Artificial Intelligence techniques: An introduction to their use for modelling environmental systems

3 authors, including:

Serena H. Hamilton
Australian National University & Edith Cowan University
23 PUBLICATIONS   **1,772** CITATIONS

SEE PROFILE

A.J. Jakeman
Australian National University
377 PUBLICATIONS   **13,156** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

An integrated hydro-ecological and socio-economic model of environmental flow delivery to wetlands: A case study of Macquarie Marshes in Australia
View project

General Hydrology View project

Artificial Intelligence techniques: An introduction to their use for modelling environmental systems

Serena H. Chen, Anthony J. Jakeman [*], John P. Norton

*Integrated Catchment Assessment and Management (iCAM), Fenner School of Environment and Society,
Building 48a, The Australian National University, Canberra, ACT 0200, Australia*

Abstract

Knowledge-based or Artificial Intelligence techniques are used increasingly as alternatives to more classical techniques to model environmental systems. We review some of them and their environmental applicability, with examples and a reference list. The techniques covered are case-based reasoning, rule-based systems, artificial neural networks, fuzzy models, genetic algorithms, cellular automata, multi-agent systems, swarm intelligence, reinforcement learning and hybrid systems.
© 2008 IMACS. Published by Elsevier B.V. All rights reserved.

Keywords: Case-based reasoning; Cellular automata; Multi-agent; Environmental modelling

1. Introduction

Use of Artificial Intelligence (AI) in environmental modelling has increased with recognition of its potential. AI mimics human perception, learning and reasoning to solve complex problems. This paper describes a range of AI techniques: case-based reasoning, rule-based systems, artificial neural networks, genetic algorithms, cellular automata, fuzzy models, multi-agent systems, swarm intelligence, reinforcement learning and hybrid systems. Other arguably AI techniques such as Bayesian networks and data mining [21,148] are not discussed.

2. Case-based reasoning

2.1. Description

Case-based reasoning (CBR) solves a problem by recalling similar past problems [57] assumed to have similar solutions. Numerous past cases are needed to adapt their solutions or methods to the new problem. CBR recognises that problems are easier to solve by repeated attempts, accruing learning. It involves four steps [1] (Fig. 1):

(1) *retrieve* the most relevant past cases from the database;
(2) *use* the retrieved case to produce a solution of the new problem;
(3) *revise* the proposed solution by simulation or test execution; and
(4) *retain* the solution for future use after successful adaptation.

---

[*] Corresponding author. Tel.: +61 2 6125 4742; fax: +61 2 6125 8395.
E-mail address: tony.jakeman@anu.edu.au (A.J. Jakeman).

Problem

New Case

RETRIEVE

Learned Case

Database

Retrieved Case

RETAIN

REUSE

Tested/ Repaired Case

Solved Case

REVISE

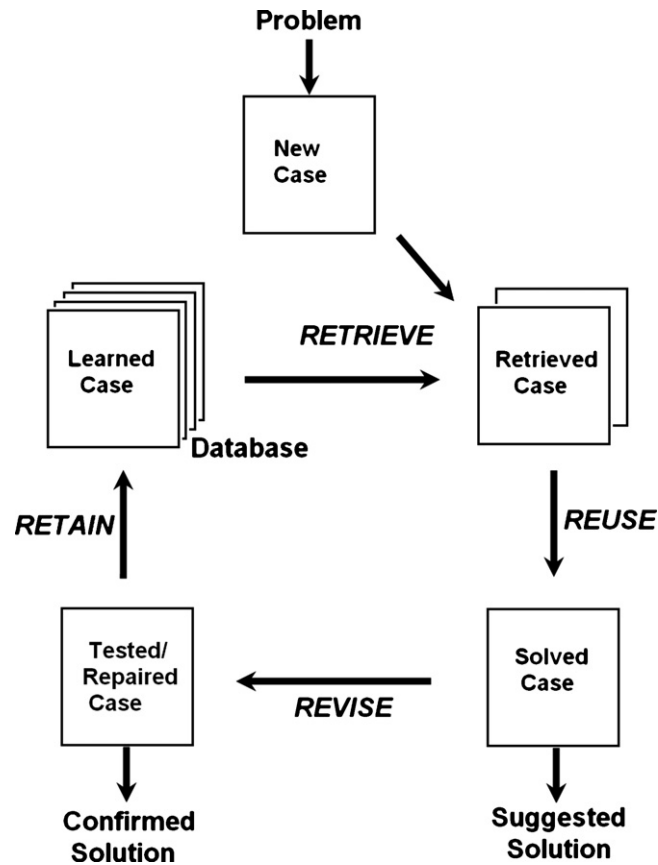**Confirmed Solution**

Suggested Solution

Fig. 1. The CBR four-step process (adapted from [1]).

Retrieval recognises either syntactical (grammatical structure) or semantic (meaning) similarity to the new case. Syntactic similarities tend to be superficial but readily applied. Semantic matching according to context is used by advanced CBR [1]. The main retrieval methods are nearest-neighbour, inductive and knowledge-guided [136].

Nearest-neighbour retrieval finds the cases sharing most features with the new one and weights them by importance. Determining the weight is the biggest difficulty [166]. The method overlooks the fact that any feature's importance, influenced by other features, is case-specific [12]. Retrieval time increases linearly with database size, so the method is time-consuming [166]. The inductive method decides which features discriminate cases best [166]. The cases are organised in a decision tree according to these features, reducing retrieval time. However, the method demands a case database of reasonable size and quality [136]. Knowledge-guided retrieval applies existing knowledge to identify the important features in each case, assessing all cases independently. It is expensive for large databases and thus often used with others [166].

After retrieval, CBR adapts past solutions to the new problem. Adaptation is structural or derivational [166]. Structural adaptation creates a solution to the new problem by modifying the solution of the past case; derivational adaptation applies the algorithms, methods or rules used in the past case to the new case. The proposed solution is then evaluated and revised if necessary. After it is confirmed, the solution is stored in the database. Redundant cases can be removed and existing cases combined [57].

2.2. Discussion

By updating the database, a CBR system continually improves its reasoning capability and accuracy and thus performance. CBR can handle large amounts of data and multiple variables. It organises experience efficiently. However,

CBR cannot draw inferences about problems with no similar past cases. Also, CBR is a black-box approach offering little insight into the system and processes involved. This is not a drawback for complex processes where understanding is neither possible nor necessary.

CBR can be employed for diagnosis, prediction, control and planning [57]. Environmental applications include decision support [90,121], modelling estuarine behaviour [126], planning fire-fighting [9], managing wastewater treatment plants [138,137,132,164], monitoring air quality [86], minimising environmental impact through chemical process design [96] and weather prediction [139].

Box 1 An application of case-based reasoning.

"Constructed wetlands: performance prediction by case-based reasoning." [101]

Lee et al. [100] tested the efficiency of constructed wetland filters by analysing the quality of water fed through them. Half the filters received inflow contaminated with heavy metals. CBR was applied [101] to improve water-quality monitoring and interpretation. Biochemical oxygen demand (BOD) and suspended solids (SS) concentrations are water-quality indicators commonly used for constructed wetlands. Measuring them is expensive, time-consuming and labour-intensive. To reduce costs, Lee et al. [101] applied CBR to predict BOD and SS concentrations of treated samples. It stored past cases with up to six variables, turbidity, conductivity, redox potential, outflow water temperature, dissolved oxygen and pH, selected for their predictive potential for BOD and SS, cost-effectiveness and ease of measurement. Using statistical equations, local similarities between each past case and the new problem were calculated with respect to one variable and global similarity with respect to all variables. The three to five past cases ranked highest by global similarity were selected. The target variables were predicted by combining their values for those cases. The local similarity of variable i for past case c and problem case p is

$$l_i = f\left(\left| \frac{V_{ip} - V_{ic}}{\mathsf{MV}_i} \right| \right)$$

where V_{ip} and V_{ic} are the values of variable i for the problem case and past case, respectively, MV_i is the mean of variable i in the case base; $|(V_{ip} - V_{ic})/\mathsf{MV}_i|$ is the local difference; and f maps local difference to local similarity. Global similarity is

$$g = \frac{\sum_i (l_i * w_i)}{\sum_i w_i}, \quad i = 1, 2, \dots, n$$

where n variables represent a case and w_i is the weight of variable i. The proportion P_j of the prediction from past case j is

$$P_j = \frac{g_j}{g_\mathsf{T}}, \quad j = 1, 2, 3, 4, 5$$

and g_T is the sum of the global similarities of the 3–5 selected cases. Prediction P is

$$P = \sum_j (P_j \times \mathsf{TV}_j)$$

where TV_j is target variable of past case j.

Outflow BOD and SS were successfully predicted by CBR, with 85% success rate in predicting whether water samples complied with regulatory requirements. Constructed wetlands are considered hard to model because of their highly complex biochemical processes. However, this study showed that CBR can be applied to such systems.

