs can be combined to make predictions. In cases where the points to be estimated are in the trust region of a local ANN, the detailed ANN provides accurate predictions. Otherwise, the global ANN can still give acceptable results. This property is very beneficial in our adaptive GA framework because the population of a GA starts from a global distribution and evolves to local clusters as the optimization progresses. When more and more solutions are sampled from local regions, local approximation models can be trained in conjunction with a global approximation model to form a hierarchical relationship. Below we present a bootstrap sampling algorithm for building the hierarchical approximation models.

A Bootstrap Multiple Sampling Algorithm. The trust region based hierarchical model is illustrated in figure 5. At the top level, a single sub-model A is globally trained on all sampled training points. The models at the second level are trained on locally sampled points in regions B_1 and B_2. They presumably have better fidelity in these regions. At the third level, two more specific models (C_1 and C_2) provide the highest accuracy in those small regions. In this example, two major local clusters exist, where good solutions exist that must be evaluated more carefully. In practice, a clustering algorithm is used to identify the cluster centers.

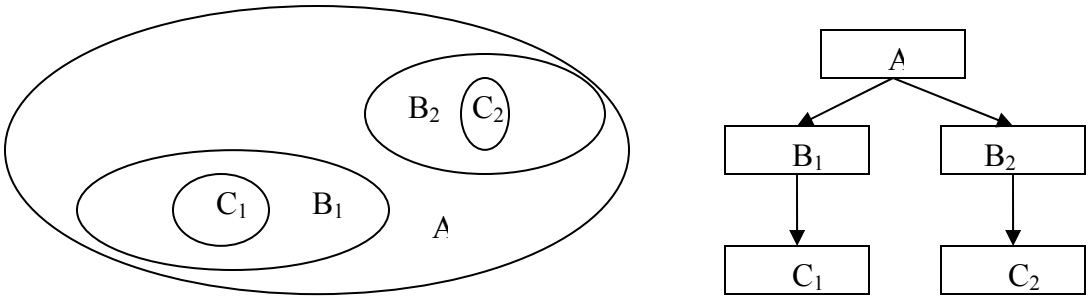

Figure 5. Trust region based hierarchical model

For simplicity, we assume that there is a single major local cluster in the data set (which is true for our remediation case study). Let r_k denote the trust region radius at level k; G_k denote the trust region that has radius of r_k; S_k denote the training set from trust region G_k; and N denote the minimum number of training points to train a model. The following is the pseudo-code for the algorithm,

1. Set $k=1$ and r_k to be the global region's radius R.
2. If the number of points in $G_k < N$, then stop.
3. Compute $r_{k+1} = c^* r_k$, where c is a coefficient in (0, 1)
4. Identify all the training points in region G_k but not in region G_{k+1} and compute the density of those points.
5. Identify training points from G_{k+1} with the same or higher density.
6. Train a model at level k based on training set S_k
7. Set $k = k+1$.

In the algorithm, step 3 computes the trust region radius for the model at the next (lower) level. In our case, c is determined by computing the radius of the new region after discarding the N most outlier points. Step 4 and step 5 identify appropriate training points for each region. Step 4 ensures that no training data at the current level are wasted before moving into lower levels. Step 5 identifies the training points in the lower level region to avoid a hole in the training set. Because the data points may cluster in lower level regions, this step uses density to restrict the number of training points to avoid the current model being over adapted to these regions. Step 5 allows training points to be reused for the lower level models. Thus the algorithm is a bootstrap process and the points close to the cluster center may be used in multiple training sets. Finally, step 6 trains a meta-model based on the just-acquired training set. The iteration is terminated if there are too few points to train the approximation models.

Although the algorithm is developed for our single cluster case study, it can be extended to multiple clusters as well. In this case, a clustering algorithm must first be called to identify the cluster centers. Then at step 3, multiple trust region radiuses are computed. At step 4 and step 5, the identification of training points is done in each of the sub-regions.

Case Study: Umatilla Chemical Depot ("Umatilla")

To demonstrate its performance on a real-world problem, TRAMGA is applied to a field-scale case study at the Umatilla Chemical Depot at Hermiston,

Oregon. A brief summary of the case study is given here; for more details see *Minsker et al.* (2003).

