Water Reservoir Control with Data Mining

Florian T. Bessler[1]; Dragan A. Savic[2]; and Godfrey A. Walters[3]

Abstract: This paper describes the development of a general operating policy for a water supply system using the methodology of data mining. To define an operating policy using this approach, both a single-reservoir and a multireservoir water system were modeled and optimized for a set of historical inflows. These optimization results defined the best possible performance for the systems with historical hindsight, and were used as input for the data mining process. The data mining algorithm then generated the set of control rules that gave the best historical operating policy. The data mining tool used in this work is based on the induction tree technique, C5.0, reported by Quinlan in 1993. However, the process of reservoir control rule extraction is not straightforward and requires several data preparation steps to enhance the performance of the data mining algorithm. To demonstrate the effectiveness of the rules developed through data mining, simulation runs of the system were performed. The results of these simulations were compared with simulation results using operating policies derived from linear regression. Another comparison between operating rules derived using different methodologies was performed for the multireservoir system where, in addition to data mining and regression-based rules, there were rules available from the U.K. Environment Agency (South West). The paper shows that "data-mined" rules come closest to the optimization results.

DOI: 10.1061/(ASCE)0733-9496(2003)129:1(26)

CE Database keywords: Reservoir operation; Data collection; Inflow; Simulation.

Introduction

Water is one of the most important natural resources, and water managers around the world are under increasing pressure to operate their systems more efficiently. The problem of water reservoir operation, be it in single-reservoir or in multireservoir systems, has been studied in the past with different optimization and simulation techniques. Various optimization models and mathematical algorithms have been developed to assist the complex decision-making process of water reservoir operation (see, e.g., Loucks and Sigvaldason 1982; Young 1967; Georgakakos 1989; and Kuczera 1989). These algorithms, such as linear programming (LP) and stochastic dynamic programming, can be used even for the real-time management of water reservoirs in multiperiod optimization models. However, as described in Oliveira and Loucks (1997) and Nalbantis and Koutsoyiannis (1997), the use of real-time optimization models still plays a minor role in real-life applications and prediction of reservoir releases. Water reservoirs are still operated mainly on the basis of predefined rules, and the primary tools in management remain simulation models in which these predefined rules are tested, evaluated, and improved.

Control (or operating) rules are usually given in the form of equations, charts or look-up tables that specify the amount to be released for various purposes as a function of system state and parameters (ReVelle 1999). Control rules have been used in the United Kingdom for more than 50 years to reduce operating costs by controlling the overdrawing and pumped refill of reservoirs. However, for over 25 years some water companies within the UK have been integrating their sources into resource zones, so there has been a need to produce conjunctive control rules applicable to a whole system (Pearson and Walsh 1982). These predefined rules usually link together the main system variables, such as inflow to each reservoir within the system, reservoir storage, releases, time of year, spillage, leakage, evaporation, and expected future conditions. However, before such a system can be operated, these rules have to be defined. It can be seen from the large number of articles about reservoir rule generation (Wurbs 1985; Yeh 1985; ReVelle 1999) that this subject has received considerable attention. Lately, new methods like genetic algorithms have also proved to be successful (Oliveira and Loucks 1997; Wardlaw and Sharif 1999). Fuzzy programming is another approach which has been applied and shows good results as long as the system is not too complex and only few variables are used (Russell and Campbell 1996; Ponnambalam et al. 2001). In addition to these computationally based approaches, the expert knowledge of the local manager is usually incorporated for safe and reliable operation.

This paper investigates a new approach to the identification of multireservoir control rules based on data mining (Bessler 1998). Data mining appears under a multitude of names, which includes knowledge discovery in databases, data or information harvesting, data archaeology, functional dependency analysis, knowledge extraction, and data pattern analysis. In addition, there exist a large number of definitions for this group of methods. The term data mining is used for both the whole process of knowledge discov-

[1]Research Assistant, School of Engineering and Computer Science, Centre for Water Systems, Univ. of Exeter, Harrison Bldg., North Park Rd., Exeter EX4 4QF, U.K.

[2]Professor, School of Engineering and Computer Science, Centre for Water Systems, Univ. of Exeter, Harrison Bldg., North Park Rd., Exeter EX4 4QF, U.K. (corresponding author). E-mail: d.savic@exeter.ac.uk

[3]Professor, School of Engineering and Computer Science, Centre for Water Systems, Univ. of Exeter, Harrison Bldg., North Park Rd., Exeter EX4 4QF, U.K.

Note. Discussion open until June 1, 2003. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on September 28, 2001; approved on February 22, 2002. This paper is part of the ***Journal of Water Resources Planning and Management***, Vol. 129, No. 1, January 1, 2003. ©ASCE, ISSN 0733-9496/2003/1-26–34/$18.00.

ery and also for the specific algorithms which are used to achieve this. Of several related definitions of data mining one that is most appropriate for real-world applications is given by Fayyad et al. (1996): *Data mining is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*

In other words, data mining is the search for relationships and global patterns that exist among parameters, but are hidden among the data. The data mining technique used here to extract reservoir control rules is the induction tree technique (C5.0), as described in Quinlan (1993).

The use of the decision tree technique C5.0 for extracting reservoir control rules is demonstrated on a single-reservoir system first. The Avon reservoir is part of the Roadford water supply system, in the South West of England. Data records for the system were obtained from the U.K. Environment Agency (EA), the environmental regulator which ensures that water utilities (as well as all other industries) do not harm the environment in the course of their business activities. To allow a comparison of rules identified using data mining an alternative set of rules was developed using linear regression and both rule sets were tested in simulation. A larger, multireservoir system was analyzed next where the data mining rules were compared to the rules obtained via linear regression and to the rules supplied by the Environment Agency.

