DOI: 10.1061/(ASCE)IR.1943-4774.0000268

(Final draft)

Canal Structure Automation Rules Using an Accuracy-based Learning Classifier System, a Genetic Algorithm, and a Hydraulic Simulation Model

Part I: Design

J.E. Hernández[1] and G.P. Merkley[2]

Abstract

Using state-of-the-art computational techniques, a genetic algorithm (GA) and an accuracy-based learning classifier system (XCS) were shown to produce optimal operational solutions for gate structures in irrigation canals. An XCS successfully developed a set of operational rules for canal gates through the exploration and exploitation of rules using a GA, with the support of an unsteady-state hydraulic simulation model. A computer program which implemented the XCS was used to develop operational rules to operate all canal gate structures simultaneously, while maintaining water depth near target values during variable-demand periods, and with a hydraulically stabilized system when demands no longer changed. This model can be applied to canal networks with constant or variable demands within the limits of current hydraulic simulation capabilities. The program output is a set of feasible and optimal operating rules for multiple gate structures, facilitating the automation of open-channel irrigation conveyance systems. Results from sample applications of this technique are presented in a companion paper (Part II: Results).

Key words: canal gate automation; hydraulic modeling; genetic algorithm; classifier system

[1]Research Agricultural Engineer, USDA-ARS-CPRL, Bushland, TX, USA 79012-0010. jairo.hernandez@ars.usda.gov Tel: +(806) 356-5718. Fax: +(806) 356-5750.
[2]Professor, Dept. of Biological and Irrigation Engrg., Utah State Univ., Logan, Utah, USA 84322-4105. gary.merkley@usu.edu. Tel: +(435) 797-1139. Fax: +(435) 797-1248. (✉ *corresponding author*).

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

Background and Previous Work

One way to achieve water management improvements in open-channel irrigation conveyance and distribution systems is through the use of advanced technologies to help make decisions to operate gate structures. A hydraulic model with a mathematical optimization procedure can be used to generate sets of operating rules for canal gate structures. The rules can be custom-developed for a specific canal system. One type of optimization procedure is based on classifier systems which use genetic algorithms.

As an optimization procedure based on Darwin's natural evolution theory, genetic algorithms are a powerful tool to solve different types of problems. Genetic algorithms (GA) are based on a principle similar to "the survival of the fittest," where characteristics of parent rules are transmitted to children by means of reproduction, cross-over, and mutation, and prevalence of the fittest rules (those that perform best in meeting operational objectives). Based on natural selection and genetics, Holland (1975) created the original genetic algorithm theory, and presented the correspondence between rules used in artificial systems to biological chromosomes, detectors or features to genes, and feature values to alleles, among other characteristics. Genetic algorithms are the core for the discovery of new rules and the exploitation of extant rules, and they have recently been used in hydraulic engineering as an optimization tool for reservoir operation for the irrigation of multiple crops (Nagesh et al. 2006), least-cost design of water distribution systems (Babayan et al. 2006), water quality prediction in water distribution systems (Zheng 2006), design of drainage systems (Peng and Jia 2004), design of composite channels (Jain et al. 2004), scheduling of water pipe replacement (Dandy and Engelhardt 2001), water distribution systems (Simpson and Wu 2001), and location of control valves in pipe networks (Pezzinga et al. 1999).

Accuracy-based classifier systems (XCS) represent a major recent development in classifier systems research. Since its origin (Wilson 1995), XCS has provided repeatable results that are generally better than those produced by the majority of models developed since Holland's initial work (Butz et al. 2004). Accuracy-based classifier systems represent the knowledge extracted from a problem as encapsulated in a set of rules. A rule is a prescribed mathematical method for taking actions to achieve operational objectives. A rule set is incrementally evaluated by means of interactions with an external environment through a reinforcement learning process, and it is improved by a search mechanism based on a GA (Bernado-Mansilla and Ho 2005). Models of physical phenomena (e.g. hydraulic simulation models) can help an XCS to determine whether an action is plausible or not, by means of a reinforcement program. The reinforcement program assigns rewards and penalties according to system behavior (real or simulated) after the application of a specific action.

Until now, XCS had not been used for developing general operating rules for canal gate structures. However, GAs have been used for decision support in irrigation project planning (Kuo et al. 2000) where relative crop yield and water demand information was applied to maximize the projected benefits. Genetic algorithms have also been applied for optimal seasonal furrow irrigation (Montesinos et al. 2002), where a model produced an irrigation season scheduling for maximizing farmer's profit, and along the same lines, GAs have been applied for optimizing off-farm irrigation scheduling (Nixon et al. 2001). More recently, a classifier system supported by a GA has been developed for rule-based operation of canal gates (Chittaladakorn and Merkley 2005).