3. Rule-based systems

3.1. Description

Rule-based systems (RBS) solve problems by rules derived from expert knowledge [74]. The rules have condition and action parts, *if* and *then* and are fed to an inference engine, which has a working memory of information about the problem, a pattern matcher and a rule applier. The pattern matcher refers to the working memory to decide which rules are relevant, then the rule applier chooses what rule to apply. New information created by the action (*then-*) part of the rule applied is added to the working memory and the match-select-act cycle between working memory and knowledge base repeated until no more relevant rules are found [118].

The facts and rules are imprecise. Uncertainty can be incorporated in RBS by approaches such as subjective probability theory, Dempster-Shafer theory, possibility theory, certainty factors and Prospector's subjective Bayesian method, or qualitative approaches such as Cohen's theory of endorsements [118]. They assign to the facts and rules uncertainty values (probabilities, belief functions, membership values) given by human experts. There are two rule systems, forward and backward chaining [167]. Forward chaining is data-driven: from the initial facts it draws conclusions using the rules. Backward chaining is goal-driven: beginning with a hypothesis, it looks for rules allowing it to be sustained. That is, forward chaining discovers what can be derived from the data, backward chaining seeks justification for decisions [56].

3.2. Discussion

RBS are easy to understand, implement and maintain, as knowledge is prescribed in a uniform way, as conditional rules. However, the solutions are generated from established rules and RBS involve no learning. They cannot automatically add or modify rules. So a rule-based system can only be implemented if comprehensive knowledge is available [41], and its application is quite limited. RBS are well suited to problems where experts can articulate decisions confidently and where variables interact little. They may be difficult to scale up, as interactions then emerge.

Ecological systems, with complex interactions and processes often not well understood, do not favour RBS. RBS are often used in narrower areas such as plant and animal identification or disease and pest diagnosis [107,177,52]. RBS can also be employed as an assessment tool, e.g. in evaluating regional environments [91] or assessing the impact of water-regime changes on wetland functions [84]. Other studies have applied elements of RBS with other modelling techniques (Section 11) in landscape-change modelling based on sea-level-rise scenarios [23] and assessing the environmental life cycle of alternative energy sources [158].

Box 2 An application of rule-based systems.
"A diagnostic expert system for honeybee pests." [106]
Mahaman et al. developed a rule-based system to diagnose honeybee pests and recommend treatments. Pests (diseases, insects, mites, birds and mammals) may reduce the quantity and quality of honey production. Threats to bee health must be identified and managed.
The first step was to learn from the literature and experts about pest symptoms. The results were converted to conditional rules by backward chaining, specifying a goal then seeking facts to support it. The system asks the user questions from the rules until a diagnosis is reached, e.g.:

- *If* the affected stage of the bee is the larva
- *and* the colour of the infected larva is black *or* white
- *and* the infected larva appears mummified
- *or* the cell cappings appear scattered
- *then* chalkbrood disease confidence = 80%.

The user is shown images of the symptoms and given possible control measures. Mahaman et al. [106] checked diagnosis by the expert system with human experts. The results showed that the system was effective, with performance close to that of a human expert.

4. Artificial neural networks

4.1. Description

Artificial neural networks (ANNs) employ a caricature of the way the human brain processes information. An ANN has many processing units (neurons or nodes) working in unison. They are highly interconnected by links (synapses) with weights [170,133,5]. The network has an input layer, an output layer and any number of hidden layers. A neuron is linked to all neurons in the next layer [71], as shown in Fig. 2.

Neuron x has n inputs and one output

$$y(x) = g\left(\sum_{i=0}^{n} w_i x_i\right)$$

where $(w_0, ..., w_n)$ are the input weights and g is the non-linear activation function [119], usually a step (threshold) function or sigmoid (Fig. 3). The step-function output is $y = 1$ if $x \geq \theta$, 0 if $x \leq \theta$. The sigmoid function, more commonly used, is asymptotic to 0 and 1 [83] and antisymmetric about $(0, 0.5)$:

$$g(x) = \frac{1}{1 + e^{-\beta x}}, \quad \beta > 0$$

ANNs may be feedforward (the commonest) or feedback. Information flow is unidirectional in feedforward ANNs, with no cycles, but in both directions in feedback ANNs so they have cycles, by which their state evolves to equilibrium [151].
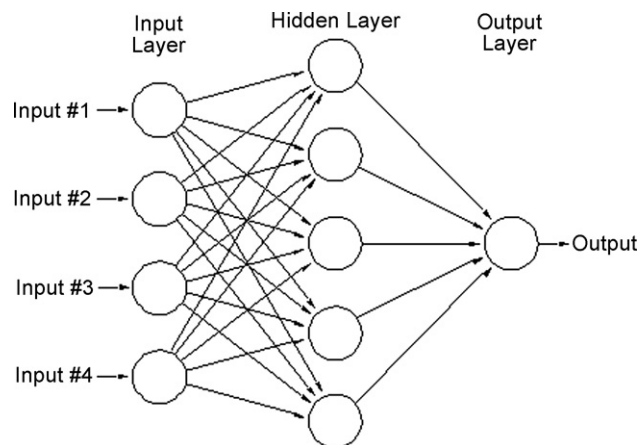

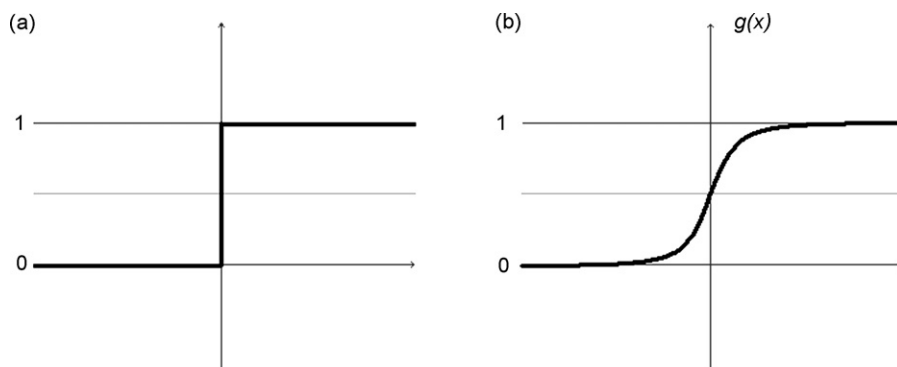Fig. 2. An example of an artificial neural network [134].

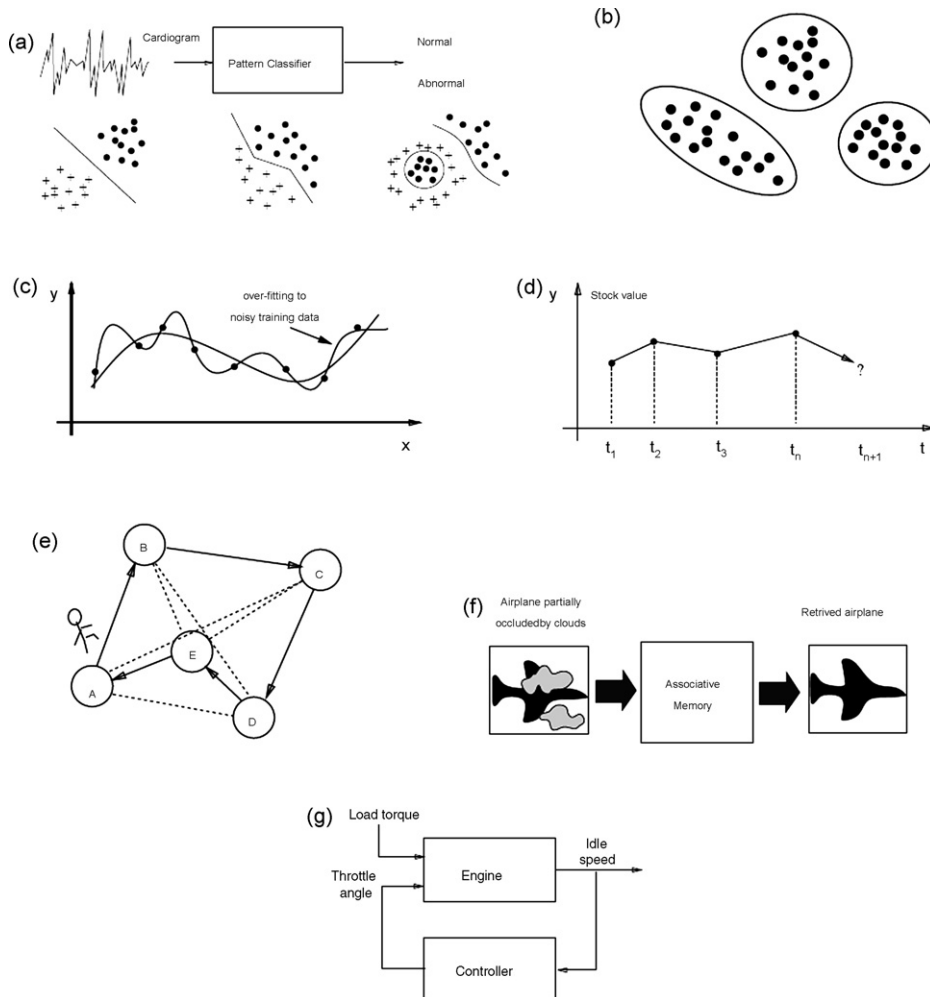Fig. 3. (a) Step function and (b) sigmoid function.