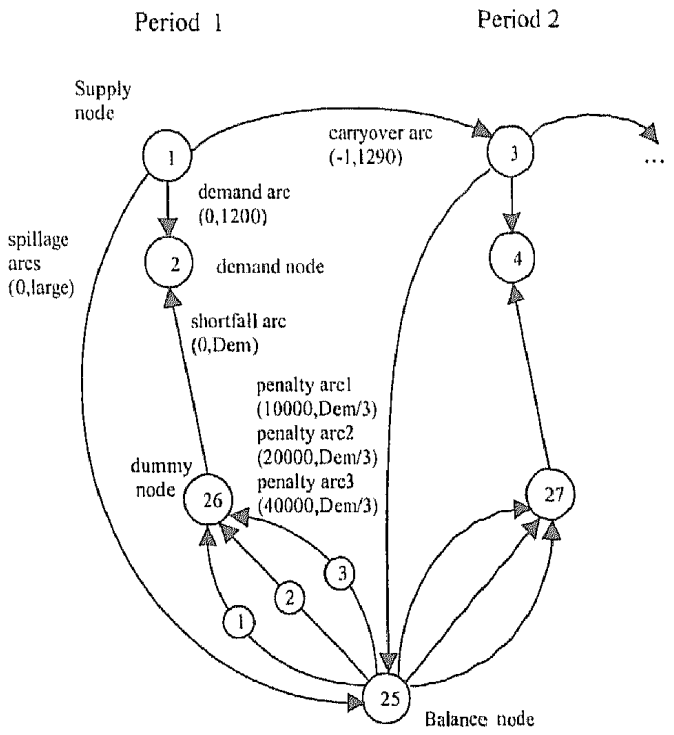


Fig. 1. Example of linear network model with one supply node and one demand node copied over two time steps

Model Design and Optimization

The model for the Avon reservoir had to be created first, and this was done using a linear network modeling approach. Linear programming (LP) could then be used to solve the optimization problem (Kuczera 1989; Karim 1997). The approach assumes that a water supply system can be modeled as a linear network, given linear constraints and linear costs. The linear network derived using this approach represents a single-reservoir system and consists of one *supply node* representing the reservoir and a number of *demand nodes*, representing the demand areas. The nodes are connected by arcs, representing the pipework between the reservoir and the demand areas. Arcs are capacitated to reflect their minimum and maximum capacities for water transfer.

To introduce time into the system, the time variable is discretized and the initial network is copied for each time step (Fig. 1). The time steps can represent any desired time period: days, weeks, months or years, depending on the data available and the behavior to be studied. Monthly data were used here for the design of control rules. In order to satisfy the continuity equation for the reservoir, the sum of the supply over all nodes must be zero, as given by the following equation:

$$\sum_{i \in N} s_i = 0 \tag{1}$$

where s_i = supply to Node i and N = number of nodes.

However, inflows and demands associated with the reservoir are usually not balanced (matched) during all time steps. Therefore, the model needs a *balance node*, which adds or subtracts the difference between inflows and demands as necessary. In each time step, the reservoir is connected to the balance node and to itself within the next time step. Fig. 1 shows the linear network model of a single reservoir system over two time steps. To meet any shortfall in demand in any time period, the balance node is connected to all demand nodes through dummy nodes. The top node always represents the reservoir for any period, e.g., Node 1 represents the reservoir in period 1 while Node 3 represents the reservoir in period 2, etc. Nodes 2 and 4 represent traditional demand, e.g., domestic, irrigation, or industrial. Node 25 of the network is the balance node. It balances the difference between supply and demand for the complete system, and is connected to all demand nodes in each time period through *shortfall arcs*. The shortfall arcs are provided to cater to any demand not met by the reservoir (i.e., to make the operation feasible), but at a much higher cost. To represent this cost, the balance node is connected to a dummy demand node by several parallel penalty arcs (three in this case), each arc having a capacity of a third of the total demand in that period. Each of the penalty arcs is assigned a different cost per unit flow to represent a nonlinear cost penalty for unsatisfied demands. The resulting cost function is piecewise linear convex, as shown in Fig. 2. The *storage* in the network is represented by the *carryover arc*, which leads from the reservoir in the current period to the same reservoir in the following time period. The *spillage* arcs point from the reservoir to the balance node. Each arc is assigned a cost per unit flow (zero in most cases) and a capacity. All costs are artificial and are used as design parameters to manage the trade-off between the consistency of monthly supplies and the total cumulative deficit in the model. For the assignment of cost values to the arcs, the *RELAX* algorithm (which will be used for optimization) requires the values to be integers (Bertsekas 1991). A cost is assigned to the penalty

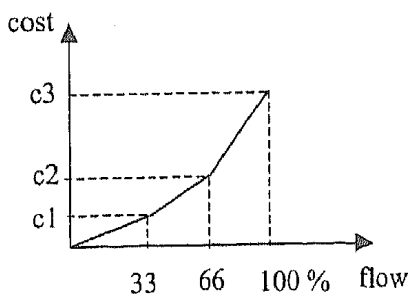


Fig. 2. Linear convex cost penalty function

arcs of several orders of magnitude higher than the costs for the other arcs in order to minimize the mismatch between demands and actual supply. For a detailed description of the cost system refer also to Kuczera (1989).

Usually, the historical data available for a reservoir are its monthly inflows, the storage in the previous time periods, and its releases to the demand areas. The historical data available for the demand areas is their monthly demands and the amounts of water they have been supplied with. Thus, any deficit that might have occurred can be identified. To complete the linear network model, the boundary conditions have to be taken into account. This is also important for the consistency of the mathematical model. The representation chosen here is to add the initial storage of the reservoir to the inflow for the first period and to connect the last carryover arc (in the last period) to the balance node.