Introduction

This paper provides a detailed description of the elements that comprise an XCS model, a GA, and the hydraulic model used in the study. A companion paper presents results from the XCS model for different operational scenarios in various simulated canal systems. The hydraulic model was used to predict the response from the application of different gate operation rules as generated by the XCS and GA. The response was evaluated through the XCS so as to develop a set of gate operating rules which could be applied to successfully control the hydraulics of a given canal system, achieving specified operational objectives. The principal operational objective was the maintenance of stable canal water depths.

Both trapezoidal and rectangular canal cross sections were used in this study. Structures included in-line gates and turnouts (water delivery points). In-line structures were located at the upstream and downstream ends of each canal reach, and they included rectangular gates and weirs. Turnouts directed water from the simulated canal system to delivery points outside of the system, and they were always located near the downstream end of each canal reach. Turnout demand hydrographs were generated randomly to simulate various operational conditions in the canal system and provide a more generally applicable rule set. Each hydraulic simulation began with all canal reaches full of water and steady-state flow conditions. The delivery of water through turnout structures caused unsteady hydraulic conditions and the XCS was tasked with developing rules to operate the in-line structures in a way that would best achieve the operational objectives and re-establish steady-state hydraulic conditions. Following a period of changing turnout delivery demands, the rules were supposed to rapidly bring the canal system to a new steady-state condition.

A computer program was written in the C# .NET language for personal computers (PCs) to implement the model. Minimum computer requirements are a processor similar to the Pentium II CPU 398 MHz, and 1 GB of RAM. Data were input and output using a console interface and saved in database and text files.

The Classifier System

An XCS was combined with a hydraulic model to simulate the response from a set of canal gate operation rules. The hydraulic model also provided information from which rewards and penalties were computed through the XCS reinforcement program. The XCS acted as a reinforcement learning agent, meaning that it learned to perform a task through trial-and-error interactions with an unknown environment which provided feedback in terms of numerical reward. The definition applied in this research corresponded to a classifier system that learned actions for the adjustment (operation) of gate structures, when it received rewards and penalties according to variations in canal water depth relative to specified target depths, hydraulic stability, and demand-supply deviation, among other aspects of the reinforcement program as indicated below. Those interactions occurred continuously during each hydraulic simulation. The input to the XCS was the current hydraulic state of the canal system, also referred to as the environment. The output from the XCS was a set of actions which was applied to the canal system, and the hydraulic response was evaluated by means of a reward, a penalty, or a combination of both. The goal of the XCS was to maximize the rewards by developing a set of optimal operational rules for the gate structures. Herein, the definition of an “action” is simply whether to open or close a gate structure, and the magnitude of the setting change (which could be zero if there is to be no gate setting adjustment).

Classifiers in this XCS consisted of a condition, an action, and three main parameters. The main parameters were the *prediction*, the *prediction error*, and the *fitness*. The prediction estimated the average payoff that the system expected when a classifier was selected for an operational action. The payoff was an outcome obtained from the system according to rewards or penalties deserved for action application. The prediction was computed as a weighted-average fitness of all classifiers that matched the current environment, and which were invoked by the current action. The prediction error gave the average error between the classifier's prediction and the received payoff. The fitness was an estimate of the classifier's accuracy relative to other classifiers, and the classifier's accuracy was calculated from the prediction error. All of these data were stored in a population of rules. The population contained possible solutions that could solve the operational problem and each classifier corresponded to a potential solution for one particular environment, which might or might not be evaluated through the process, depending on whether the classifier was selected as an action to be applied to the canal system. As the population grew, the average classifier fitness increased if all the parameters were defined adequately, whereby the population was evolving.

Through rules, conditions and actions interacted with the canal system, the status of which is known as the environment. In the computer program which implemented the XCS, the condition, action, and environment were coded independently as binary sequences (or "strings"). Rules in the population were made up of pairs of condition and action strings. The three main parameters, as described above, guided the evolution of the process with the help of a GA. A condition specified the input states that the classifier could have, which could be from a real or hypothetical canal system. Hydraulic conditions can be sensed or calculated from the canal system to provide input to the XCS in the form of an environment string. Each environmental situation had a corresponding condition value.

The XCS determined the implementation of actions upon the environment. The role of the XCS was to decide what set of actions the system should take to reach the operating goals, and it selected one set of actions at a time to operate the canal system. One set of actions corresponded to a set of gate structure settings for every gate in the canal system. These settings were applied to the hydraulic system as simultaneous gate operations. There were many potential actions that might be selected for application, but the XCS used only one set at a time. For one specific condition, there might be many

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

actions according to the number of gates to operate, and the number of different gate settings. All actions were created randomly from pseudo-random numbers.

A schematic view of the XCS general architecture is presented in Fig. 1. The computer model started with initializations for the hydraulic status of the canal system and generated a random population of rules. The XCS created what is referred to as a matching rule set (described below), a prediction array, and an action set. One set of actions was selected to be applied to the gates in a canal system through a hydraulic model. Therefore, the XCS relied on a hydraulic model, as well as on the reinforcement program, which had an embedded GA, to evaluate the performance of the entire canal system and decided whether to continue improving the population or not. When the population matured, there was little or no improvement in its strength and at this time the learning process was complete.