Fig. 4. Seven categories of tasks ANN can perform [83].

ANNs require few prior assumptions, learning from examples [133] by adjusting the connection weights. Learning may be supervised or unsupervised. Supervised learning gives the ANN the correct output for every input pattern. The weights are varied to minimise error between ANN output and the given output. One form of supervised learning, reinforcement learning, tells the ANN if its output is right rather than providing the correct value [170]. Reinforcement learning is discussed in Section 10. Unsupervised learning gives the ANN several input patterns. The ANN then explores the relations between the patterns and learns to categorise the input [83]. Some ANNs combine supervised and unsupervised learning.

4.2. Discussion

ANNs are useful in solving data-intensive problems where the algorithm or rules to solve the problem are unknown or difficult to express [178]. The data structure and non-linear computations of ANNs allow good fits to complex, multivariable data. ANNs process information in parallel and are robust to data errors. They can generalise, finding relations in imperfect data so long as they do not have enough neurons to overfit data imperfections. A disadvantage of ANNs is that they are uninformative black-box models and thus unsuitable for problems requiring process explanation. If an ANN fails to converge, there is no means to find out why.

ANNs can be applied to seven categories of problems [83] (Fig. 4): pattern classification, clustering, function approximation, prediction, optimisation, retrieval by content and process control. Pattern classification assigns an

input pattern to one of the pre-determined classes, e.g. land classification from satellite imagery [140] or sewage odour classification [122]. Clustering is unsupervised pattern classification, e.g. of input patterns to predict ecological status of streams [163]. Function approximation, also called regression, generates a function from a given set of training patterns, e.g. modelling river sediment yield [33] or catchment water supply [79], predicting ozone concentration [147], modelling leachate flow rate [88] or estimating the nitrate distribution in groundwater [8]. Prediction estimates output from previous samples in a time series, e.g. of weather [95], air quality [7,145] or water quality [178]. Optimisation maximises or minimises a cost function subject to constraints, e.g. calibrating infiltration equations [82]. Retrieval by content recalls memory, even if the input is partial or distorted, e.g. producing water-quality proxies from satellite imagery [129]. An example of process control is engine speed control, keeping speed near-constant under varying load torque by changing throttle angle [83].

Box 3   An example of an application of artificial neural networks.

"Use of ANNs for modelling cyanobacteria *Anabaena* spp. in the River Murray, South Australia."
Maier et al. [108] used ANNs to model the occurrence of cyanobacteria *Anabaena* spp. in the River Murray at Morgan, South Australia. This species group, the commonest in the lower Murray, is of great concern as Adelaide gets its water from there. Other studies had shown failure of process-based and traditional statistical approaches to predict the size and timing of incidence of algae species. ANNs seemed well suited to modelling ecological data, not requiring the probability distribution of the input data.

The concentrations of *Anabaena* from 1985/1986 to 1992/1993 were tested against eight weekly variables: colour, turbidity, temperature, flow, total phosphorus, soluble phosphorus, oxidised nitrogen and total iron. The model included time lags of up to 21 weeks. Various network geometries were examined. The number of hidden layer nodes was limited to ensure the ANN could approximate a wide range of continuous functions yet avoid overfitting the training data:

$$N_H \leq 2N_I + 1; \quad N_H \leq \frac{N_{TR}}{N_I + 1}$$

where N_H is the number of hidden layer nodes, N_I is the number of inputs and N_{TR} is the number of training samples (here 365).

Training data were from 1985/1986 to 1991/1992. Earlier trials had determined the optimum learning rate and number of training samples. After training, data for November 1992 to April 1993 tested the network's ability to provide 4-week forecasts. Sensitivity analyses were conducted on the best models to determine the relative strength of relations between the input and output variables.

The ANN model indicated that flow, temperature and colour had most influence on the *Anabaena* spp. population. Addition of nutrients, iron and turbidity did not significantly improve the forecasts. The ANN successfully forecast the major peak in *Anabaena* spp. concentration in the River Murray, although it missed a minor peak early in the test period. It was suggested that the dataset used for model calibration contained insufficient data on that type of event. Maier et al. [108] concluded that ANNs were a useful tool for modelling cyanobacteria incidences in rivers.

---

5. Genetic algorithms

5.1. Description

A genetic algorithm (GA) is a search technique mimicking natural selection [24]. The algorithm evolves until it satisfactorily solves the problem, through the fitter solutions in a population surviving and passing their traits to offspring which replace the poorer solutions. Each potential solution is encoded, for example as a binary string, called a chromosome. Successive populations are known as generations [23].

The initial population $G(0)$ is generated randomly. Thereafter $G(t)$ produces $G(t+1)$ through selection and reproduction [24]. A proportion of the population is selected to breed and produce new chromosomes. Selection is according to fitness of individual solutions, i.e. proximity to a perfect solution [26], most often by roulette selection and deterministic sampling. Roulette selection randomly selects a parent with probability calculated from the fitness f_i of each individual by [24]:

$$F_i = \frac{f_i}{\sum_i f_i}$$

Deterministic sampling assigns a value $C_i = \mathrm{RND}(nF_i) + 1$ to organism i of the n organisms in the population, where RND rounds its argument to the nearest integer. Each organism is selected as a parent C_i times [24].

Reproduction is by genetic crossover and mutation. Crossover produces offspring by exchanging chromosome segments from two parents [26]. Mutation randomly changes part of one parent's chromosome. This occurs infrequently and introduces new genetic material [24]. Although mutation plays a smaller part than crossover in advancing the search, it is critical in maintaining genetic diversity. If diversity is lost, evolution is retarded or may stop [76]. In steady-state GAs, offspring generated by the genetic operators replace less fit members, resulting in higher average fitness. Simple or generational algorithms replace each entire generation [24]. Selection and reproduction are repeated until a stopping criterion is met [24], e.g. all organisms are identical or very similar, a given number of evaluations has been completed, or maximum fitness has been reached; evolution no longer yields better results.

5.2. Discussion

GAs are computationally simple and robust, and balance load and efficacy well [67]. This partly results from only examining fitness, ignoring other information such as derivatives [94]. GAs treat the model as a black box, an advantage when detailed information is unavailable.

An important strength of GAs is implicit parallelism; a much larger number of code sequences are indirectly sampled than are actually tested by the GA [76]. Unlike most stochastic search techniques, which adjust a single solution, GA keeps a population of solutions. Maintaining several possible solutions reduces the probability of reaching a false (local) optimum [67]. Therefore GAs can be very useful in searching noisy and multimodal relations. However, the latter may take a large computation time.

In most cases, GAs use randomisation in selection. They avoid picking only the best individual and thus prevent the population from converging to that individual. However, premature convergence on a local optimum can occur if the GA magnifies a small sampling error [63]. If a very fit individual emerges early and reproduces abundantly, early loss of diversity may lead to convergence on that local optimum.

GAs often optimise model parameters or resource management. Some application examples include modelling species distribution [153], air quality forecasting [87], estimating soil bulk density [38], calibrating water-quality models [127], finding the best solution in an integrated management system [157] and water management [80].

6. Cellular automata

6.1. Description

Cellular automata are dynamic models, discrete in space, time and state. They consist of a regular lattice of cells which interact with their neighbours. The cell states are synchronously updated in time according to local rules, which calculate the new state of a cell at time $t+1$ using its state and those of neighbouring cells at time t [34]. The neighbours are all those in given proximity, as illustrated in Fig. 5.

The simplest 'elementary' cellular automaton (CA) is a linear chain of n cells. The cell states in a one-dimensional CA are

$$a_i(t+1) = f(a_{\{i\}}(t))$$

Box 4   An example of an application of genetic algorithms.

"Using genetic algorithms to optimise model parameters." [165]

Wang [165] used a GA to calibrate a conceptual rainfall-runoff model of the Bird Creek catchment, Oklahoma. The model had nine parameters, dealing with water balance and routing. The GA optimised the parameters by minimising the sum of squares of differences between computed and observed daily discharges.

The GA had a population size of $n = 100$ potential parameter solutions. The chromosomes had 7 bits, allowing 128 values. Selection assigned probability p_j to point j:

$$p_j = p_1 + \frac{(j-1)(p_n - p_1)}{n-1}$$

The average probability was $1/n$, the best point's was 1.5 times the average and the probabilities summed to 1 over all points. After ranking, the best point is $j = n$, with the largest probability value p_n. The worst is $j = 1$, with smallest probability value, p_1.

