Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Reinforcement Learning for Human-Machine Collaborative Optimization: Application in Ground Water Monitoring

Meghna Babbar-Sebens* and Snehasis Mukhopadhyay+

*Department of Earth and Environmental Sciences, Indiana Univ Purdue Univ Indianapolis, IN 46202
+Department of Computer and Information Science, Indiana Univ Purdue Univ Indianapolis, IN 46202

Abstract

In this paper, we introduce reinforcement learning as a methodology to solve complex multi-criteria optimization problems for ground water monitoring. Multiple analytical criteria are used to assess design decisions and human feedback is simulated by adding random noise. Different learning automata based reinforcement learning methods as well as a genetic algorithm based method are used in experimental studies, which demonstrate the efficiency of reinforcement learning approaches.

1. INTRODUCTION

Freshwater resources are currently under increasing pressure in the United States and in the rest of the world due to growth in human population, increase in economic activity, improvements in the standard of living, and climate change. The complex nature of various driving forces that affect the water quality and water supply problems has led to many nations showing considerable interest in the Integrated Water Resources Management (IWRM) problem-solving paradigm (GWP, 2000). IWRM approaches integrate knowledge from various disciplines and insights from diverse stakeholders in a comprehensive and participatory manner during the problem-solving process.

Previous research in the two scientific disciplines of multiple criteria decision making (MCDM) and multiple criteria decision aid (MCDA) have investigated various strategies for selecting best alternatives based on multiple qualitative preferences of decision makers [e.g., Roy (1990), Munda (1993), etc.]. However, most classical strategies for MCDA and MCDM separate the search (or optimization) and multi-criteria decision making processes. These methods assume that the optimization process for generating alternatives can accurately incorporate various decision making criteria (qualitative and quantitative) through appropriate choice of quantitative objectives and constraints. The alternatives generated are, therefore, assumed to be representative of users' qualitative knowledge and preferences, and fit to be used for multi-criteria negotiations. This assumption is, however, not always true in real-world water resources and environmental applications. In order to address this shortcoming of optimization methods some recent research efforts have proposed optimization algorithms based on genetic algorithm that explicitly, via a transparent interactive framework, include the decision maker's feedback in real time within the optimization process. Such algorithms are called Interactive Genetic Algorithms (Takagi (2001), Kamalian et al. (2004), Babbar-Sebens et al. (2008)). In these interactive algorithms, the human user assesses the comparative quality of the alternatives generated by the optimization technique based on her/his subjective criteria and assigns a preference rank to each of the alternatives. This preference rank is then included within the optimization algorithm as an objective that represent the subjective preferences of the user.

Reinforcement learning provides another very promising framework for problem-solving in semi-programmed and complex decision making environments. In the area of problem-solving for water resources planning and management, previous researchers have applied many

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

techniques such as genetic algorithms, supervised neural networks, and fuzzy set modeling to specific environmental problems. However, the use of reinforcement learning as an approach to solve complex environmental problems has remained largely unexplored.

In this paper, we formulate a groundwater monitoring problem as an identical pay-off game of multiple reinforcement learning agents, each using a conceptually simple learning algorithm. Multiple criteria incorporating various aspects of the desired solution, e.g., the cost of operation and the pollutants' simulated contamination levels, are suitably combined to generate a scalar binary-valued reinforcement signal for the agents. Experimental studies are conducted using different learning algorithms for the agents. These preliminary studies clearly indicate that the approach has the potential of determining high quality solutions. Further, reinforcement learning being naturally suited for stochastic environments, the approach is considerably robust with respect to noise in the computed criteria, which arises naturally in real-world applications and when humans are included in the optimization process. Two of the Reinforcement Learning algorithms explored are also compared with a genetic algorithm, which has been recently used by many studies for ground water monitoring design problems (e.g., Reed et al. 2001, Babbar-Sebens and Minsker, 2008). The results in this paper present some of the preliminary research done in this area and provide investigative comments on the design of optimization strategies for combined human-computer problem-solving environments. Future work will investigate the ability of the solution approach to deal with subjective human-generated reinforcements, in addition to those generated by analytical models.