Matching Procedure

After initializations, the current population was compared to the environment to create a “match set,” which was composed of classifiers whose condition corresponded to the hydraulic situation in the canal system. The condition parts of the classifiers (rules) were compared one-by-one to the current environment. If a particular classifier corresponded to the environment, it was added to the match set; otherwise it was ignored and the process continued until considering all members of the population. Every time there was a match, a counter was incremented to keep track of the match set size. If the match set was smaller than a specified threshold value, after comparing all classifiers from the population, a “covering procedure” was invoked as described below.

The covering procedure was used to guarantee that there was a sufficient number of classifiers that matched the environment. Therefore, when covering was required, classifiers were created in such way that conditions were equal to the environment, and actions were generated using pseudo-random numbers. Classifiers generated by covering were added to the match set, and to the current population. If the population was considered to be too large, the same amount of classifiers generated by covering were removed from the population, deleting the classifiers with the lowest fitness.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at Journal of Irrigation and Drainage Engineering, published by American Society of Civil Engineers. Copyright restrictions may apply. DOI: 10.1061/(ASCE)IR.1943-4774.0000268

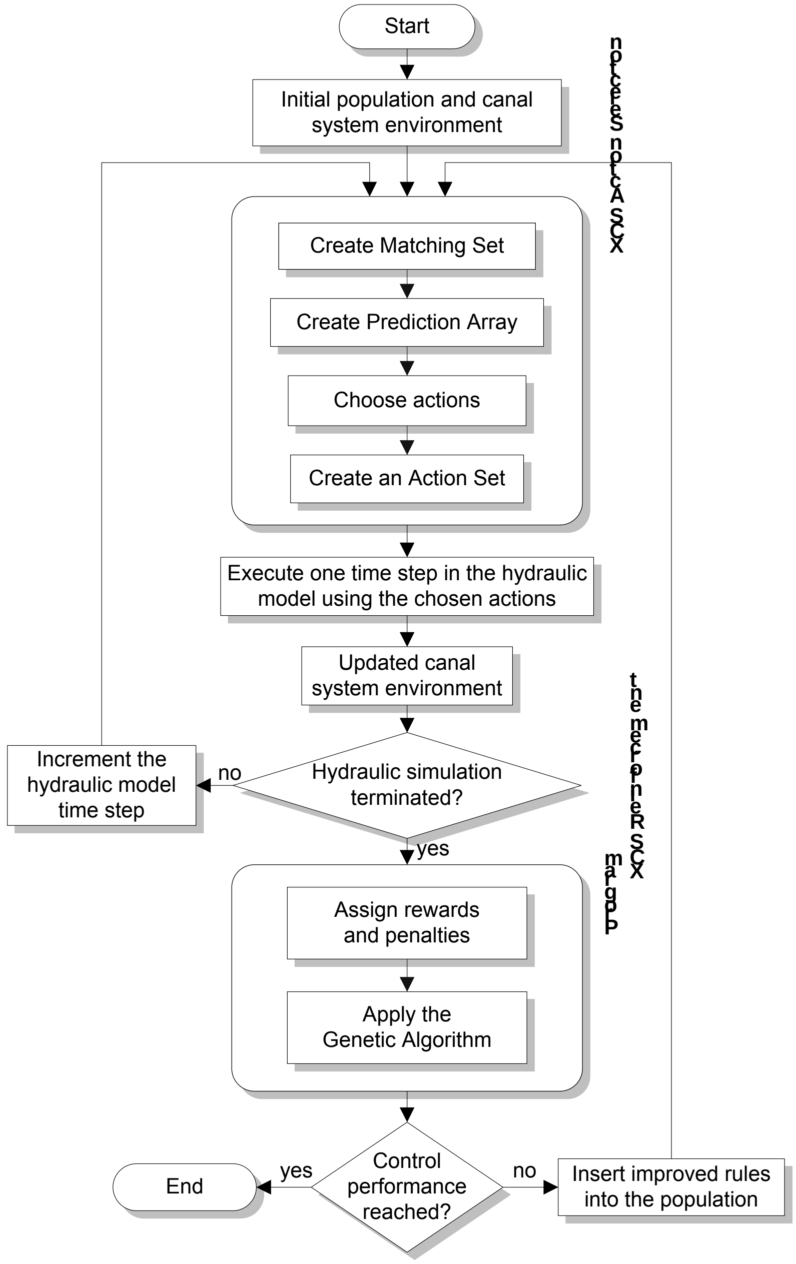

Fig. 1. General architecture for generating canal structure automation rules

System Prediction and Action Selection

The *prediction* estimated the payoff that the system expected if a given classifier were used to take an action in the canal system. At every time step, the XCS created a new matching set, [M], which contained the classifiers that matched the current environment. For each action in the match set, the

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

XCS computed the system prediction *P(a)*. The system prediction was computed as the weighted-average of the predictions based on classifier fitness, as shown in Eq. 1:

$$P(a) = \frac{\sum_{cl.=cl.\wedge cl.a\in[M]} (cl.p)(cl.F)}{\sum_{cl.=cl.\wedge cl.a\in[M]} (cl.F)} \tag{1}$$

where *cl.a* is the action of classifier *cl*; *cl.p* is the prediction of classifier *cl*; and, *cl.F* is the fitness of classifier *cl*.