Two points A and B are selected at random according to the probability distribution. Two bit positions, k_1 and k_2, randomly selected in the chromosome coding, determine where crossover occurs. A new point is produced by taking the values of bits k_1 to $k_2 - 1$ of A's coding and values of bits from k_2 to the end and from 1 to $k_1 - 1$ of B's coding. Mutation of new points occurs with probability 0.01.

The GA generated 10 runs with randomly selected initial populations. Each run was stopped after 10,000 fitness evaluations. For all runs, the best objective function values were very close to the global minimum. When the search converged on a local optimum, the local optimum had a similar objective value to that of the global optimum. Wang [165] concluded that GAs are capable and robust.

---

where a_i is the state of cell i at time t and $f(a_{\{i\}}(t))$ is a function of the states of its neighbouring cells [150]. If the neighbourhood is the immediately adjacent cells, then

$$f(a_{\{i\}}(t)) \equiv f(a_{\{i-1\}}(t), \; a_{\{i\}}(t), \; a_{\{i+1\}}(t)).$$

For a 5-neighbour square two-dimensional CA with cells $(i, j+1)$ $(i+1, j)$ $(i, j-1)$ $(i-1, j)$ as the neighbours of cell (i, j) [150],

$$a_{i,j}(t+1) = f(a_{\{i,j\}}(t), \; a_{\{i,j+1\}}(t), \; a_{\{i+,j\}}(t), \; a_{\{i,j-1\}}(t), \; a_{\{i-1,j\}}(t))$$

As the cells are updated, the disordered initial state of the CA usually evolves [150] into one of four patterns I–IV illustrated in Fig. 6: homogeneous, simple separated periodic structures, chaotic aperiodic patterns or a complex pattern of local structures.

These patterns correspond to four classes of CA described by Wolfram [168]. Classes I, II and III are analogous to the limit points, limit cycles and chaotic attractors observed in continuous dynamical systems. Class IV CAs have no direct analogue, exhibiting complex behaviour neither completely repetitive nor completely random.

6.2. Discussion

One of the main problems with cellular automata concerns boundary conditions. In the periodic condition, the cells on one edge are made neighbours with those on the opposite edge, producing a torus. Other possibilities are reflecting and absorbing boundary conditions, which assume no off-lattice neighbours and assign the border states unique values or rules. Unfortunately these boundary conditions are unlikely to reflect real life. The problem could be avoided by making the lattice significantly larger than the area under study, at a higher cost in resources and computation [62].

Cellular automata are simple mathematical models that can simulate complex physical systems. They can incorporate interactions and spatial variations, and also spatial expansion with time [54]. For some parameter values, they are very
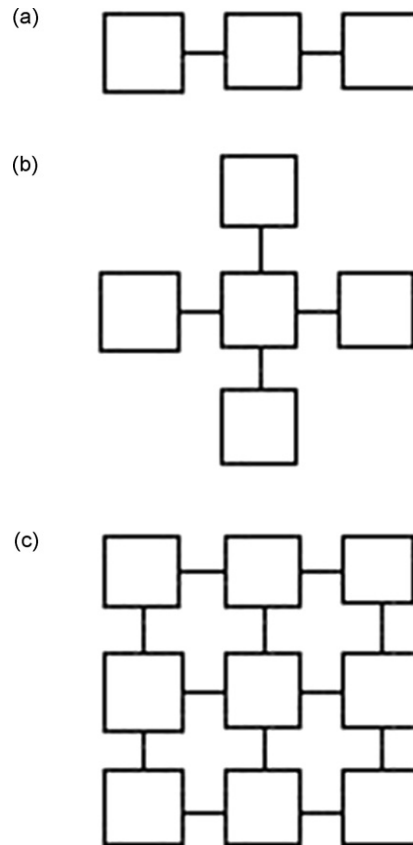
Fig. 5. Common neighbourhood configurations: (a) neighbourhood $\{-1, 0, 1\}$, (b) 5-neighbour square, also referred to as 'von Neumann neighbourhood', and (c) 9-neighbour square, also referred to as 'Moore neighbourhood' (adapted from [68]).

sensitive to the initial state and transition functions. Therefore, they have limited capacity to make precise predictions [81], but are often used to understand real environments.

Examples of CA applications include modelling population dynamics [51,152], animal migration [143], unsaturated flow [61], landscape changes [55], debris flow [39], earthquake activity [65], lava flow [37] and fire spread [75,89]. In these examples it is clear how the state of a cell is influenced by the previous state of its neighbours.
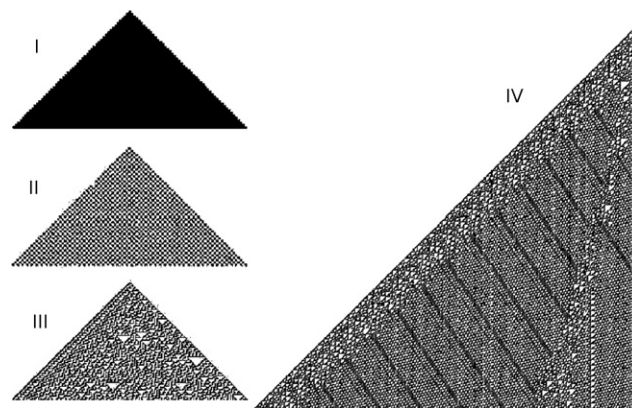

Fig. 6. The four classes of evolved patterns in CA: (I) homogeneous, (II) periodic, (III) chaotic, and (IV) complex pattern of localised structures [114].

Box 5 An example of an application of a cellular automaton.

"Simulation of vegetable population dynamics based on Cellular Automata." [10]

Bandini and Pavesi [10] developed a CA-based model that simulated the population dynamics of robiniae, oak and pine trees on the foothills of the Italian Alps. A two-dimensional CA was created, with individual cells representing portions of a given area. The state of each cell was defined by:

  (i) the presence or absence of a tree,
 (ii) the amount of resources (water, light, N and K) present, and
(iii) if a tree was present, its features (species, size and resource needs)

A cell can host a tree if the resources are suitable. The sprouting of a tree also requires that the cell contains a seed and no other tree. Seeds can be introduced into the cell from fruiting trees in neighbouring cells. As the tree grows, it has a greater resource requirement, obtainable from neighbouring cells.

The development of this model involved defining rules describing tree sustenance and reproduction, resource production and flow, and the influence of a tree on neighbouring cells. Two neighbourhood configurations, 'von Neumann' (5-neighbour) and 'Moore' (9-neighbour) were considered.

The initial parameters of each cell are defined by the user. As the model runs, the system evolves step by step. Bandini and Pavesi [10] found that the simulations were qualitatively similar to real cases.

---

7. Fuzzy systems

7.1. Description

Fuzzy systems (FS) use fuzzy sets to deal with imprecise and incomplete data. In conventional set theory an object is a member of a set or not, but fuzzy set membership takes any value between 0 and 1. Thus fuzzy models can describe vague statements as in natural language [117]. Roberts [131] gives an example of a FS, a vegetation model assigning plant communities to community types, with fuzzy membership indicating the degree to which the vegetation meets each type definition. The membership values for each plant community sum, over all community types, to 1. For example, plant community χ had fuzzy memberships $\{\mu_i\}$: $\mu_{(PIPO/PIPO)} = 0.2$, $\mu_{(PIPO/PSME)} = 0.3$, $\mu_{(PIPO/PIPU)} = 0.1$ and $\mu_{(PIPO/ABCO)} = 0.4$ [131].

Fuzzification transforms exact (crisp) input values into fuzzy memberships [93]. Fuzzy models are built on prior rules, combined with fuzzified data by the fuzzy inference machine. The resulting fuzzy output is transformed to a crisp number (defuzzification [142]). Techniques include maximum, mean-of-maxima and centroid defuzzification. Fig. 7 shows the components of a fuzzy system.

7.2. Discussion

Human reasoning handles vague or imprecise information. The ability of fuzzy systems to handle such information is one of its main strengths over other AI techniques, although they are mostly easier to understand and apply. One of the main difficulties in developing a fuzzy system is determining good membership functions. Fuzzy systems have no learning capability or memory [64]. To overcome such limitations, fuzzy modelling is often combined with other techniques to form hybrid systems, e.g. with neural networks to form neuro-fuzzy systems (see Section 11).

Fuzzy systems handle incomplete or imprecise data in applications including function approximation, classification/clustering, control and prediction. Instances include modelling vegetation dynamics [131], estimating soil hydraulic properties [58], modelling macroinvertebrate habitat suitability [162], evaluating habitat suitability for riverine forests
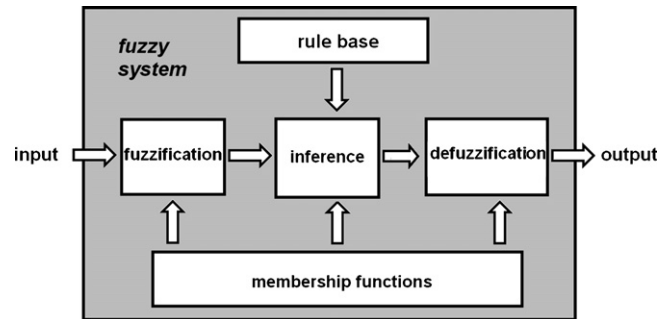
Fig. 7. The main components of a fuzzy system (adapted from [142]).