The two major chemical contaminants of concern at this site are RDX (Hexahydro-1,3,5-trinitro-1,3,5-triazine) and TNT (2,4,6-Trinitrotoluene). A pump-and-treat system is in place for the groundwater operable unit and is shown in Figure 6.

To evaluate potential remediation designs, previously-developed numerical flow and transport models are used (*Zheng and Wang*, 1999), which were created with MODFLOW and MT3DMS (*Singh et al.*, 2004). The conductivity of the aquifer is heterogeneous, ranging from 375 ft/yr to 1.8×10^6 ft/yr. The boundary conditions are simulated as constant head along all four edges of the model domain, with net recharge applied to the layer at a rate of 0.0417 ft/yr.

The goal of the optimization is to identify pumping and infiltration rates and locations that minimize costs of the pump and treat system, without exceeding the current capacity of the treatment plant, until RDX and TNT are cleaned up. The cleanup levels for RDX and TNT are 2.1 μg/l and 2.8 μg/l.

The GA population size, determined using the population sizing approach developed by *Reed* (2000), is 160. The GA runs for 60~80 generations before converging, depending on the initial random population. The optimized design solution found by the GA costs 1.66 million dollars.

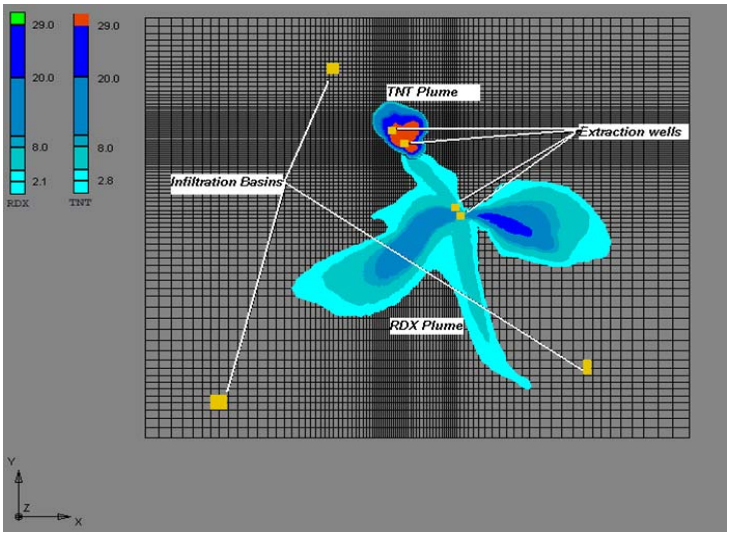

Figure 6. Existing pump and treat system (October 2000)

Previous studies found that the optimal solutions to this case study can achieve the cleanup level within 4 years (*Zheng and Wang*, 2002, and *Peralta* 2002). To save computational effort, we therefore modified the constraints (from the original cleanup target of 20 years) to require cleanup within 5 years. The modified simulation models run in about 2 to 3 minutes on a Pentium-4 1.7G personal computer.

Two meta-models were used to substitute for the flow and transport models. The first one uses the pumping well and infiltration basin locations and pumping rates as inputs. It predicts the maximum concentrations of RDX and TNT in the fifth year, as well as a variable cost component, which depends on the source area concentrations.

The second metamodel uses the same inputs, and predicts the ratios of the maximum concentrations of RDX and TNT in the fifth year to their values in the fourth year.

The ANNs used for the model have 28 units in the input layer and 12 nodes in the hidden layer. The first ANN has three outputs and the second one has 2 outputs. The SVM regression model can have only one output. Therefore five independent SVMs were created for the five outputs. Every SVM has 28 inputs and they all use the radial basis function as the kernel functions. The gamma value for the SVMs is 0.1. Inputs and outputs of the meta-models are scaled to the range *[-1.0, 1.0]*.

Results.

In a previous publication, we have demonstrated the performance of a neural network AMGA (Yan and Minsker, 2004), which uses a single global neural network trained adaptively as with TRAMGA. Here we compare the performance of TRAMGA to the performance of AMGA using both ANNs and SVMs as the meta-models. The PDE sampling parameters for AMGA are from the best settings identified in previous research using a hypothetical groundwater remediation case study.