The summations in Eq. 1 were performed over all classifiers that invoked action *"a,"* and which belonged to match set [M]. The different values of P(a) were collected in the prediction array, which was the basis for action selection. In each case, the XCS selected the action with the greatest system prediction from the values in the prediction array. Then, all classifiers in the match set which contained this action were added to the action set.

Parameter Updates

During the learning process, the prediction, the prediction error, and the fitness were periodically updated. As a result, fitness was computed for every rule in the action set. Initial values for prediction (p) were very close to zero for the initial population. The prediction, which gave an estimate of the payoff that the system was expected to gain when the classifier was selected, was updated as follows:

$$p = p + \beta(R - p) \tag{2}$$

where β is the learning rate ($0 < \beta \le 1$); and, R is the desired reward, which was computed as:

$$R = r + \gamma\left(P(a)_{max}\right) \tag{3}$$

in which r is the previous reward; γ is the discount factor ($0 < \gamma \le 1$); and, $P(a)_{max}$ is the classifier's maximum prediction for this action.

The prediction error (ε), which estimated the precision of the classifier's prediction, was updated as follows:

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

$$\varepsilon = \varepsilon + \beta\left(\left|R - p\right| - \varepsilon\right) \tag{4}$$

To update the classifier fitness, the accuracy (κ) was determined as follows:

$$\text{if } \varepsilon < \varepsilon_{0,} \text{ then } \kappa = 1 \tag{5}$$

otherwise,

$$\kappa = \alpha\left(\frac{\varepsilon}{\varepsilon_0}\right)^{-\nu} \tag{6}$$

If the prediction error (ε) was below the threshold error (ε_0), the classifier was said to be accurate, meaning that had an accuracy of one, and errors were regarded as having equal accuracy; otherwise, the accuracy (κ) dropped off quickly, depending on the values of α and ν, which were parameters used to define the exponential function during the model calibration period.

The accuracy values (κ) for the classifiers in the action set were converted to relative accuracies (κ') as follows:

$$\kappa' = \frac{\kappa}{\sum_{x \in [A]} \kappa_x} \tag{7}$$

The relative accuracy was used to compute the classifier fitness (F), which was an estimate of the classifier's accuracy relative to other classifiers in the action set [A]. The fitness evaluated the accuracy of the prediction. Classifier fitness was updated toward the classifier's current relative accuracy as:

$$F = F + \beta(\kappa' - F) \tag{8}$$

Genetic Algorithm

The GA selected two classifiers with probabilities proportional to their fitness. These classifiers acted as parents to generate two children. Initially, one child was identical to one of the parents, and the second child was the same as the second parent. Subsequently, crossover and or mutation could take place in the children with specified probabilities χ and μ, respectively. Two-point crossover was applied and one random bit was mutated whenever crossover and or mutation were/was invoked, respectively. The children were inserted into the population to compete with their parents and all other members of the population. The prediction and the prediction error for child rules were set as the respective averages of the two parents. The fitness was set as 10% of the average parent fitness. A “numerosity” value was defined as the number of identical (redundant) classifiers in a population and it was created to reduce the database size by manipulating the magnitude of this parameter while deleting repetitions. The child numerosity, which was the number of identical classifiers in the population, was given a value of one. The children’s experience value, which was the number of times that the child participated as parent in a genetic algorithm, was set to zero. When a member of the current population was found to be identical to a generated child rule, the classifier’s numerosity for the existing member was incremented by one, and the child was not added to the population.

In this study, different initial populations were tested. The first versions contemplated huge populations containing millions of members which were stored in a relational database to accommodate the size. The XCS main parameters were not randomly selected; instead, default values were assigned for the initial population. The approach of starting with a huge population had two disadvantages. First, it was time consuming, because all members were generated bit-by-bit using pseudo-random numbers, and second because database input and output consumed a large amount of computer processing time. After trying different sizes smaller than the initial approach, with thousands of members, and without success, a tiny population of ten members was selected as an initial population. The advantage of this approach was that a small population could be defined with relatively strong rules, its generation took less than one second (instead of hours), and database processing was more streamlined. The presence of “good” rules in the initial population was found to be irrelevant, because the population quickly grew in size, and the XCS created rules with better fitness in the long run. Therefore, the final version for the initial population was one with only ten members, and it was generated randomly.