[141], modelling fishery landings [98], identifying ecological conditions in a lagoon [111], identifying oil spills from satellite images [93], modelling solute transport [48], controlling wastewater treatment [173], combining rainfall-runoff forecasts [169], predicting soil erosion [113], assessing environmental risk of drilling waste discharge [135], estimating fire risk [78], predicting disease risk [59] and predicting regional drought [128].

Box 6 An example of an application of fuzzy systems.

"One-, two- and three-dimensional modelling of water movement in the unsaturated soil matrix using a fuzzy approach." [11]

Bardossy et al. [11] aimed at a simplified model of one-, two- and three-dimensional unsaturated water flow in an unsaturated soil matrix, using fuzzy rules based on the Richards equation. The Richards equation combines the unsaturated Darcy law and conservation of mass. However, calculating two- and three-dimensional flow by this equation is made impracticable in many cases by high computational demand.

The flow between two adjacent cells, j and $j+1$, was determined by:

$$q_{j,j+1} = f\left[\left(\frac{\theta}{\theta_s}\right)_j, \left(\frac{\theta}{\theta_s}\right)_{j+1}, \theta_{s,j}, \theta_{s,j+1}, k_{s,j}, k_{s,j+1}\right]$$

where θ/θ_s is the relative soil moisture of the cells, θ_s is the moisture at saturation, and k_s is the saturated hydraulic conductivity. Different rules were used for vertical and horizontal flow. An example of a fuzzy rule for vertical flow (q_v) is:

If $(\theta/\theta_s)_u$ is 'high' AND $(\theta/\theta_s)_l$ is 'low'

THEN q_v is 'high'

The fuzzy model with only two parameters, θ_s and k_s, performed well, running faster than its classical counterpart. However, Bardossy et al. [11] stressed that the fuzzy model is not a replacement for the other model but a simplification of it. The Richards equation can provide information on the cause of the water movement but the fuzzy model cannot, as it essentially only mimics water movement. The fuzzy model only suits applications that require a solution that only describes water movement.

8. Multi-agent systems

8.1. Description

A multi-agent system (MAS) comprises a network of agents interacting to achieve goals [102]. An agent is a software component containing code and data [125]. It is incapable of solving the problem assigned to the MAS on its own [60]. The agents communicate by a high-level Agent Communication Language (ACL) through which they share information, request services and negotiate with each other [125].

The most commonly used ACL is KQML (Knowledge Query and Manipulation Language). It has a communication layer covering low-level parameters such as sender, recipient and communication identifiers. A message layer specifies the performative and interpretation protocol, and a content layer contains further information on the performative [60].

Agent coordination is important, as messages are relayed and responses processed asynchronously between agents. Coordination depends on the system's infrastructure, which governs the pathways of information flow, degree of concurrency, resource use and nature of interactions between agents [125]. The simplest infrastructure is a peer-to-peer network, where all agents communicate directly. It suits problems needing fast, concurrent processing to arrive at a solution without conflict resolution. Two other infrastructures, federated agent and multi-agent blackboard, are commoner in MASs. A federated agent network contains facilitator agents as mediators between agents. Multi-agent blackboard networks contain a central controller to coordinate agent activity and a public blackboard space where all data are shared [125].

8.2. Discussion

MASs can model complex systems with multiple interactions among dynamic and autonomous entities. Their effectiveness largely depends on the agent organisation. With the peer-to-peer infrastructure, network maintenance can be problematic, as all agents are updated every time a new network component is added. In infrastructures with central coordination, only the facilitator's directory must be updated with new additions. However, processing bottlenecks may occur [125]. Other difficulties in constructing MASs arise from their dynamic nature and the sophisticated interactions between agents, where goals or allocation of tasks and resources may conflict [156].

Most early MAS applications included manufacturing, process control and information management [156]. Recently MASs have been widely applied in natural-resource management [22,16,20,72]. Examples include rangelands [70], the fishing [14], dairy [125] and forestry industries [130], forest ecosystems undergoing land-use change [115], urban catchments [123] and irrigated farming systems [13] (Box 7). MASs are well suited to this field because of their ability to represent complex systems with several stakeholders and allow exploration of alternative management approaches. They do, however, seem more suited to social learning among interest groups than prediction of system behaviour.

9. Swarm intelligence

9.1. Description

Swarm intelligence (SI) is a form of agent-based modelling inspired by colonies of social animals such as ants and bees [40] or schools of fish. While individual agents are simple, as a swarm they exhibit higher intelligence. Self-organisation is a key feature, whereby global patterns emerge from local interactions, without centralised control or a global model. These interactions can occur through direct (agent-to-agent) or indirect (via the environment) communication. Another feature of SI is this indirect communication by modification of the environment, also referred to as stigmergy [18]. An example of stigmergy is when ants deposit a pheromone while walking towards food sources or their nest so that other ants can follow their trail.

The behaviour of ants is the basis of Ant Colony Optimisation (ACO), one of two main types of SI algorithms. Isolated ants move randomly until they encounter a pheromone trail, which they are likely to follow and subsequently reinforce with their own pheromone. Ants tend to choose trails with stronger pheromone concentrations, resulting in autocatalytic behaviour, where the more traffic a trail receives, the more appealing it becomes [47]. Shorter paths between destinations require less transit time and become dominant as they are reinforced with more pheromone [47,176]. Through autocatalytic behaviour, all ants eventually choose the shortest trail [47]. Decay of pheromone with

---

Box 7   An example of an application of multi-agent systems.

"Suitability of Multi-Agent Simulations to study irrigated system viability: application to case studies in the Senegal River Valley." [13]

Barreteau et al. [13] assessed the suitability of MASs to study viability of an irrigated system in the Senegal River Valley. An MAS called SHADOC was used to test scenarios in the virtual irrigated system. The Senegal River Valley provides water for pumping, and has ideal soil and climate for rice cultivation and adequate labour resources. Development over the last few decades has been less than expected, prompting exploration of new tools to improve prospects. The irrigated system is complex, with multiple schemes, stakeholders and interactions, favouring an MAS.

Water is pumped from the river and run through the scheme. A modification, for instance of the amount of water available, is perceived by each agent (farmer or group). The agents may then get messages from others, such as an agreement to share water, to which they respond by acting on the environment and sending messages. Each farmer belongs to groups but is uniquely represented in the system, which includes the state of his plot, water availability and friendship group, i.e. other farmers with whom he is willing to cooperate. Each farmer has a goal defining investment priorities and other activities, and rules describing his behaviour.

A scenario was defined by the state of the environment, a set of individual rules and a set of collective rules. One hundred random scenarios were run and assessed for viability. These simulations allowed the irrigated system behaviour to be explored, providing a new means of understanding and learning and minimising impacts on real systems.

---

time leads to random or less travelled (longer) paths decaying to zero concentration, analogously to gradual memory loss [176].

ACO addresses discrete combinatorial optimisation, as in the travelling salesman and quadratic assignment problems [144]. Populations of potential solutions are generated, each represented by a single path. Each path can have multiple decision nodes. An 'ant' constructs a path from source to destination, selecting an edge at each node by a probabilistic decision policy which takes account of pheromone intensity and the desirability of the edge, measured by its objective function. For example, if the aim is to find the minimum distance between two points, shorter edges are more desirable and the desirability of an edge is inversely proportional to its length. The probabilistic decision policy is commonly [47,110]

$$p_{i,j}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{l \in \theta_i}[\tau_{i,l}(t)]^{\alpha}[\eta_{i,l}]^{\beta}}$$

where $p_{i,j}(t)$ is the probability of edge (i, j) being chosen at iteration t, $\tau_{i,j}$ is the pheromone intensity associated with edge (i, j), $\eta_{i,j}$ is the desirability of edge (i, j), α and β are parameters controlling the respective importance of the pheromone intensity and desirability, and θ_i is the set of edges l available at decision node i.

When the ants reach the destination, the process is iterated over a specified number of cycles or until all ants use the same path.