Mathematical Problem Formulation

The problem of optimizing the flow within a linear network at minimum cost, including consistent supply to the demand zones, can be formulated as the so-called "minimum cost flow problem" (Bertsekas 1991):
Objective Function:

$$\text{Minimize} \sum_{t=1}^{np} \sum_{(i,j)\in A_t} c_{ij}x_{ij} \tag{2}$$

where, np=number of time periods; c_{ij}=unit cost of arc (i,j); x_{ij}=flow in arc (i,j); and A_t=all arcs of one time period, subject to the constraints

$$\sum_{\{j|(i,j)\in A\}} x_{ij} - \sum_{\{j|(j,i)\in A\}} x_{ji} = s_i, \quad \forall i \in N \tag{3}$$

$$\text{and} \quad 0 \leq x_{ij} \leq \text{cap}_{ij}, \quad \forall (i,j) \in A \tag{4}$$

where, cap_{ij}=capacity or the upper flow bound of arc (i,j); $[0,\text{cap}_{ij}]$=feasible flow range of arc (i,j); and s_i=supply to Node i.

To solve this optimization problem, many linear programming algorithms are available and the one used in this research is the *RELAX* algorithm (Bertsekas 1991). The necessary programming work was carried out in *FORTRAN 77*.

Data Mining and Knowledge Discovery

Water utilities currently recognize that in addition to making data available across a company, it is equally important to be able to efficiently extract information from data, i.e., to have procedures for identifying logical, nontrivial, useful, and ultimately understandable patterns in data (Savic and Walters 1999). Knowledge discovery can be defined as the process of identifying these useful patterns in data. The core of the process is data mining, the automated analysis of large or complex data sets in order to discover *significant patterns or trends* that would otherwise go unrecognized. To make things more difficult, data mining appears under a multitude of names, which includes knowledge discovery in databases, data or information harvesting, data archaeology, functional dependency analysis, knowledge extraction, and data pattern analysis. The most commonly applied technique develops models that group data according to preclassified examples. One of these techniques, decision tree induction, discovers a set of rules that can be applied to new (unclassified) data to predict

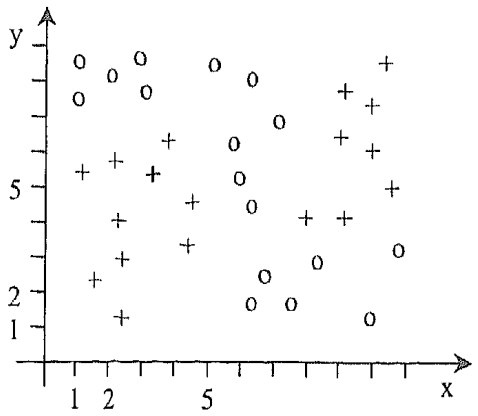


Fig. 3. Data points in two-dimensional state space

which data records will have a given outcome. For example, when dealing with reservoir operation, one would like to predict, based on the values of other attributes (reservoir storage, season, and inflow in the previous month, etc.) what should be the release from the reservoir. This is particularly important for real-time operations.

In this study, the focus is on decision tree data-mining algorithms, in particular the C5.0 algorithm (Quinlan 1993). To apply this algorithm, there are some requirements for the data sets that are to be analyzed. The information about each case in the data set is defined by *attributes* and each case is assigned to belong to a *class*.

- *Attributes*: All information about cases has to be presentable in the form of attributes. Attributes can be either numerical or represent a nonnumerical, discrete category. Numerical values can be either discrete or continuous. The attributes must be the same for all cases, but data mining is able to handle cases with attributes whose values are unknown.
- *Classes*: Each case in the data set must be assigned to a predefined and discrete class. The classes must be defined sharply, so a case either belongs to a class or it does not. For good results, the number of classes should be small (around 2–10), whereas the number of sample data cases should be relatively large (around 100–10,000, sometimes more).

The knowledge discovery process is described in general terms by the following stages: data selection, preprocessing, transformation, data mining, and interpretation, (Fayyad et al. 1996). Naturally, the quality of the data mining process requires the domain expert to perform good selection and preprocessing of the data, the earlier stages of the knowledge discovery process. An important step to ensure the success of the data mining application is to identify the preprocessing steps for the data set. One of the problems that can arise is that if the class descriptor is originally continuous, it has to be divided manually into fuzzy, but discrete classes (e.g., all, much, half, few, none). Analogous methods for handling continuous classes can be found in Breiman et al. (1984). In the actual development of the decision tree, for the case of the C5.0 algorithm, all continuous attributes are actually discretized for the internal handling (Quinlan 1993). Also, the data represented has to be diverse. If 95% of cases are of one particular class, the algorithm will predict that class as a default result for all cases, with the resulting error of 5% misclassifications deemed quite acceptable.