2. Ground Water Monitoring Case Study

The case study examined in this paper is a ground water monitoring design problem at a 1313 feet by 865 feet BP (formerly British Petroleum) site in Michigan. Ground water at this site was contaminated with benzene, toluene, ethylbenzene, and xylene (BTEX) for a period of 14 years. Active remediation has been completed in recent years and the site has reached a stage where there is a need for long term ground water monitoring. Currently, 36 ground water wells at this site are being used for regular monitoring of the pollution levels. The main objective for this monitoring problem formulated in this paper is to shut off sampling wells that are spatially redundant from a pool of pre-selected 8 wells. A quantile kriging interpolation model (see Goovaerts et al, 1997, for details) was implemented to interpolate the contaminant concentrations throughout the plume. The monitoring decision variables for the problem were the sampling flags (x_i = 0/1) for the 8 wells, where i varies from 1 to 8. Hence, if a flag is 1 then the well at the i^{th} location is sampled. The quantitative objectives for this problem are to minimize the number of wells sampled and to minimize the maximum error between actual benzene concentrations and those estimated with the benzene interpolation models using a smaller subset of wells. It is clear that if we assign one learning automaton for each well to decide whether it is sampled or not, it reduces to an identical pay-off game model of learning automata with multiple criteria corresponding to the multiple objectives. The multiple criteria - benzene error, BTEX error, and the number of wells - are calculated as:

$$\text{Number of wells} = \sum_{i=1}^{n} x_i \tag{1}$$

$$\text{Benzene Error} = \left[\underset{K}{Max} \left\{ Error = \frac{\left| c_j^{\,actual} - c_j^{\,est}(K) \right|}{E_{allow,Benzene}} \right\} \right] \tag{2}$$

$$\text{BTEX Error} = \left[\underset{K}{Max} \left\{ Error = \frac{\left| c_j^{\,actual} - c_j^{\,est}(K) \right|}{E_{allow,BTEX}} \right\} \right] \tag{3}$$

Where, $x_i = \begin{cases} 1 \; if \; well \; i \; is \; sampled \\ 0 \; otherwise \end{cases}$; n = 8; K = number of wells out of 36 wells that are sampled (in other words, used for monitoring); $c_j^{\,actual}$ = actual observed concentration of the contaminant at the j^{th} well in the sampled set of K wells; $c_j^{\,est}(K)$ = concentration of the contaminant at the j^{th} well in the sampled set of K wells, estimated by using quantile kriging;

$E_{allow,Benzene}$ = a user-specified allowable error limit for Benzene (5 parts per billion for this case study); $E_{allow,BTEX}$ = a user-specified allowable error limit for BTEX (100 parts per billion for this case study).

The multiple criteria are scaled to create scaled rewards for each criteria:

$$\text{Benzene Reward} = 1/e^{\left(\frac{\text{Benzene Error}}{\text{Error}}\right)} \quad (4)$$

$$\text{BTEX Reward} = \begin{cases} 1/e^{\left(\frac{\text{BTEX Error} - 2.0}{\text{Error}}\right)}, & \text{if BTEX Error} > 2.0 \\ 1.0, & \text{if BTEX Error} \leq 2.0 \end{cases} \quad (5)$$

$$\text{Number of Wells Reward} = 1/e^{\left(\frac{\text{Number of Wells}}{1}\right)} \quad (6)$$

The total reward that combines all multiple criteria into one single criteria are calculated by weighting each reward with a set of reward weights that have real values between 0 and 1, and sum up to a total of 1.0 for all rewards.