The make the comparison fair, TRAMGA and AMGA used the same PDE sampling parameters. In the initial training generations (1-3), the entire population is evaluated to generate the initial training set. Then the ANNs or SVMs are trained to replace the numerical models. In subsequent generations, the solutions continue to be sampled at a much lower rate. After enough solutions are generated, the bootstrap sampling algorithm is called to create hierarchical training sets and the corresponding hierarchical surrogates are trained. In this case study, hierarchical training occurs after about 20~30 generations when there are about 800 training points accumulated. Note that TRAMGA and AMGA differ only after the first local surrogate is created. During early generations the two algorithms are identical since they use the same global model for predictions. Hence, the trust region meta-model approach only affects later generations when the GAs are exploring local regions to identify optimal solutions.

Results for ANNs. Figure 7 shows the converged optimal solutions found from eleven random initial populations using ANNs as the metamodels. The number of generations required for converging depends on the initial population, but typically requires about 60~120 generations. For each initial population, AMGA and TRAMGA require similar number of generations (PDE evaluations). However, TRAMGA converged to better solutions (smaller fitness) except for the first initial population, which shows an insignificantly improved fitness over AMGA.

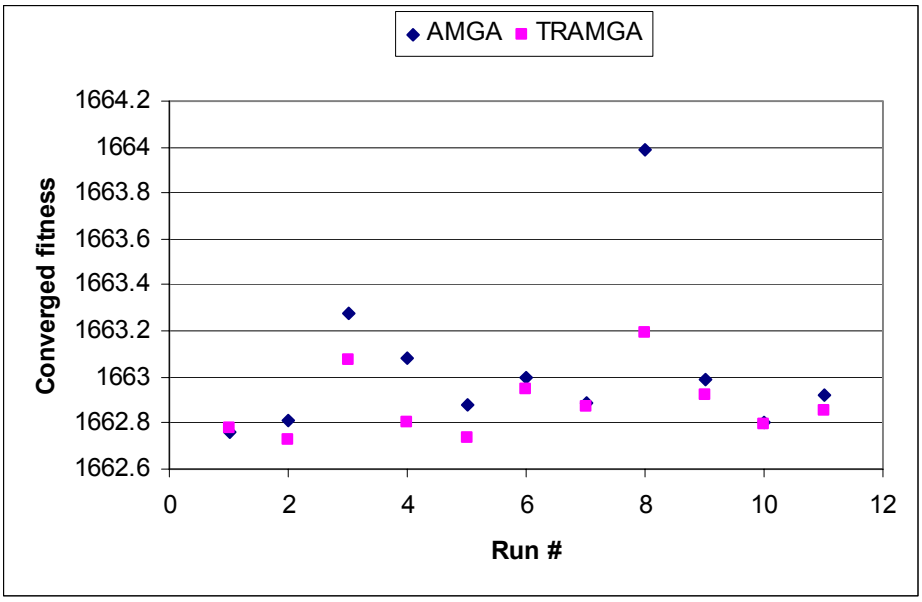

Figure 7. AMGA and TRAMGA results using ANN as meta-models

Figure 8 shows the average converged fitness over the eleven initial populations in each generation for AMGA and TRAMGA. Note that the results before the 18th generation are not shown because the convergence is identical in early generations. The comparison is stopped after 60 generations, after which some of the seeds converged and their fitness values are no longer available. Figure 8 shows that after the trust region models were applied, TRAMGA converged faster and found lower fitness comparing to AMGA.