The population usually grew each time the GA was applied. Due to the small size of the initial population the deletion of classifiers from the population was disabled, thereby allowing the population to grow. Default values for the main XCS parameters were selected according to Butz and Wilson (2000), which established that prediction, prediction error, and fitness should have very small values. Thus, the default value used for prediction was 0.05; the prediction error was 0.001; and, the fitness was 0.001.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at Journal of Irrigation and Drainage Engineering, published by American Society of Civil Engineers. Copyright restrictions may apply. DOI: 10.1061/(ASCE)IR.1943-4774.0000268

Local and Global Hydraulic Variables

The variables of interest to run the program were defined by taking into account that they had to satisfy operational objectives. Local hydraulic variables were those that were taken in close proximity to a given gate in the canal system, and global variables were based on system-wide conditions. A strong emphasis was given to the balance of local and global hydraulic variables and their impact on the classifier system as a whole. Operational actions were influenced by the magnitudes of hydraulic variables and they were applied locally as well as globally, depending on how beneficial they were for individual canal reaches, and for the entire system. Hydraulic indicators governed the actions to be applied to the canal system and included the following local and global variables:

Local Variables:

- Deviation from the target depth for the previous and current time steps of the hydraulic model;
- A local hydraulic stability index;
- Current gate structure setting;
- Average velocity change between hydraulic model time steps in a reach; and,
- Infeasible operations.

Global Variables:

- Supply flow deviation from required amounts of water delivery at turnouts;
- A system hydraulic stability index;
- Average velocity change between consecutive canal reaches;
- Average velocity change between hydraulic model time steps in a reach; and,
- Average discharge change at gate structures throughout the system.

Condition String

The condition string was the concatenation of substrings which defined the magnitude of a particular variable (local or global). Every substring was a concatenation of binary bits. A substring defined an instance value that a variable represented. The condition string contained variables that

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

described the hydraulic status of the canal system. However, a given condition string might or might not describe a physically feasible situation because each string was randomly generated. For the purpose of constructing the condition string some local and some global variables were used, as follows:

- Supply flow deviation from total water demand;
- System hydraulic stability index;
- Average velocity change between consecutive reaches;
- Current depth deviation from the target; and,
- Current structure setting.

The current depth deviation and gate setting were computed for every “pivot point” and every gate, respectively. The pivot point was the location in a reach where a target depth was specified, and was always either the upstream or downstream end of a reach. This implied that larger canal systems had larger condition strings than small systems, because they have more gates and pivot points. These two parameters were included in the condition string as local variables, but they behaved as a single global variable. The current hydraulic state for the canal system, known as the environment string, had the same format as a condition string, but it represented a particular situation which was currently occurring in the canal system at that time step. This current state was translated from the hydraulic model results to a string, which was subsequently matched to one or more condition strings in the population.

Substrings were the supply flow deviation from delivery demands, the system stability index, the average velocity change between consecutive reaches, the water depth deviation, and the current gate setting for the time step in consideration. The substring representing the current gate setting was located along the condition string immediately after the depth deviation partial string. These two substrings were repeated as many times as necessary in the canal system, but their magnitudes were different, in general.

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

Action String

The action string was a concatenation of actions taken for all gates in the canal system. Each part of this concatenation contemplated the option of opening or closing each gate by a specific amount. Therefore, the length of the action string depended upon the number of gates in the system. The virtue of this concatenation was that it combined all local actions into a single string, and at the same time it had a global context as it was a complete set for the entire system under consideration. Global actions generated widespread structure operations involving one or several structures along the canal system. Local actions generated a change in structure settings which were closer to the place in reference.

Local actions produced an effect upstream and downstream structures. Global actions were applied over the whole canal system. Thus, both global and local actions merged as a unique action for each structure in the system. The resultant action was applied over structures located throughout canal reaches between the upstream source of water and the downstream end of the last reach. Of course, in some cases a local or global action was "maintain the previous setting, and do nothing." The final action was bounded depending on the current gate setting to disable possible actions which could go beyond physical limits.

Hydraulic Model

The RootCanal hydraulic model (Merkley 2007) was used to interact with the XCS as an environment that changed according to applied gate structure actions, as shown in Fig. 1. The hydraulic model predicted how the system responded, hydraulically, to gate setting actions generated by the XCS. The hydraulic model was condensed into a dynamic link library and was used to define the canal system, including reaches, structures, and other system features. The first interaction of the hydraulic model with the XCS was to compute the initial situation. This response from the hydraulic model was an input to the XCS after conversion to an environment string. Each time an action was selected by the XCS, it was passed to the hydraulic model, and based on the simulated response from the canal system, rewards and penalties were applied by the reinforcement program. The selected action was a set of particular operations which were applied to different gate structures in the canal system. With these new structure settings, the hydraulic model computed a new current system status, and as a result, the following variables were updated: water depths, average velocities, reach inflows and outflows, stability indexes, and structure settings. This new situation was evaluated by the reinforcement program, and it was used to analyze the classifier population's evolution process.

Computed values were the basis for comparison for reward and penalty estimation by means of the reinforcement program. Depths were compared to target values and depth deviations were calculated. Velocities and discharges were compared to determine how fast the system status was changing, and how close it was to a stable hydraulic condition. Current turnout delivery demands were compared to previous demands as a measure of operational performance. Gate settings were used to detect the magnitude of changes and the occurrence of infeasible operations.