To explain and visualize the functionality of a decision tree algorithm, a first simple example is given here. Fig. 3 illustrates this first example, where only two variables are considered and they form a state space, in which the decision tree algorithm is

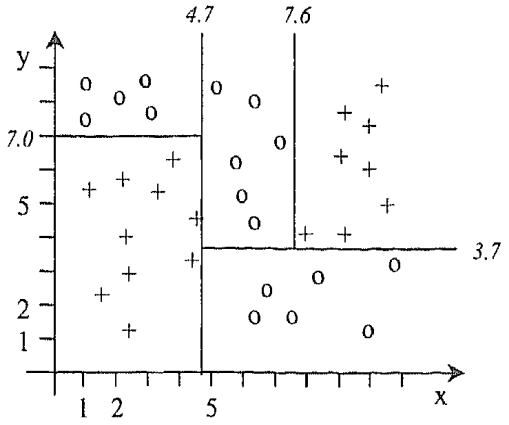


Fig. 4. State space divided by decision tree algorithm

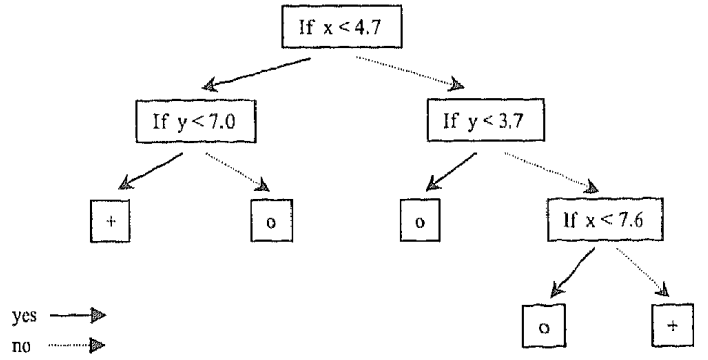


Fig. 5. Decision tree for this example of dividing two-dimensional state space

trying to find a pattern. The data points are also only considered to belong to exactly one of two classes, the easiest form of a classification differentiation. Most decision tree algorithms (including C5.0) are only able to search for patterns by making divisions that are parallel to the axes. This is done to limit the computational effort in deciding which split is the best split to make, which is done by the brute force method of considering every possible split and continuing to divide state space as long as any further split will improve the prediction model, until all cases are defined. Without explaining the detailed method, it becomes clearly visible that the brute force method requires a high computational effort, especially when multidimensional state space is considered. Fig. 4 then shows how this two-dimensional state space could be divided in this example. In this case, it can be seen that the state space was divided into five different regions. The data are clearly distinguishable and no splits accepting misclassifications are made. The corresponding decision tree then would appear as in Fig. 5. The decision tree consists of either a *leaf* (+ or o), indicating a value of the classification variable, or of a *decision node* (If $x<4.7$), specifying a test to be carried out. For each outcome of the test there is assigned either a leaf or a decision node again, until all *branches* end in leaves of the tree. Simply put, a decision tree is a series of "if"..."then" decisions, which divide the n-dimensional state space. Without explaining the detailed function of the construction of the decision trees as described in Quinlan (1993) and Breiman et al. (1984), their main idea is to generate an initial decision tree by a recursive partitioning method. This continues to subdivide the set of training cases until each subset in the partition contains cases of a single class or no further split offers any improvement. Such a tree will usually overfit the data. Thus, different *pruning* methods are applied to reduce the size of the tree, to produce a tree with a smaller error rate. The most important pruning method is to predict an error rate for each subtree and compare it to the error prediction if the subtree would be replaced by a leaf or the most frequently used branch. If the predicted error rate is better than the one of the detailed tree, it is replaced and thus a less complex and more comprehensible tree is received. In this case, no pruning was necessary as the data points are clearly distinguishable so no generalization had to be made. The C5.0 algorithm can also handle missing data and when a value is not known at a node of the decision tree, it explores all possible outcomes and combines the resulting classifications arithmetically and chooses the class with the highest probability as "the" predicted class.

A decision tree then also represents a set of control rules, with the characteristic that the rule set is structured such, that only one rule is activated (fires) for any given and complete case. There are methods to generate more general rule sets from decision trees, but for the simulations here only complete decision trees were used. One of the advantages of decision trees as data mining algorithms is that such a set of rules can be derived, and the validity of these rules can be tested against other examples and domain experts can decide on the quality of the rules. This stands contrary to other data mining methods, such as neural networks, which act as a "black box" and it cannot be derived how the prediction is achieved there.

Characteristics of Data Mining for a Single-Reservoir System

Taking the linear network modeling example and solving the corresponding minimum cost flow problem results in the optimal prediction of releases from the reservoir. For the Roadford system there were 35 years of data available. These data were split into two halves, the first half was used for the development of the rules and the second half was used for the testing of the rules in a simulation. The data were examined on a monthly basis. The historical data have been altered here, as normally the demand would always be met, which would lead to the simple data mining prediction of always releasing the water required. So instead, demands were increased and the storage capacity of the reservoir was decreased to create artificial shortfalls of water in summertime when inflow is low and demands are high. Since failure to meet demands involves a large penalty cost which increases sharply at certain critical values (in this example at 66 and 33% of the demand being met), the optimization tends to reduce some early releases, conserving water for more grave water shortage problems later on. The demands, although different for each month, are constant over the years, that is, every year will find the same demand in the same month. The initial state of the system is known, starting in the very first time step (January, year 1) with a full reservoir.

For the choice of class, it is apparent that the variable "release" cannot be chosen directly, as it is of a continuous nature and this is an impossible choice for a class. Therefore, the class is discretized into 10% steps, corresponding to the percentage of the demand met. This can be seen in Table 1, which describes a typical year for the Avon water reservoir under optimized operation.