The other commonly applied SI algorithm is Particle Swarm Optimisation (PSO), inspired by the synchrony of flocks of birds and schools of fish. The system is initialised with a random population of potential solutions or particles, which are 'flown' through the N-dimensional problem-parameter space [92], in which each solution is a point. At each iteration, the particles evaluate their fitness (positions relative to the goal) and share memories of their best positions with the swarm. Subsequently each particle updates its velocity and position according to its best previous position and that of the best particle in the swarm [53]. The algorithm also contains random functions in the range [0,1] to help prevent convergence on local optima. The PSO algorithm adjusts the velocity $V(t)$ vector of each particle by [159]:

$$V(t + 1) = aV(t) + bR_1[\text{pbest} - X(t)] + cR_2[\text{gbest} - X(t)]$$

where $X(t)$ is the current position vector, 'pbest' and 'gbest' are the best solutions achieved to date by the particle and population, respectively, R_1 and R_2 are the random numbers in [0,1], a is the momentum parameter, b is the

self-influence parameter, and c is the swarm influence parameter. This is followed by adjustment of the position [159] by

$$X(t + 1) = X(t) + V(t + 1)$$

The process ends when a specified number of iterations or CPU time is reached or 'gbest' cannot be further improved.

9.2. Discussion

These two SI techniques use algorithms versatile yet simple enough to be readily accessible for practical applications and easy to implement. They distribute control through local rules between multiple agents, enabling the group to perform its tasks even when one or more individuals fail [46]. This self-organisation allows adaptation to changes in the environment. These features make SI systems robust against failures and perturbations [18], and give them the ability to solve dynamic problems.

An important feature of ACO is autocatalytic behaviour, positive feedback to allow quick convergence to the optimum. However, feedback may lead to stagnation and sub-optimal convergence. For small problems, this is less of an issue as the optimum is more likely to be located before stagnation [6]. Exploitation and exploration must be traded to balance optimality against computational efficiency, by adjusting the parameters of the decision policy and pheromone decay. Refinements of the original ACO algorithm improve exploitation of good solutions [25,45,36], e.g. the Max-Min Ant System algorithm which only adds pheromone to the best solutions and limits the strength of pheromone added, to help avoid premature convergence [154].

As for GAs, PSO algorithms perform non-linear optimisation with a population of potential solutions, so are resistant to trapping in local optima. However, rather than evolving 'offspring', PSO particles evolve their behaviour (velocities and positions) [53]. For success, PSO must balance self-influence and swarm influence. If swarm influence is neglected, poor exchange of information can lead to slow convergence. Overemphasis on self-influence can lead to premature convergence through inadequate exploration [172]. The influence parameters are often set equal.

ACO was originally developed for the travelling-salesman problem, and has become popular in engineering for optimal routing in telecommunication networks [42], of vehicles [44], water distribution systems [109,175], and stormwater networks [6]. The technique has also been applied in management for time scheduling [146] and resource allocation [171]. Environmental application of ACO has so far been limited, but the previous examples demonstrate its potential. Examples of environmental applications include optimising sampling locations for long-term groundwater monitoring [103] (Box 8), estimating unsaturated soil hydraulic parameters [2] and clustering ecological data for river health assessment [110].

PSO was developed for global minimisation, but as for ACO has a wide range of applications, including pattern recognition [174], project scheduling [179], resource allocation [172] and predictive temperature control [35]. Environmental examples include optimising pump-and-treat systems for groundwater remediation [112] and optimising parameters in a rainfall-runoff model and a soil moisture prediction model [66]. PSO has been successfully used to train ANNs, for instance ANNs to predict river water levels [30], water quality [29] and air pollutant levels [105].

10. Reinforcement learning

10.1. Description

Reinforcement learning (RL) is learning through interaction between a learning agent and its environment [155]. The agent learns to achieve a goal by trial and error. An RL problem has three parts: environment, reinforcement function and value function [73]. The environment is dynamic and has a set of possible states. For each state s_t at time t there is a set $A(s_t)$ of possible actions [155].

At time step t_k the agent perceives the state s_t of the environment and chooses an action a which then alters s_{t+1}. The environment responds with a scalar reinforcement (reward) signal r. The agent aims to select actions that maximise its total reward on the way from the initial to the goal state [85]. Three classes of reinforcement function [73] are pure delayed reward, minimum time to goal and games. Pure delayed reward gives no reward until the terminal state is reached. The reinforcement at the terminal state can be positive or negative, indicating whether it is a goal state ($r = 1$) or unfavourable ($r = -1$). Minimum-time-to-goal reinforcement return $r = -1$ if the goal state is not reached on

---

Box 8   An example of an application of a swarm intelligence technique.
"Optimal groundwater monitoring design using an ant colony optimisation paradigm." [103]
An aquifer in Pierce County, Washington, USA was contaminated by trichloroethylene up to the 1970s. Long-term monitoring (LTM) of the groundwater had been conducted to assess the performance of remediation and the risk to human health. LTM is by a network of 30 wells, this number of locations making the program quite costly. The aim is to identify redundant monitoring locations to reduce costs while still getting sufficient information about the contaminant plume. ACO was considered appropriate for this non-linear combinatorial problem. The objective was

$$\min Z = \sqrt{\frac{\sum_{i=1}^{m}((C_{\text{est},i} - C_i)/\min(C_{\text{est},i}, C_i))}{m}}$$

where m is the number of removed wells, C_i is the measured, $C_{\text{est},i}$ is the estimated concentration in removed well i (the latter based on the remaining wells) and $m = S_{\text{goal}}$ is the desired number of removed wells.
ACO was applied. Ants select wells to remove. An ant chooses well j from its current position at well i with probability

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(n_j)^{\beta}}{\sum_{l \in L}(\tau_{il})^{\alpha}(n_l)^{\beta}}$$

where α and β equal 1 and -1, respectively, and candidate well l is in set L. After an ant has visited S_{goal} wells, the overall data loss of the reduced LTM network is calculated as the root mean square error (RMSE) of the estimated concentrations of the removed wells. Each ant colony consists of 30 ants and the search terminates after 50 runs. Optimal solutions were found for reduced LTM networks of 21–27 wells. The RMSE was found to increase nonlinearly with number of removed wells. A 30% reduction in the number of wells produced minor differences in the concentration contours compared to contours from the existing 30-well network. The authors considered ACO efficient and effective in solving the LTM optimisation problem.

---

performing an action, and zero if it is. This reinforcement function causes the agent to find the shortest path to the goal. While systems using either of these two classes aim to maximise their score, the 'games' class involves finding the maximum, minimum or saddle points of the reinforcement function. 'Games' is used when two or more competing agents have opposing goals and actions are chosen independently but executed simultaneously [73].

While the reinforcement function specifies immediate desirability, a value function determines long-term desirability [155]. The value function considers the states likely to follow an action and the reinforcements available in those states. The value of a state is the sum of reinforcement expected over the future starting from that state. It is possible for a state to attain a low (immediate) reinforcement but to have a high value if it is likely to be followed by states yielding high reinforcements [155]. Commonly the value function is approximated through use of the Markov decision process.

10.2. Discussion

Most applications of RL have been in robotics and game playing [85], where RL creates new behaviour rather than modelling existing behaviour. Computer programs use RL progressively to improve efficiency and outcomes. Limited studies have applied RL alone to solve environmental problems. However, its use with other AI techniques has increased. As RL models are agent-based, they may have to model the behaviour of individual animals or humans. For example, Tsoularis and Wallace [161] used RL to model feeding by a predator presented with similar-looking palatable and unpalatable prey. In other examples, RL was used in combination with ANN to model the behaviour of fishermen [50] and bees [120]. In all these examples RL was applied to simulations, so the environment and agents were created to fit the given algorithm. It is difficult to formulate a policy that works successfully in real life [3].

11. Hybrid systems

Hybrid systems combine two or more techniques ('paradigms') to gain strengths and overcome weaknesses. There are three main types of hybrid systems according to how the techniques are combined: sequential, auxiliary and embedded [69]. In a sequential hybrid, the first paradigm passes its output to the second to generate the output. In an auxiliary hybrid, the first paradigm obtains some information from the second to generate the output. In an embedded hybrid, the two paradigms are contained within one another [69].

The commonest hybrids are neuro-fuzzy systems, combining ANNs and fuzzy systems. They are effective: fast, efficient and easily designed, implemented and understood [49]. By combining them, the need to prime fuzzy systems is reduced by learning in ANNs. Fuzzy systems attenuate 'noise', from which some ANNs suffer.

Each AI technique has capabilities and limitations, making its suitability for environmental modelling problem-specific. For example, CA has been successful in modelling spatial changes in landscape [55]. However, by combining CA with MAS the model can incorporate human decisions driving the landscape changes [116], as in policy making and land management. In another case, an ANN and GA were combined to optimise control of catchment-sourced nutrient loads for water-quality management [99]. The ANN modelled total phosphorus concentrations in a reservoir and the GA identified the best control scheme to reduce phosphorus loads [99]. These examples demonstrate the usefulness of hybrid systems in complex problems.

On the other hand, a hybrid system may retain the weaknesses of both techniques and little of their strengths [69]. Hybrid techniques also raise the problem of communication; different representations have to be translated into a common language. Another problem of hybrid systems with learning is credit assignment [77]. If one component cannot distinguish changes caused by its own actions from those due to others, penalties and rewards will be ineffective.