Total Reward = Benzene Reward * Benzene Reward Weight + BTEX Reward * BTEX Reward Weight + Number of Wells Reward * Number of Wells Weight (7)

3. Methodology

3.1 Reinforcement Learning

Originally motivated by mathematical psychology models of animal and child learning, reinforcement learning refers to the ability of an agent to learn long-term optimal behavior through the use of a reinforcement, i.e., an on-line performance feedback from a teacher or environment. The reinforcement, in turn, may be qualitative, infrequent, delayed, or stochastic. There is a rich body of reinforcement learning literature encompassing a wide array of learning models and algorithms. One of the earliest models of reinforcement learning is called a learning automaton [Narendra and Thathachar, 1989] where the agent attempts to learn the optimal action from a finite set using reward/penalty reinforcement from a stationary teacher/environment with unknown reward probabilities. The learning problem is formulated as updating the agent's action probabilities on the basis of trials consisting of an action performed and the reinforcement received. While a wide variety of model-based and model-free learning algorithms has been proposed for a learning automaton with different asymptotic convergence properties, a popular model-free algorithm is the so-called L_{RI} (*Linear Reward-Inaction*) algorithm described by:

$$\begin{aligned} p_i(k+1) &= p_i(k) + \alpha r(k)(1 - p_i(k)) \\ p_j(k+1) &= p_l(k) - \alpha r(k) p_j(k) \end{aligned} \quad (8)$$

where $p_l(k)$ is the agent's probability of choosing action a_l at trial k, a_i is the action chosen at trial k, $r(k)$ is the reinforcement received (with 0 signifying penalty, and 1 signifying reward) , and $\alpha > 0$ is the learning step-size. The idea is to increase the probability of the chosen action linearly if a reward is received (while reducing the other action probabilities) and not to change the action probabilities if a penalty is received. The L_{RI} algorithm has been shown to be ε-optimal, i.e., the asymptotic probability of converging to the optimal action can be made as close to 1 as desired by choosing a sufficiently small step-size α.

While L_{RI} is an example of a model-free learning algorithm (since it does not maintain and use any estimate of the environmental reward probabilities), an example of a model-based learning algorithm is the so-called *Pursuit Learning Algorithm* (PLA) which belongs to the broader class of estimator learning algorithms (Thathachar and Sastry, 1985). In PLA, a learning automaton maintains a vector $\hat{d}$ (with dimension equal to the number of actions), which is an estimate of the true reward probabilities for the corresponding actions. The elements of $\hat{d}$ are maintained as running averages of the rewards received whenever the corresponding actions are tried. At each instant k, the agent determines the action m that corresponds to the maximum element of $\hat{d}$ (i.e., the action that appears to be the best at that instant), creates a unit vector E(k) whose m-th element is 1 and the other elements are 0, and

moves the action probability vector by a small step towards E(k), i.e.,

$$p(k+1) = p(k) + \alpha(E(k) - p(k)) \quad (9)$$

An extension of a single learning automaton is the so-called identical pay-off game of learning automata where a team of automata receive a common reinforcement whose probability depends on the actions of all the automata. It has been shown [Narendra and Thathachar, 1989] that, if each automaton uses the L_{RI} algorithm, the team converges to a mode (local maximum) of the underlying game matrix.

For the ground water case study, the decision variables (i.e. whether i^{th} well is sampled or not, where i varies from 1 to 8 wells at pre-selected locations) make the agents in the reinforcement learning algorithms. The total reward calculated in Equation (7) is used as a reward probability in the two Learning Automata algorithms (L_{RI} and PLA), and a final binary reward of 1 is awarded to the set of learning automata if the total reward is greater than a random number, or 0 otherwise. as a maximizing objective. The population size of 10 was chosen to iterate through 100 generations, in order to have computational expense of evaluating designs equal to the computational expense of the reinforcement learning algorithms. The uniform crossover with a probability of 0.75, mutation rate of 0.01 probability, and tournament selection with no replacement were the evolution operators selected for the SGA.