This data set shows typical behavior for a water reservoir operating with deficiencies. The deficit in the summer months is spread over all of the summer, instead of dropping to extreme levels of demand shortfall. Also, full use is made of the reservoir storage, which is reduced to zero by the end of month 10. That is,

Table 1. Main Figures of Typical Year of Water Supply System

Month	Inflow ML[a]	Change in inflow ML[a]	Storage ML[a]	Sum ML[a]	Demand ML[a]	Release ML[a]	Demand Met %	Class
1	1074	−1256	1290	2364	1004	1004	100 →	10
2	1468	394	1290	2758	914	914	100 →	10
3	2512	1044	1290	3802	1045	1045	100 →	10
4	1342	−1170	1290	2632	1122	1122	100 →	10
5	1012	−330	1290	2302	1156	1012	87 →	9
6	326	−686	1290	1616	1243	415	33 →	4
7	273	−53	1201	1474	1516	843	55 →	6
8	712	439	631	1343	1540	1027	66 →	7
9	438	−274	316	754	1130	754	66 →	7
10	1266	828	0	1266	1071	1071	100 →	10
11	1109	−157	195	1304	998	998	100 →	10
12	1510	401	306	1816	1052	1052	100 →	10

[a]ML = megaliters.

the storage is at the lowest operating level, as the reservoir is allowed to operate from minimum pool level to its maximum storage capacity.

The data mining algorithm was then applied to this data set, in this case using the C5.0 algorithm by Quinlan (1993) with the See5 software. The five attributes used for decision tree generation were: Inflow, Storage, Sum (=Inflow+Storage), Demand, and Month. The attribute Sum is included explicitly because decision tree algorithms are unable to take linear combinations of attributes into account and rule defining splits are always made parallel to the axes of state space. Trials, including the monthly change in inflow as an additional attribute (in anticipation of an influence of the first derivative of inflow), showed no improvement in the resulting decision tree. Fig. 6 then, shows the best decision tree developed for the case of the single-reservoir water system. The top node split reflects the fact that if the inflow is large enough, usually the demand can be supplied fully, unless it is exceptionally high (one summer month) when the release is reduced to 80% of the demand. Should the inflow fall below 934 ML the tree differs for each Month, supplying water more generously in the winter months and deriving more detailed branches for the summer months. Also note, that class 4 is the lowest prediction result which occurs.

The quality of the decision tree can be analyzed using a confusion matrix, on which are shown, for every case, the class to which a case actually belongs and the class to which it was assigned by the data mining prediction. As can be seen from the confusion matrices in Tables 2 and 3, the training cases are classified better than the test cases. Nevertheless, even test cases are classified correctly in 84% of the cases, especially where there are enough cases provided. Furthermore, if a misclassification occurs, the results tend to be classified rather as a neighboring class than as a remote class. On these grounds, a simulation tool was programmed which incorporates the decision tree.

Simulation and Comparison to Linear Regression

For further comparison, linear regression was implemented as a different prediction method, to provide an alternative set of control rules. The same set of variables was used for linear regression as for data mining. But because the variable "Sum" (total available water) is a linear combination of "Inflow" and "Storage" it is sufficient to use the variables Inflow and Sum here. Furthermore, the variable "Demand" is a constant in each month which is why it is not used as a regression variable, but the obtained results of the parameter vector are only valid for the specific value of that constant in each month. Thus, the independent variables for linear regression are Inflow and Sum determined in each month separately and the dependent variable is "Release"

$$\text{Release} = a \cdot \text{Inflow} + b \cdot \text{Sum} \tag{5}$$

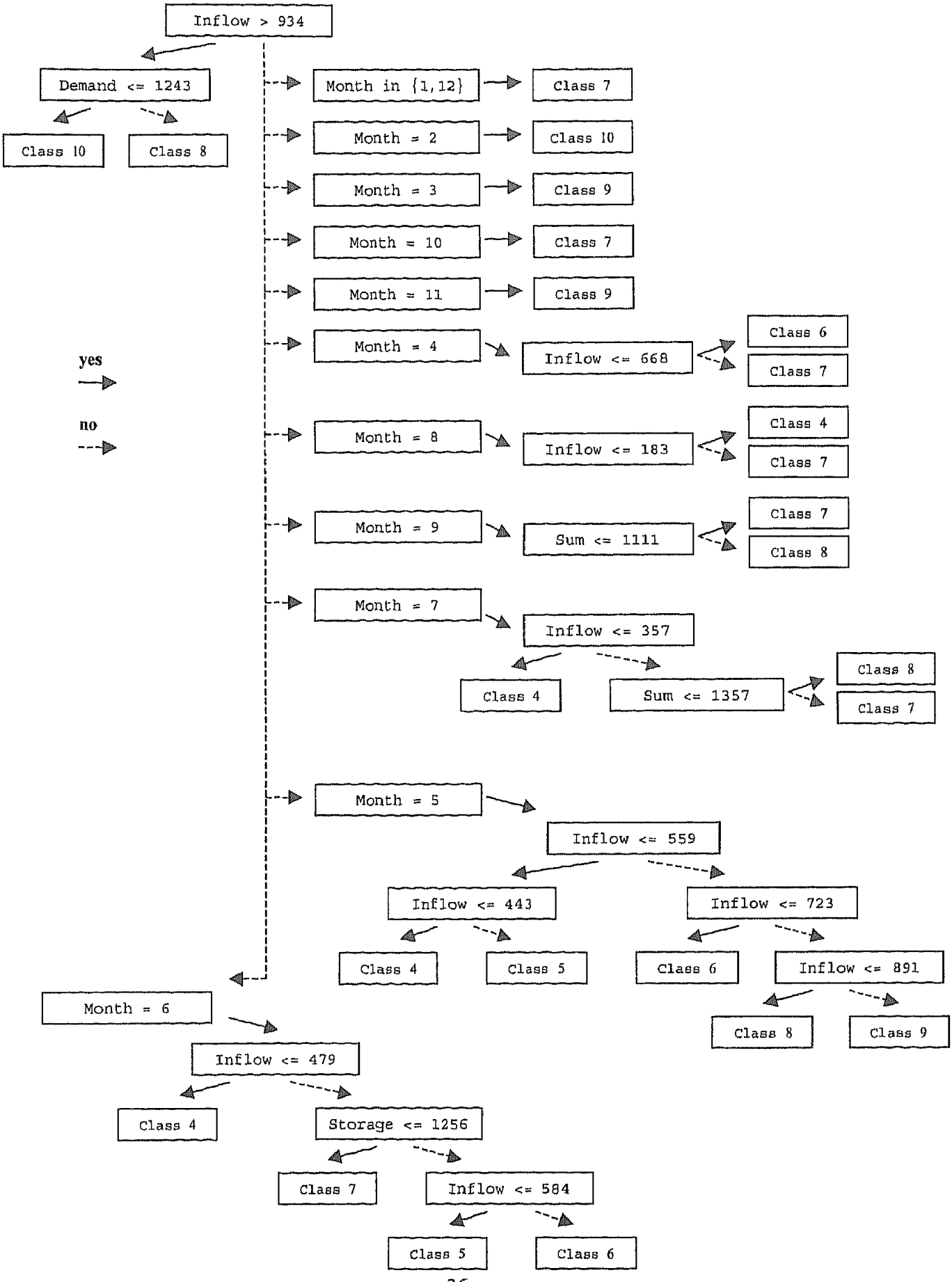
As a third method, control rules were designed in the way the Environment Agency (EA) designs them. The rules from the Environment Agency are purely storage focused and predict a release from the reservoir according to its current storage and the time of year. This is the way classic reservoir control curves are designed, although it remains questionable if the EA would actually apply such a rule set to the altered problem that is analyzed here. The primary effect here is to show that the EA rules are not prepared to deal with the management of the trade-off between the consistency of monthly supplies and the total cumulative deficit, but only manage reservoir storage in well balanced systems.