In principle, hybrids can be formed by any two or more techniques. Examples are:

- FS and ANN: water-level prediction [15] and predicting groundwater vulnerability [43].
- MAS and CA: simulating land-use and land-cover change [124,116], policy analysis [17], modelling urban sprawl [104].
- Bayesian and ANN: predicting hydrological variables [97,31].
- FS and CA: modelling algal blooms [31] and insect infestation [19].
- GA and ANN: water-quality management [99], rainfall-runoff modelling [149].
- GA and Bayesian: modelling species distribution [160].
- FS and GA: modelling rainfall [28], calibrating a rainfall-runoff model [32].
- CBR and RBS: decision support for wastewater management [27].
- PSO and GA: developing pollution prevention and control strategies [159].
- ANN, FS and RL: optimising water allocation [4].

12. Overview of AI techniques

The suitability of AI techniques for environmental modelling is case-specific. Complex and poorly understood processes may favour a black-box approach by e.g. CBR, ANN or GA. CBR requires similar past cases. ANNs learn from training cases and capture relationships among data. They are versatile and have had a range of environmental applications for classification, function approximation, optimisation and prediction. GAs evolve a set of potential solutions towards a global optimum. ACO and PSO also do population-based optimisation. SI systems involve simple agents, collectively solving a problem by local interaction. For problems with well-understood processes, RBS may be applied. They are often used for diagnosis and identification of plants and animals. CAs and MASs are often used to simulate complex systems. CAs model systems discrete in space, time and state and with local interactions. They are used to understand and perhaps predict behaviour. MASs employ agents interacting to solve a common problem. They are often suited to natural-resource management, exploring management strategies with stakeholders. In contrast, RL models the trial-and-error learning of individual agents. However, RL has mostly been limited to computer programming, with few environmental applications so far. FSs, by contrast, have been applied to many environmental problems because they can handle imprecise and incomplete data. FSs are often combined with other techniques to form hybrid systems.

Advances in AI techniques and increased use in environmental studies will continue. This paper presents only a brief overview. The references provide further detail.

References

[1] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Commun. 7 (1994) 35–59.
[2] K.C. Abbaspour, R. Schulin, M.T. van Genuchten, Estimating unsaturated soil hydraulic parameters using ant colony optimization, Adv. Water Resour. 24 (2001) 827–841.
[3] P. Abbeel, M. Quigley, A.Y. Ng, Using Inaccurate Models in Reinforcement Learning, http://ai.stanford.edu/~ang/papers/icml06-usinginaccuratemodelsinrl.pdf, 2006.
[4] B. Abolpour, M. Javan, M. Karamouz, Water allocation improvement in river basin using Adaptive Neural Fuzzy Reinforcement Learning approach, Appl. Soft Comput. 7 (2007) 265–285.
[5] M. Adya, F. Collopy, How effective are neural networks at forecasting and prediction? A review and evaluation, J. Forecasting 17 (1998) 481–495.
[6] M.H. Afshar, Improving the efficiency of ant algorithms using adaptive refinement: application to storm water network design, Adv. Water Resour. 29 (2006) 1371–1382.
[7] E. Agirre-Basurko, G. Ibarra-Berastegi, I. Madariaga, Regression and multilayer perceptron-based models to forecast hourly O_3 and NO_2 levels in the Bilbao area, Environ. Modell. Softw. 21 (2006) 430–446.
[8] M.N. Almasri, J.J. Kaluarachchi, Modular neural networks to predict the nitrate distribution in ground water using the on-ground nitrogen loading and recharge data, Environ. Modell. Softw. 20 (2005) 851–871.
[9] P. Avesani, A. Perini, F. Ricci, Interactive case-based planning for forest fire management, Appl. Intell. 13 (2000) 41–57.
[10] S. Bandini, G. Pavesi, Simulation of vegetable population dynamics based on Cellular Automata, in: Proceedings of 5th International Conference on Cellular Automata for Research and Industry, ACRI, Geneva, Switzerland, 2002, pp. 202–209.
[11] A. Bardossy, A. Bronstert, B. Merz, 1-, 2- and 3-dimensional modelling of water movement in the unsaturated soil matrix using a fuzzy approach, Adv. Water Resour. 18 (1995) 237–251.
[12] R. Barletta, An introduction to case-based reasoning, AI Expert 6 (1991) 42–49.
[13] O. Barreteau, F. Bousquet, C. Millier, J. Weber, Suitability of Multi-Agent Simulations to study irrigated system viability: application to case studies in the Senegal River Valley, Agric. Syst. 80 (2004) 255–275.
[14] D. Batten, Are some human ecosystems self-defeating? Environ. Modell. Softw., in press.
[15] B. Bazartseren, G. Hildebrandt, K.P. Holz, Short-term water level prediction using neural networks and neuro-fuzzy approach, Neurocomputing 55 (2003) 439–450.
[16] N. Becu, P. Perez, A. Walker, O. Barreteau, C. Le Page, Agent based simulation of a small catchment water management in northern Thailand. Description of the CATCHSCAPE model, Ecol. Model. 170 (2003) 319–331.
[17] T. Berger, Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis, Agric. Econ. 25 (2001) 245–260.
[18] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, New York, 1999.
[19] C. Bone, S. Dragicevic, A. Roberts, A fuzzy-constrained cellular automata model of forest insect infestations, Ecol. Model. 192 (2006) 107–125.
[20] D. Borri, D. Camarda, A. De Liddo, Mobility in environmental planning: an integrated multi-agent approach, Lect. Notes Comput. Sci. 3675 (2005) 119–129.
[21] Borsuk, Environ. Modell. Softw., in preparation.
[22] F. Bousquet, C. Le Page, Multi-agent simulations and ecosystem management: a review, Ecol. Model. 176 (2004) 313–332.
[23] I. Brown, Modelling future landscape change on coastal floodplains using a rule-based GIS, Environ. Modell. Softw. 21 (2006) 1479–1490.
[24] B.P. Buckeles, F.E. Petry, Genetic Algorithms, IEEE Computer Society Press, Los Alamitos, CA, 1992.
[25] B. Bullnheimer, R.F. Hartl, C. Strauss, An improved ant system algorithm for the vehicle routing problem, Ann. Oper. Res. 89 (1999) 319–328.
[26] A. Bundy (Ed.), Artificial Intelligence Techniques: A Comprehensive Catalogue, fourth ed., Springer-Verlag, New York, 1997.
[27] L. Ceccaroni, U. Cortes, M. Sanchez-Marre, OntoWEDSS: augmenting environmental decision-support systems with ontologies, Environ. Modell. Softw. 19 (2004) 785–797.
[28] C.L. Chang, S.L. Lo, S.L. Yu, Applying fuzzy theory and genetic algorithm to interpolate precipitation, J. Hydrol. 314 (2005) 92–104.
[29] K.W. Chau, A split-step PSO algorithm in prediction of water quality pollution, Lect. Notes Comput. Sci. 3498 (2005) 1034–1039.
[30] K.W. Chau, Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River, J. Hydrol. 329 (2006) 363–367.
[31] Q. Chen, A.E. Mynett, Modelling algal blooms in the Dutch coastal waters by integrated numerical and fuzzy cellular automata approaches, Ecol. Model. 199 (2006) 73–81.
[32] C.T. Cheng, C.P. Ou, K.W. Chau, Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration, J. Hydrol. 268 (2002) 72–86.
[33] H.K. Cigizoglu, M. Alp, Generalized regression neural network in modelling river sediment yield, Adv. Eng. Softw. 37 (2006) 63–68.
[34] E.F. Codd, Cellular Automata, ACM Monograph Series, Academic Press, New York, 1968.
[35] J.P. Coelho, P.B. de Moura Oliveira, J. Boaventura Cunha, Greenhouse air temperature predictive control using the Particle Swarm Optimisation algorithm, Comput. Electron. Agric. 49 (2005) 330–344.

[36] O. Cordón, I. Fernández de Viana, F. Herrera, Analysis of the Best-Worst Ant Systems and its variants on the QAP, Lect. Notes Comput. Sci. 2463 (2002) 228–234.

[37] G.M. Crisci, S. Di Gregorio, F. Nicoletta, R. Rongo, W. Spataro, Analysing lava risk for the Etnean area: simulation by cellular automata methods, Nat. Hazards 20 (1999) 215–229.

[38] A.M. Crowe, C.J. McClean, M.S. Cresser, An application of genetic algorithms to the robust estimation of soil organic and mineral fraction densities, Environ. Modell. Softw. 21 (2006) 1503–1507.

[39] D. D'Ambrosio, S. Di Gregorio, G. Iovine, V. Lupiano, R. Rongo, W. Spataro, Modelling surface flows for macroscopic phenomena by cellular automata: an application to debris flows, Lect. Notes Comput. Sci. 2493 (2002) 304–314.

[40] B. Denby, S. Le Hégarat-Mascleb, Swarm intelligence in optimisation problems, Nucl. Instrum. Meth. A 502 (2003) 364–368.