4. Results and Discussion

Table 1 documents the performance of the L_{RI} and PLA reinforcement algorithms and the SGA for different sets of weights and different levels of uncertainty in the reward functions (Equations (4), (5), and (6)). Uncertainty in reward functions was artificially simulated, for the purpose of examining how each of the algorithm would perform in presence of uncertainty. In these experiments no real human was included in providing feedback for the rewards, however, by including artificial uncertainty we can simulate the performance of the search algorithms in presence of a naturally noisy human feedback. The following modifications were made to the calculated rewards in Equations (4), (5), and (6) to randomly generate multiple uncertain realizations of the rewards:

Benzene Reward in uncertain case =
Benzene Reward (i.e. Equation (4)) ±
Random Number between 0 and 1 ×
Standard Deviation in Noise
(10)

BTEX Reward in uncertain case =
BTEX Reward (i.e. Equation (5)) ±
Random Number between 0 and 1 ×
Standard Deviation in Noise
(11)

Number of Wells Reward in uncertain case =
Number of Wells Reward (i.e. Equation (6)) ±
Random Number between 0 and 1 ×
Standard Deviation in Noise
(12)

Equations 10, 11, and 12 were then summed up to calculate the weighted total reward for the uncertain case, in a manner similar to Equation (7).

A learning step size of 0.01 is used and the algorithms are allowed to proceed to a maximum of 1000 iterations.

3.2 Simple Genetic Algorithm

Simple Genetic Algorithms (SGA) are heuristics-based optimization algorithms that emulate natural selection and genetics to search for optimal designs. They have been extensively used in the water resources problems [Aly and Peralta, 1999 as an example] for solving single objective optimization problems. They work with "strings" of decision variables mapped in binary space (also called "chromosomes"), and search from a population of possible designs ("individuals") using the information provided by the objective function ("fitness function"). Using three probabilistic operators - reproduction, crossover, and mutation - the SGA evolves the population to solutions with higher fitness, until it converged to optimal or near-optimal solution. For the ground water case study, the total reward in Equation (7) was used

In Table 1, we can observe that the PLA algorithm outperforms the L_{RI} algorithm in deterministic and uncertain cases, for the different choice of reward weights and standard deviation in noise. Since the BTEX error in all the cases is lesser than 2.0, hence all the solutions in Table 1 are equivalent in performance with respect to BTEX error (since Equation (5) treats all solutions with BTEX error ≤ 2.0 equally while allotting a reward). With respect to Benzene error and Number of Wells, PLA converges for most of the experiments to better solutions that are cheaper (i.e. fewer number of wells). SGA, on the other hand, performs comparably with respect to PLA, and no clear conclusion can be made for the 8 wells problem.

5. Conclusions

Reinforcement learning constitutes an attractive approach to solve complex multi-criteria problems in the presence of uncertainty, noise, and non-stationarity. In this paper, we presented some preliminary results to demonstrate that it is feasible to use such an approach in the ground water monitoring problem which is used as an example of environmental decision-making problems. While these preliminary studies are inadequate the superiority of reinforcement learning over other heuristic optimization techniques, they at least establish the former as another efficient methodology in the arsenal of environmental scientists. Future work will incorporate real human experts and will deal with other human factors (apart from noise), such as bias, varying expertise, non-stationarity, stress, and fatigue.