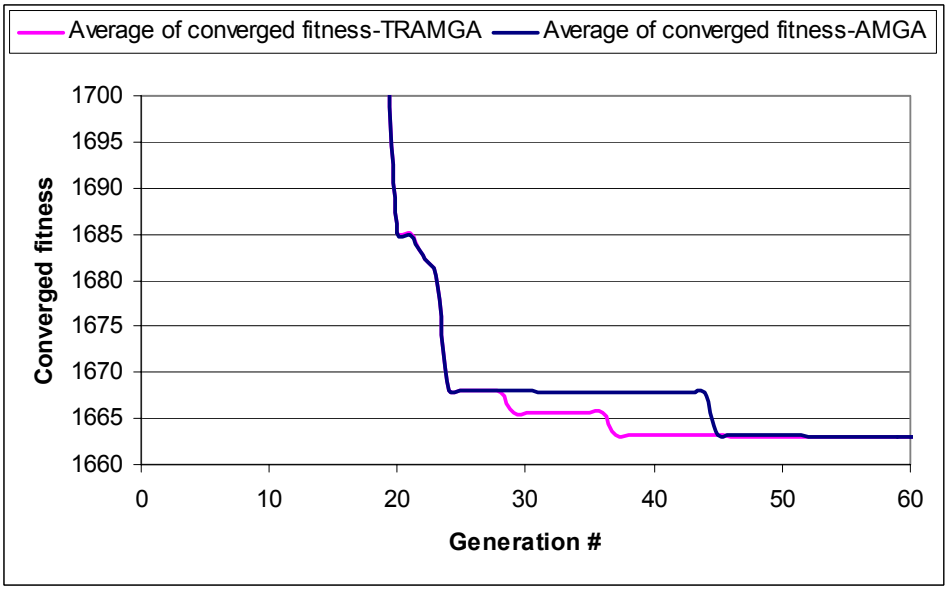

Figure 8. AMGA and TRAMGA converged fitness comparison

Results of SVMs. Figure 9 shows the results finished on for six initial populations using SVMs as surrogates. For most populations, TRAMGA still converged to better solutions than AMGA. Furthermore, the average of converged fitness value for SVMs

(1662.78) is slightly better than ANNs (1663.04). We are currently running more initial populations using SVMs to verify this result.

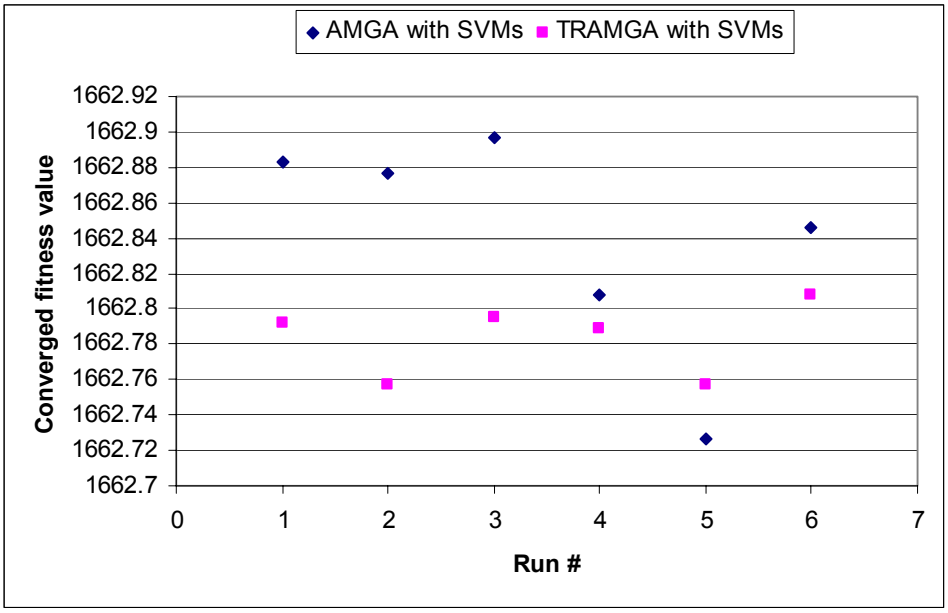

Figure 9. AMGA and TRAMGA results using ANN as meta-models

Conclusion

Complex water resource optimization problems may have complex response surfaces with many local details. Approximating such a surface using a single learning model can be difficult because the learning model has to make a compromise between the global approximation and local details. If the training data points are clustered in local regions, however, local approximation models can be established using the trust region approach proposed herein to better approximate the smoother local details. The bootstrap sampling algorithm proposed in this paper can be used for creating hierarchical approximation models to find a better balance between general structure and local details. The method was applied to an adaptive GA and was tested on a field-scale groundwater remediation design case study using both ANNs and SVMs as meta-models. Our results show that TRAMGA led to more accurate solutions regardless of which meta-models were used.