For all methods a simulation was carried out and the results of the simulation can be seen in Fig. 7. Note that in the simulation, predictions are tied to reality and only feasible actions are carried out. So if an algorithm predicts a release of water which is not there, or which exceeds capacity limits, such a request is carried out only up to the physical limitations.

Fig. 7 shows the distribution during the drought periods and the minimum levels of supply that are reached. The demands were increased quite heavily to train the system for extreme situations. In reality, shortfalls of this extent are very rare. In general, the results show that the performance of data mining is closer to the optimal solution than the performance of linear regression or the EA control rules. Furthermore, it can be seen that linear regression sometimes produces unnecessary deficits during the winter months, when supplying the full demand should create no difficulties. One of the problems of the data mining solution can be seen in its tendency to drop to its lowest levels towards the end of the summer months. This is caused by occasional misclassifications, which reduce the storage in the water reservoir even further. Nevertheless, the predictions of data mining clearly lead to a better simulation result than either of the other methodologies.

Multireservoir System

The above methodology of modeling, optimization, extracting rules with data mining and linear regression, and finally simula-



36

Fig. 6. Example of decision tree for single-reservoir system

Table 2. Confusion Matrix for Classification of Training Cases

	Classified As							
True class	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(a): class 3								
(b): class 4		16		1				
(c): class 5		1	4		2			
(d): class 6		4		6	1			
(e): class 7		1		2	26			
(f): class 8					1	10		1
(g): class 9		1	1		1		6	1
(h): class 10			1		2	1		110

Table 3. Confusion Matrix for Classification of Test Cases

	Classified As							
True class	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(a): class 3								
(b): class 4		8	1					
(c): class 5		1	3	2	3			
(d): class 6				7				
(e): class 7		3		1	31	3		
(f): class 8				1	6	7		
(g): class 9				1	3			1
(h): class 10		1		1	1	3		112

tion was also carried out for a complex multireservoir system. Such a system has special characteristics, as shown in Fig. 8, which gives a complete view of the system. Although each reservoir is assigned its main supply area, water can be transferred to different demand nodes as well. The network shown in Fig. 8 represents again one time step of the system, where each of the four reservoirs is connected to one or more demand nodes, the balance node, and each reservoir to itself in the next time step. The reservoirs are allowed again to operate from minimum pool level to their maximum storage capacity, and the initial state of the system is known. The reservoir storage is specified in its carryover arc, and regular arcs are capacitated according to their maximum flow. The balance node is connected through shortfall arcs again to each demand node, involving the penalty cost system as before. In addition, the demand node of the fisheries bank is given a higher priority, because a shortage there is more serious (the fish die) than if some households are not supplied fully. For the links from each reservoir to each demand node, the flows were predicted with data mining and linear regression and then simulated with the added physical constraints of capacity and available water. In the simulation with the EA control rules though, they only predict the release from each reservoir and supply the associated demand nodes according to a priority system. That is, each reservoir supplies its directly associated demand node first and then continues on to shared demand nodes.

For the multireservoir system the original training data were modified again, so that shortfalls occur, in order to have a diverse data set. As before, the data set of 35 years of supplied data was split into two halves, the first half used for training and the second half for testing and simulation purposes. As the results for all individual reservoirs are quite extensive, just the final results are presented here and the main characteristics of the system are discussed.