References

1. Aly, A. H., and Peralta, R. C. (1999a). "Comparison of a genetic algorithm and mathematical programming to the design of groundwater cleanup system," Water Resour. Res., 35(8), 2415–2425, 1999.
2. Babbar, M. (2006). *Interactive genetic algorithms for adaptive decision making in groundwater monitoring design*. Ph.D. thesis, Environmental Engineering (Civil Engineering), UIUC.
3. Babbar-Sebens, M., and Minsker, B.S. (2008). "Standard Interactive Genetic Algorithm (SIGA):A Comprehensive Optimization Framework for Long-Term Ground Water Monitoring Design," J. of Water Resources Planning and Management, pp. 538-547.
4. Goovaerts, P., (1997). Geostatistics for Natural Resources Evaluation, Oxford University Press, New York.
5. GWP. (2000). "Global Water Partnership: Integrated water resources management". TAC Background paper, Stockholm: Global Water Partnership Secretariat.
6. Kamalian, R. R. Takagi, H., and Agogino, A. M., (2004). "Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation," Proceedings of the Genetic and Evolutionary Computation (GECCO2004), Seattle, WA, 1030-1041
7. Mamdani, E.H. (1974) "Applications of fuzzy algorithms for simple dynamic plant," Proceedings of IEEE, 121, 1585-1588.
8. Munda, G. (1993). "Multiple-criteria decision aid: Some epistemological considerations," Journal of Multi-Criteria Decision Analysis, 2:41-55.
9. Narendra, K.S.,Thathachar,M. (1989).Learning automata: an introduction. Prentice-Hall.
10. Reed, P., Minsker, B.S. and Goldberg, D., (2001). "A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data". Invited paper, Journal of Hydroinformatics, 3, 71-89.
11. Roy, B. (1990). "Decision-aid and decision-making," In Carlos A. Bana e Costa, editor, readings in Multiple Criteria Decision Aid, pages 17-35. Springer-Verlag, Berlin.
12. Takagi, H., (2001). "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," Proceedings of the IEEE, 89(9), 1275-1296.
13. Thathatchar, M. A. L. & Sastry, P. S. (1985). A new approach to the design of reinforcement learning for learning automata, IEEE Transactions on Systems, Man, and Cybernetics , 15, pp. 168-175, 1985.

No Noise				Noise (N(0,0.01))				Noise (N(0,0.1))				Noise (N(0,1.0))			
Benzene (0.9) and BTEX (0.1)				*Benzene (0.9) and BTEX (0.1)*				*Benzene (0.9) and BTEX (0.1)*				*Benzene (0.9) and BTEX (0.1)*			
	Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells
L_{RI}	0.00	0.00	33	L_{RI}	0.00	0.00	33	L_{RI}	0.00	0.00	33	L_{RI}	0.00	0.00	33
PLA	0.00	0.00	33	PLA	0.00	0.00	33	PLA	0.12	1.45	30	PLA	0.05	0.19	32
SGA	0.00	0.20	32	SGA	0.10	0.91	31	SGA	0.02	0.89	30	SGA	0.00	0.24	32
Benzene (0.45), BTEX (0.1), and Numwells (0.45)				*Benzene (0.45), BTEX (0.1), and Numwells (0.45)*				*Benzene (0.45), BTEX (0.1), and Numwells (0.45)*				*Benzene (0.45), BTEX (0.1), and Numwells (0.45)*			
	Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells
L_{RI}	0.00	0.00	33	L_{RI}	0.00	0.00	33	L_{RI}	0.00	0.00	33	L_{RI}	0.00	0.00	33
PLA	0.20	0.21	30	PLA	0.23	0.66	30	PLA	0.22	0.57	29	PLA	0.22	0.57	29
SGA	0.24	1.47	28	SGA	0.24	1.47	28	SGA	0.24	1.47	28	SGA	0.02	0.75	31
BTEX (0.1), and Numwells (0.9)				*BTEX (0.1), and Numwells (0.9)*				*BTEX (0.1), and Numwells (0.9)*				*BTEX (0.1), and Numwells (0.9)*			
	Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells		Benzene Error	BTEX Error	Numwells
L_{RI}	0.10	0.15	32	L_{RI}	0.10	0.15	32	L_{RI}	0.10	0.15	32	L_{RI}	0.00	0.00	33
PLA	0.22	0.57	29	PLA	0.22	0.57	29	PLA	0.24	1.47	28	PLA	0.24	0.90	29
SGA	0.24	1.47	28	SGA	0.24	1.47	28	SGA	0.24	1.47	28	SGA	0.21	0.73	30

Table 1. Solutions found by L_{RI}, PLA, and SGA algorithms for the 8 wells ground water monitoring problem.