Reinforcement Program

Rewards and penalties were assigned depending upon rule matching to the environment. The magnitude of these rewards or penalties depended upon the impact of global and local variables on the canal system. The core of the "apportionment of credit" (reward and penalty) algorithm was based on reaching and maintaining water depths at the respective target values, and on minimizing the demand-supply deviation and stability index for the whole canal system. In addition, the velocity changes between consecutive reaches, and the velocity change in time for every reach through the canal system, had a strong influence over the apportionment of credit calculation.

When assigning rewards, it was important to keep the objective function in mind. The main role of the objective function was to minimize water depth fluctuations generated by gate operations, while maintaining the water depths at or near the respective targets. The objective function was also designed to minimize the hydraulic stabilization time after the completion of a series of demand hydrograph changes. Rewards were based on both global and local system responses to a particular gate setting action (opening or closing).

Local Rewards

Local rewards were applied per structure depending upon the variable used in the analysis, and the location in the canal system. Local rewards were applied to gates to reinforce local behavior, as global rewards did for the entire canal system.

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

Depth Deviation from the Target. After running the hydraulic model following a given gate structure setting change, the model recomputed water depths in each reach. These water depths were compared with the target depths, and a deviation was computed to show how far the current water depth was from the target depth. Whenever the current depth was at the target depth in a given reach, a full reward was applied for the corresponding action.

Deviations from the target depth were positive when current depth was above the target, and negative otherwise. When the last deviation was compared with the previous deviation, a response from the system was analyzed, and local rewards assigned. When the current water depth was moving toward the target after a change in gate structure setting, a partial reward was applied proportionally, taking 100% for the last deviation as a basis for computation. In the case of target depth overshooting (changing the sign of the deviation from positive to negative, or vice versa), the reward was computed as mentioned above, but it was reduced according to the percentage of overshoot. For the case where the previous depth was closer to the target depth than the current depth, a penalty was assigned instead of a reward. Whenever a previous depth deviation was equal to the current deviation, no reward was applied, irrespective of how far the current depth was from the target.

A mechanism was developed to maintain water depths constant after they have reached the respective target values. For every reach, a “flag” variable, C_r, was computed for the cases when the current depth was equal to the target depth. The flag variable was incremented by one each time the current depth remained at the target depth, and it was set equal to zero when the water depth did not remain at the target depth since the last time step. When the calculated depth was maintained at the target depth a higher reward (a bonus) was assigned. To avoid excessively large rewards with respect to other parameters, the reward for reaching the target depth was assigned an upper limit.

When the water depth reached the target the first time, it was rewarded by a basic amount, R_b, and if it was maintained at the target depth, the reward grew exponentially as follows:

$$R_{tl} = R_b + C_o C_r^n \tag{9}$$

where R_{tl} is the reward for reaching the target depth; R_b is the base reward; C_o is a constant; C_r is the number of consecutive time steps remaining at the target; and, n is a constant (n > 1).

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

Local Stability Index. The reward from the stability index (SI) was a function of the current and target water depths. A zero stability index was rewarded in full if the current water depth reached the target depth. Whenever the current depth was below the target depth, a positive stability index was desirable because it increased the amount of water coming into the canal system at the source, leading to an increase in the depths, in general. If the current depth was above the target depth, decreasing the amount of water entering the system was a way to deplete water depths, and in this case a negative stability index was rewarded.

Demand supply deviation at turnouts was a factor that influenced the stability index. An extra reward was assigned for the case when demand was equal to the supply, having a stability index of zero, as well as a current water depth at the target depth. This extra reward was because this is the most desirable situation when considered on a local basis.

To assign a reward for local stability index, R_{SI}, the absolute value of the stability index, |SI|, was used. The absolute value of the variable was required, because the reward should be the same for the case when outflows were greater than inflows, and vice versa. As the system approached a hydraulically stable condition, the stability index approached zero. The stability index had a range from -1.0 to +1.0. The reward for local stability index was inversely proportional to the absolute value of the stability index. As the stability index reached zero when inflows were equal to outflows, the local stability index reward, R_{SI}, reached its maximum value of C_{si}. To compute the local stability index reward, the absolute value of the stability index, |SI| was affected by a coefficient, C_{si}, and an exponent m which was used to weight it with respect to other rewards, as:

$$R_{SI} = \frac{C_{SI}}{1+\left|SI\right|^{m}} \tag{10}$$

Reach Velocity Change between XCS Time Steps. Reach velocity change between XCS time steps represented how fast the water was flowing through the system according to the magnitude of the last action applied to the canal system. In order to assign rewards for reach velocity change between XCS time steps, current and target depth were considered. Rewards were applied for velocity changes that reduce the difference between the current depth and the target. Whenever the current depth was below the target depth, a positive reach velocity change between simulations was rewarded. A null reach velocity change between simulations generated a reward, if the current depth was at the target depth. When the target depth was below the current depth, a negative reach velocity change leaded to a decrease in the depth, and this situation was rewarded.