The results of the simulation show again that data mining performs better than linear regression or the EA control rules. Table 4 shows the cumulative deficit for each reservoir for a simulation over 204 test cases, that is 17 years. The main problem that still remained for data mining was the insufficiently diverse data, which led to few classifications in the lower classes. So class 10 is classified misleadingly too often with the data mining developed rule set, which then produces occasional steep drops to 0% coverage when a smaller reservoir runs out of water during the summer months, where it should have cut down consistently instead and guaranteed a minimum supply level. Another effect occurred from the dependency of the reservoirs on each other. A well supplied reservoir would not cut down soon enough to a lower demand and supply the urgent need of another demand node which was getting very low on supply, when either of the prediction methods were used. Particularly the EA rules do not take that effect into account at all, as they decide on the distribution simply by a priority system. Linear regression in addition produces again some of the unnecessary shortfalls in coverage during the winter months, when the system could, in fact, be supplied fully. So, in general, when the results in Table 4 are viewed, it is easy to identify data mining as the best of the tested prediction methods for the generation of predefined control rules for the operation of

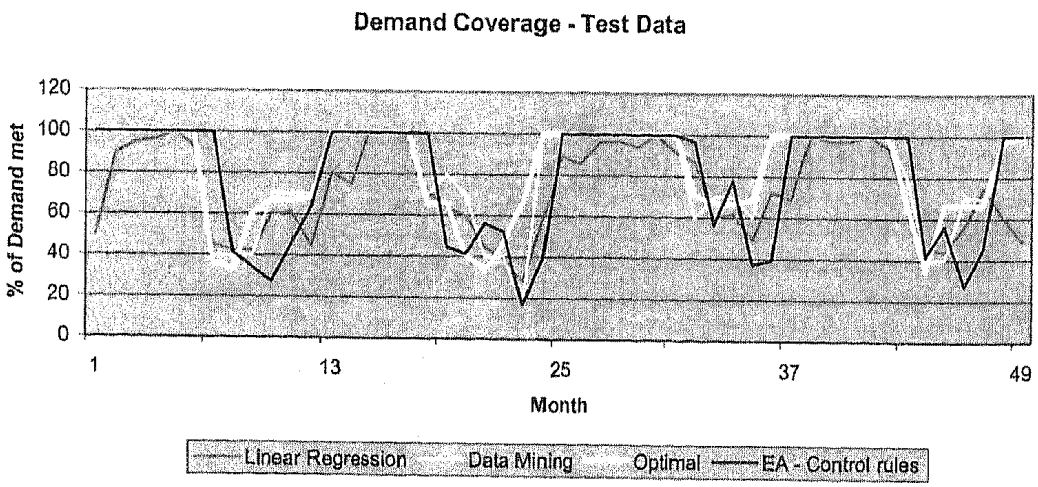


Fig. 7. Comparison of simulation results

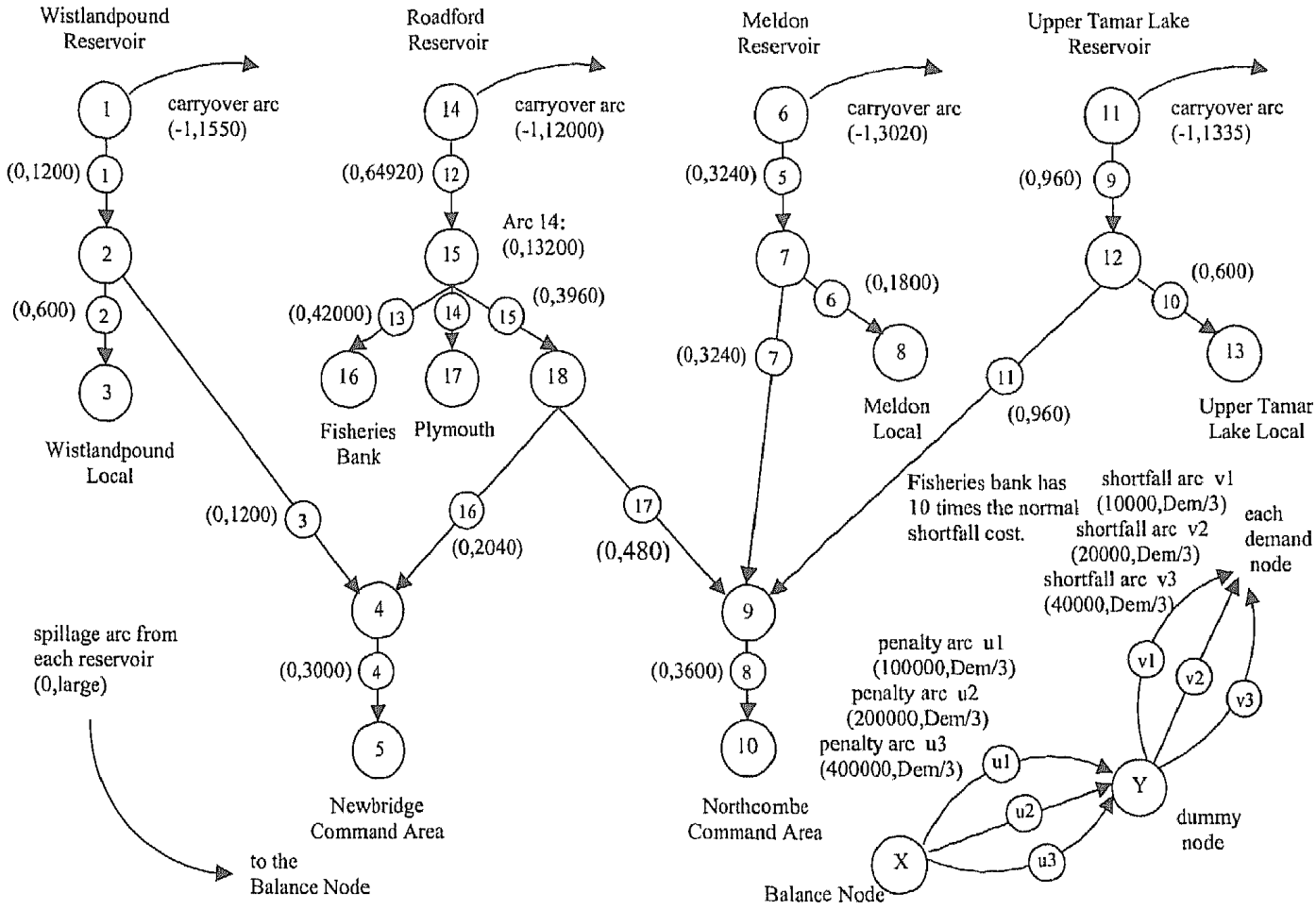


Fig. 8. Part of Roadford network as model for multireservoir system

this water system. Thus data mining certainly proves a valuable method and further research in this area would be of interest.