Acknowledgement

This material is based upon work supported by the U.S Army Research Office under grant number DAAD19-001-1-0025.

References

Alexandrov, N. M., Dennis, Jr., J. E., Lewis, R. M., and Torczon, V. (1998). *A Trust-region Framework for Managing the Use of Approximation Models in Optimization*, Struc. Optimiz., vol. 15, p. 16-23, 1

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

Brooker , A. J., J. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. Trosset. (1998). *A rigorous framework for optimization of expensive functions by surrogates*, Struc. Optimiz., vol. 17, pp. 1–13.
Chang, C.-C. and C.-J. Lin. (2001). *LIBSVM: a library for support vector machines.* Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
Cooper, G., R. C. Peralta, and J. J. Kaluarachchi. (1998). *Optimizing separate phase light hydrocarbon recovery from contaminated unconfined aquifers*, Adv. Water Resour., 21(5), 339-350.
Dougherty, D.E., and R.A. Marryott. (1991). *Optimal groundwater management 1. Simulated Annealing*, Water Resour. Res., 27(10), 2493-2508.
Gopalakrishnan, G., B. S. Minsker, and D. Goldberg. (2003). *Optimal sampling in a noisy genetic algorithm for risk-based remediation design*, J. of Hydroinformatics.
Jin, Y., M. Olhofer, and B. Sendhoff. (2002). *A Framework for Evolutionary Optimization With Approximate Fitness Functions*, IEEE Transactions on Evolutionary Computation, 6(5), 481-494.
Minsker, B. S., Y. Zhang, R. Greenwald, R. Peralta, C. Zheng, K. Harre, D. Becker, L. Yeh, and Yager (2003), K, *Final Technical Report for Application of Flow and Transport Optimization Codes to Groundwater Pump and Treat Systems, Environmental Security Technology Certification Program (ESTCP)*, Available at http://www.frtr.gov/estcp/
Peralta, R. C. (2002). *Optimal Pumping Strategies For Umatilla Chemical Depot RDX And TNT Plumes*, Draft final report presented to Navy Facilities Engineering Command, Systems Simulation/Optimization, Utah State.
Reed., P. M., B. S. Minsker, and D. E. Goldberg. (2000). *Designing a Competent Simple Genetic Algorithm for Search and Optimization*, Water Resour. Res., 36(12), 3757-3761.
Rogers, Leah L., Farid U. Dowla, and Virginia M. Johnson. (1995). *Optimal Field-Scale Groundwater Remediation Using Neural Networks and the Genetic Algorithm*, Environ. Sci. Technol. 29, 1145-1155.
Russel, Stuart, Peter Norvig. (1995). *Artificial Intelligence-A Modern Approach*, Prentice Hall.
Schölkopf, B. (1999). *Support vector learning*, Tutorial given at DAGM'99.
Singh, A., and B. S. Minsker (2004), *Uncerntainty Based Multi-Objective Optimization of Groundwater Remediation at the Umatilla Chemical Depot*, ASCE. EWRI. 2004, Salt Lake City, UT.
Smalley, J. Bryan, B. S. Minsker, and David E. Goldberg. (2000). *Risk-based in situ bioremediation design using genetic algorithm*, Water Resour. Res., 36(10), 3043-3051.
Yan, S. and B. S. Minsker. (2004). *A Dynamic Meta-Model Approach to Genetic Algorithm Solution of a Risk-Based Groundwater Remediation Design Model*, ASCE. EWRI. 2004, Salt Lake City, UT.
Zheng, C. and Wang, P.P. (1999). *MT3DMS: Documentation and User's Guide*, Report to the US Army Corps of Engineers Waterways Experiment Station, (available at http://hydro.geo.ua.edu).
Zheng, C. and Wang, P.P. (2002). *A Field Demonstration of the Simulation-Optimization Approach for Remediation System Design*, Ground Water.