Global Rewards

Global rewards had a major impact on the environment because they involved one, many, or all gate structures in the canal system. These rewards were applied spatially over all of the structures in the canal system, and they were based on the criteria described below.

Demand Supply Deviation. System demand and supply were computed by the hydraulic model after an action was applied. Whenever demand equaled supply, a maximum reward was granted. If there was a deviation between demand and supply, proportional rewards were assigned according to the deviation. Demand supply deviation values greater than a threshold value did not generate any reward. This applied to the apportionment of credit in general, so that if the absolute turnout demand supply deviation was greater than a specified threshold value for the deviation, no reward was applied.

Global Stability Index. To compute the global stability index, total inflow and outflow from the system were considered. Reward from global stability index was also a function of demand-supply deviation relationship. A zero stability index was rewarded in full if demand was equal to supply, otherwise a partial reward was applied. Partial rewards were applied depending on the sign of the stability index. Whenever the demand was greater than the supply flow rate, a positive stability index was desired because it helped compensate for the volumetric deficit. Thus, a positive stability index generated a partial reward when the demand was greater than the supply of water to the system. On the other hand, if supply was greater than demand, a negative stability index was partially rewarded. The most desirable situation considering the entire system was the case when the current water depths were within a dead band, and the stability index was zero. An extra reward is granted whenever these goals were met.

Change in Gate Discharge between XCS Time Steps. The magnitude of the change in discharge through gate structures between XCS time steps was indicative of the impact of the last applied set of actions over the whole system. Higher magnitudes of change meant that the last set of actions was more drastic relative to the current set of actions. A zero value meant that there was no net discharge change after the last action was applied. Rewards for change of discharge through gates between XCS time steps were applied according to the magnitude of the demand supply deviation. Greater demand supply deviations generated larger rewards for larger discharge changes than for smaller changes.

Velocity Change between Reaches. This parameter indicated how fast the water was moving through the system and, consequently, how fast the system responded Full reward was applied for zero velocity change between reaches when demand equaled supply. For positive demand supply deviations, meaning demand greater than supply, rewards were applied for positive changes in velocity between reaches. For a case when supply was greater than demand, rewards were applied for decreasing velocities.

Therefore, applying the superposition principle to all mentioned rewards, the total reward was defined as the summation of rewards that apply to a particular gate structure. A gate structure had different sources of rewards depending on its particular condition. It had rewards granted locally, globally, or both at the same time.

Penalties

Penalties were assessed as a function of the number of consecutive times the water depth had not reached the target depth and the absolute value of the stability index. Infeasible structure adjustments were also penalized.

Penalty for Leaving the Target Depth. A large penalty was assessed when a given rule caused the calculated depth to deviate from the target depth after it had been reached in the previous time step, and this penalty was increased if it was not achieved consecutively. After reaching the target depth, any deviation from the target depth was penalized. In this case, the penalty was computed as:

$$P_{tl} = P_b C_m^k \tag{11}$$

where P_{tl} is the penalty, P_b is the base penalty, C_m is the number of times missing the target depth since last time it was at the target; and, k is a calibration exponent, which is less than 1.0. The depth was

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

considered to be correct whenever the calculated value was within a ±8% dead band around the target depth.

Penalty for Instability. An instability penalty, P_{SI}, was apportioned when the system had reached a hydraulically stable conditions, and the stability was lost during the next simulation. For large differences in the stability index between two XCS time steps, greater penalties were applied. Low penalties were considered when the stability index difference value was close to zero and maximum penalties when it was close to 1.0. The penalty was proportional to aforementioned difference, and it was adjusted according a coefficient, C_{ps}, and a specified exponent, u, to make the penalty proportional to other penalties.

$$P_{SI} = C_{ps} \left| SI_{t+1} - SI \right|^{u} \tag{12}$$

Penalty for Infeasible Operation. A structure operation penalty was computed when a rule corresponded to an infeasible operational adjustment. These were the cases when a structure with a current small opening was required to be closed by an amount greater than what was physical permissible, when a completely open structure was required to be opened an additional amount, or when a specified gate structure operation forced a flow regime change (e.g. orifice to non-orifice flow) at the structure.

Summary

An accuracy-based learning classifier system (XCS) and a multi-objective genetic algorithm were developed using an unsteady-flow hydraulic model to simulate operational conditions in irrigation canals. The model was defined to answer the question of generating operational rules that maintain water depths inside a dead band, surrounding a target depth, with acceptable stability. The XCS was developed to produce acceptable operational solutions for gate structures in canal systems based on a specified objective function. A multi-objective function was defined to minimize: (1) water depth fluctuations; (2) the absolute value of the stability index; and, (3) the demand-supply flow rate difference. The resulting set of operational rules obtained from the XCS can be applied in the field by deploying them in a data-logger, which will require the development of code to communicate the recommended gate structure adjustments, according to the current status of the canal system.