Conclusions and Future Outlook

In this paper general operating policies for a single-reservoir and a multireservoir water system were developed. These were the Avon water reservoir and part of the Roadford water system in the South West of England. The methodology of network modeling, optimization with linear programming, rule extraction with data mining and linear regression, flow prediction, simulation, and analysis was applied. The generation of predefined rules with data mining proved practicable and provided good results for the operation of a single-reservoir system. The advantage of using decision tree algorithms is that rule sets are produced which are auditable by domain experts, who can further improve their quality. For both, the single and multiple systems, the simulation with the data mining derived rule sets alone provided better results than the simulation with rule sets developed by the classic statistical method of linear regression or simulation with the existing EA control rules. In all cases, there remain some difficulties with the rather complex multireservoir system. However, in the fast moving software development industry there are continuous changes and improvements happening to the method of data mining. It is likely that the data mining process might be improved with different methods generating better decision trees. Ideally, though, it would be better not to rely entirely on data mining generated control rules, but to improve and combine the

Table 4. Cumulative Deficits in Megaliters for All Demand Zones After Simulation

Demand area	Node	EA-control rules	Lin. reg.	Data mining	Optimal
Wistlandpound	3	1125	10,000	5,000	7,100
Newbridge	5	56,092	29,602	29,823	19,118
Meldon Local	8	3,700	47,975	22,556	23,790
Northcombe	10	137,260	52,297	37,994	28,875
U. Tamar Lake	13	0	12,476	5,624	6,230
Fisheries Bank	16	1,499	5,368	2,940	0
Plymouth	17	37,648	69,029	67,397	46,108
		...	...	...	...
Combined		237,324	226,747	171,334	131,221

rule sets incorporating the knowledge and experience of reservoir operation managers in an advance towards an optimal expert system.

References

Bertsekas, D. P. (1991). *Linear network optimization: Algorithms and codes*, MIT Press, Cambridge, Mass.

Bessler, F. T. (1998). "Linear network optimisation and identification of control rules to develop a general operating policy for a water supply system." Masters thesis, Univ. of Exeter, U.K.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*, Wadsworth, Belmont, Calif.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). *Advances in knowledge discovery and data mining*, AAAI Press and MIT Press, Chap. 1, 1–34.

Georgakakos, A. P. (1989). "Extended linear quadratic Gaussian control: Further extensions." *Water Resour. Res.*, 25(2), 191–201.

Karim, K. (1997). "An improved approach to the development of operating policies for multiple reservoir systems." PhD thesis, Central Queensland Univ., Rockhampton, Queensland, Australia.

Kuczera, G. (1989). "Fast multireservoir multiperiod linear programming models." *Water Resour. Res.*, 25(2), 169–176.

Loucks, D. P., and Sigvaldason, O. T. (1982). "Multiple-reservoir operation in North America." *Operation of Multiple Reservoir Systems, ILASA Collab. Proc. Ser. CP-82-53*, Z. Kaczmarek and J. Kinder, eds., Laxenburg, Austria.

Nalbantis, I., and Koutsoyiannis, D. (1997). "A parametric rule for planning and management of multi-reservoir systems." *Water Resour. Res.*, 33(9), 2165–2177.

Oliveira, R., and Loucks, D. P. (1997). "Operating rules for multireservoir systems." *Water Resour. Res.*, 33(4), 839–852.

Pearson, D., and Walsh, P. D. (1982). "The derivation and use of control curves for the regional allocation of water resources." *Optimal Allocation of Water Resources, Proc., Exeter Symposium*, IAHS Publ. No. 135.

Ponnambalam, K., Monsavi, S. J., and Karray, F. (2001). "Regulation of Great Lakes reservoirs by a neuro-fuzzy optimisation model." *Int. J. Comput. Anticipatory Syst.*, 4, 272–285.

Quinlan, J. R. (1993). *C4.5, Programs for machine learning*, Morgan Kaufmann, San Mateo, Calif.

ReVelle, C. (1999). *Optimizing reservoir resources*, Wiley, New York.

Russell, S. O., and Campbell, P. E. (1996). "Reservoir operating rules with fuzzy programming." *J. Water Resour. Plan. Manage.*, 122(3), 165–170.

Savic, D. A., and Walters, G. A. (1999). "Hydroinformatics, Data Mining and Maintenance of UK Water Networks." *J. Anti-Corros. Methods Mater.*, 46(6), 415–425.

Wardlaw, R., and Sharif, M. (1999). "Evaluation of genetic algorithms for optimal reservoir system operation." *J. Water Resour. Plan. Manage.*, 125(1), 25–33.

Wurbs, R. A. (1996). *Modelling and analysis of reservoir systems operations*, Prentice Hall, Englewood Cliffs, N.J.

Yeh, W. W.-G. (1985). "Reservoir management and operation models: A state-of-the-art review." *Water Resour. Res.*, 21(12), 1797–1818.

Young, G. K. (1967). "Finding reservoir operating rules." *J. Hydraul. Div., Am. Soc. Civ. Eng.*, 93(6), 297–321.