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

References

Babayan, A. V., Kapelan, Z. S., Savic, D. A., and Walters, G. A. (2006). "Comparison of two methods for the stochastic least cost design of water distribution systems." *Engrg. Optimization*, 38(3), 281-297.

Bernado-Mansilla, E., and Ho T. K. (2005). "Domain of competence of XCS classifier system in complexity measurement space." *IEEE Trans. on Evolutionary Computation,* 9(1), 82-104.

Butz, M. V., and Wilson S.W. (2000). "An algorithmic description of XCS." *Advances in Learning Classifier Systems: Proceedings of the Third International Workshop*, Lanzi, P., Stolzmann, W., and Wilson, S., editors. Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin Heidelberg, 1996, 253-272.

Butz, M. V., Kovacs, T., Lanzi, P. L., and Wilson, S. W. (2004). "Toward a theory of generalization and learning in XCS." *IEEE Trans. on Evolutionary Computation,* 8(1), 28-46.

Chittaladakorn, S., and Merkley G. P. (2005). "Classifier system for rule-based operation of canal gates." *J. Water Resour. Planning and Manage.,* 131(1), 3-13.

Dandy, G. C., and Engelhardt, M. (2001). "Optimal scheduling of water pipe replacement using genetic algorithms." *J. Water Resour. Planning and Manage.*, 127(4), 214-223.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* The Univ. of Michigan Press, Ann Arbor, MI.

Jain, A., Bhattacharjya, R. K., and Sanaga, S. (2004). "Optimal design of composite channels using genetic algorithm." *J. Irrig. Drain. Engrg.*, 130(4), 286-295.

Kuo S.F., Merkley, G.P., and Liu, C.W. (2000). "Decision support for irrigation project planning using a genetic algorithm." Agricultural Water Manag, 45(3), 243-266.

Merkley, G.P. (2007). "RootCanal: Hydraulic simulation model." Users' Guide and Technical Reference. BIE Dept., Utah State Univ., Logan, UT.

Montesinos, P., Camacho, E., and Alvarez, S. (2002). "Application of genetic algorithms for optimal seasonal furrow irrigation." *J. Hydroinformatics,* 4(3), 145-156.

Nagesh, D., Srinivasa, K., and Ashok, B. (2006). "Optimal reservoir operation for irrigation of multiple crops using genetic algorithms." *J. Irrig. Drain. Engrg.*, 132(2), 123-129.

Nixon, J. B., Dandy, G. C., and Simpson, A. R. (2001). "A genetic algorithm for optimizing off-farm irrigation scheduling." *J. Hydroinformatics,* International Assoc. of Hydr. Research, 3(1), 11-22.

Peng, W., and Jia, R. (2004). "Improved genetic algorithms for optimal design of drainage systems." *8th International Conference in Control, Automation, Robotics, and Vision.* Kunming, China. pp. 227-231.

Pezzinga, G., Gueli, R., Reis, L. F., Porto, R. M., and Chaudhry F. H. (1999). "Optimal location of control valves in pipe networks by genetic algorithm. Discussion." *J. Water Resour. Planning and Manage.*, 125(1), 65-66.

Simpson, A. R., and Wu, Z. Y. (2001). "Competent genetic-evolutionary optimization of water distribution systems." *J. Computing Engrg.*, 15(2), 89-101.

Wilson, S. W. (1995). "Classifier fitness based on accuracy." *Evolutionary Computation,* 3(2), 149-175.

Zheng, Y. W. (2006). "Optimal calibration method for water distribution water quality model." *J. Environmental Science and Health Part A,* 41, 1363-1378.

Acknowledgments

Support from the Utah Agricultural Experiment Station is gratefully acknowledged.

Symbols

The following symbols are used in this paper:

cl.a	classifier action
cl.F	classifier fitness
cl.p	classifier prediction
C_m	number of times missing the target depth
C_o	coefficient for water depth deviation from the target
C_{ps}	coefficient for stability penalty
C_r	number of consecutive repetitions when reaching the target depth
C_{si}	coefficient for stability reward
k	calibration exponent for penalty for leaving the target depth
m	exponent for stability rewards
[M]	match set
n	exponent for water level deviation from the target depth

DOI: 10.1061/(ASCE)IR.1943-4774.0000268

$P(a)$	system prediction
$P(a)_{max}$	classifier's maximum prediction for an action
P_b	base penalty
P_{SI}	stability penalty
P_{tl}	penalty for leaving the target depth
R	desired reward
r	previous reward
R_b	base reward
RSI	reward for stable rules
R_{tl}	reward for reaching the target depth
$\lvert SI\rvert$	absolute value of the stability index
u	exponent for stability penalty
α	parameter for accuracy
β	learning rate
ε	prediction error
ε_0	prediction error threshold
γ	discount factor
κ	accuracy values for classifiers
μ	probability of mutation
ν	exponent for accuracy
χ	probability of crossover
κ'	relative accuracy