[41] V. Dhar, R. Stein, Intelligent Decision Support Methods. The Science of Knowledge Work, Prentice Hall, New Jersey, 1997.

[42] G. Di Caro, M. Dorigo, AntNet: distributed stigmergetic control for communications networks, J. Artif. Intell. Res. 9 (1998) 317–365.

[43] B. Dixon, Applicability of neuro-fuzzy techniques in predicting ground-water vulnerability: a GIS-based sensitivity analysis, J. Hydrol. 309 (2005) 17–38.

[44] A.V. Donati, R. Montemanni, N. Casagrande, A.E. Rizzoli, L.M. Gambardella, Time dependent vehicle routing problem with a multi ant colony system, Eur. J. Oper. Res., in press.

[45] M. Dorigo, L.M. Gambardella, Ant Colony System: a cooperative learning approach to the Traveling Salesman Problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66.

[46] M. Dorigo, E. Bonabeau, G. Theraulaz, Ant algorithms and stigmergy, Future Gener. Comp. Syst. 16 (2000) 851–871.

[47] M. Dorigo, V. Maniezzo, A. Colorni, Ant System: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1996) 29–41.

[48] C. Dou, W. Woldt, I. Bogardi, Fuzzy rule based approach to describe solute transport in the unsaturated zone, J. Hydrol. 220 (1999) 74–85.

[49] G. Dounias, Hybrid Computational Intelligence in Medicine, http://www.cs.queensu.ca/home/cisc875/Dounias_paper.pdf, 2003.

[50] M.J. Dreyfus-Leon, Individual-based modelling of fishermen search behaviour with neural networks and reinforcement learning, Ecol. Model. 120 (1999) 287–297.

[51] M. Droz, A. Pękalski, Dynamics of populations in extended systems, Lect. Notes Comput. Sci. 2493 (2002) 190–201.

[52] E. El-Azhary, H.A. Hassan, A. Rafea, Pest control expert system for tomato (PCEST), Knowl. Inform. Syst. 2 (2000) 242–257.

[53] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Adv. Eng. Inform. 19 (2005) 43–53.

[54] S. El Yacoubi, A. El Jai, Cellular automata modelling and spreadability, Math. Comput. Model. 36 (2002) 1059–1074.

[55] S. El Yacoubi, A. El Jai, P. Jacewicz, J.G. Pausas, LUCAS: an original tool for landscape modelling, Environ. Modell. Softw. 18 (2003) 429–437.

[56] R.S. Engelmore, E. Feigenbaum, Expert systems and artificial intelligence, in: Knowledge-Based Systems in Japan, http://www.wtec.org/loyola/kb/, 1993.

[57] F. Fdez-Riverola, J.M. Corchado, Improved CBR system for biological forecasting, EOAI, Workshop 23, Binding Environmental Sciences and Artificial Intelligence, Valencia, Spain, 2004.

[58] G. Fila, M. Donatelli, G. Bellocchi, PTFIndicator: an IRENE DLL-based application to evaluate estimates from pedotransfer functions by integrated indices, Environ. Modell. Softw. 21 (2006) 107–110.

[59] G. Fleming, M. van der Merwe, G. McFerren, Fuzzy expert systems and GIS for cholera health risk prediction in southern Africa, Environ. Modell. Softw. 22 (2007) 442–448.

[60] R.A. Flores-Mendez, Towards a standardization of multi-agent system frameworks, ACM Crossroads 5, http://www.acm.org/crossroads/xrds5-4/multiagent.html, 1999.

[61] G. Folino, G. Mendicino, A. Senatore, G. Spezzano, S. Straface, A model based on cellular automata for the parallel simulation of 3D unsaturated flow, Parallel Comput. 32 (2006) 357–376.

[62] M.A. Fonstad, Cellular automata as analysis and synthesis engines at the geomorphology–ecology interface, Geomorphology 77 (2006) 217–234.

[63] S. Forrest, Genetic Algorithms: principles of natural selection applied to computation, Science 261 (1993) 872–878.

[64] R. Fuller, Introduction to Neuro-Fuzzy Systems, Physica-Verlag Heidelberg, New York, 2000.

[65] I.G. Georgoudas, G.C. Sirakoulis, E.M. Scordilis, I. Andreadis, A cellular automaton simulation tool for modelling seismicity in the region of Xanthi, Environ. Modell. Softw., in press.

[66] M.K. Gill, Y.H. Kaheil, A. Khalil, M. McKee, L. Bastidas, Multiobjective particle swarm optimization for parameter estimation in hydrology, Water Resour. Res. 42 (2006) W07417.

[67] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Co., Reading, MA, 1989.

[68] E. Goles, S. Martinez, Neural and Automata Networks—Dynamic Behavior and Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.

[69] A. Gray, R. Kilgour, Frequently Asked Questions: Hybrid Systems, http://www.cecs.missouri.edu/~rsun/hybrid-FAQ.html, 1997.

[70] J.E. Gross, R.R.J. McAllister, N. Abel, D.M. Stafford Smith, Y. Maru, Australian rangelands as complex adaptive systems: a conceptual model and preliminary results, Environ. Modell. Softw. 21 (2006) 1264–1272.

[71] D. Hammerstrom, Working with neural networks, IEEE Spectrum (July) (1993) 46–53.

[72] M. Hare, P. Deadman, Further towards a taxonomy of agent-based simulation models in environmental management, Math. Comput. Simul. 64 (2004) 25–40.

[73] M.E. Harmon, S.S. Harmon, Reinforcement Learning: A Tutorial, http://www.nbu.bg/cogs/events/2000/Readings/Petrov/rltutorial.pdf, 2000.

[74] F. Hayes-Roth, Rule-based systems, Commun. ACM 28 (1985) 921–932.

[75] L. Hernandez Encinas, S. Hoya White, A. Martin del Rey, G. Rodriguez Sanchez, Modelling forest fire spread using hexagonal cellular automata, Appl. Math. Model., in press.

[76] J.H. Holland, Genetic Algorithms, http://www.econ.iastate.edu/tesfatsi/holland.GAIntro.htm, 2005.

[77] S.H. Huang, H.C. Zhang, Neural-expert hybrid approach for intelligent manufacturing: a survey, Comput. Ind. 26 (1995) 107–126.

[78] L.S. Iliadis, A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation, Environ. Modell. Softw. 20 (2005) 613–621.

[79] L.S. Iliadis, F. Maris, An artificial neural network model for mountainous water-resources management: the case of Cyprus mountainous watersheds, Environ. Modell. Softw., in press.

[80] A.V.M. Ines, K. Honda, A. Das Gupta, P. Droogers, R.S. Clemente, Combining remote sensing-simulation modeling and genetic algorithm optimization to explore water management options in irrigated agriculture, Agric. Water Manage. 83 (2006) 221–232.

[81] R.M. Itami, Simulating spatial dynamics: cellular automata theory, Landscape Urban Plan. 30 (1994) 27–47.

[82] A. Jain, A. Kumar, An evaluation of artificial neural network technique for the determination of infiltration model parameters, Appl. Soft Comput. 6 (2006) 272–282.

[83] A.K. Jain, J. Mao, K. Mohiuddin, Artificial Neural Networks: a tutorial, IEEE Comput. 29 (1996) 31–44.

[84] R. Janssen, H. Goosen, M.L. Verhoeven, J.T.A. Verhoeven, A.Q.A. Omtzigt, E. Maltby, Decision support for integrated wetland management, Environ. Modell. Softw. 20 (2005) 215–229.

[85] L. Kaelbling, M. Littman, A. Moore, Reinforcement learning: a survey, J. Artif. Intell. Res. 4 (1996) 237–285.

[86] E. Kalapanidas, N. Avouris, Short-term air quality prediction using a case-based classifier, Environ. Modell. Softw. 16 (2001) 263–272.

[87] E. Kalapanidas, N. Avouris, Feature selection for air quality forecasting: a genetic algorithm approach, AI Commun. 16 (2003) 235–251.

[88] F. Karaca, B. Ozkaya, NN-LEAP: a neural network-based model for controlling leachate flow-rate in a municipal solid waste landfill site, Environ. Modell. Softw. 21 (2006) 1190–1197.

[89] I. Karafyllidis, A. Thanailakis, A model for predicting forest fire spreading using cellular automata, Ecol. Model. 99 (1997) 87–97.

[90] D.S. Kaster, C.B. Medeiros, H.V. Rocha, Supporting modelling and problem solving from precedent experiences: the role of workflows and case-based reasoning, Environ. Modell. Softw. 20 (2005) 689–704.

[91] S. Kawano, V.N. Huynh, M. Ryoke, Y. Nakamori, A context-dependent knowledge model for evaluation of regional environment, Environ. Modell. Softw. 20 (2005) 343–352.

[92] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings from IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.

[93] I. Keramitsoglou, C. Cartalis, C.T. Tiranoudis, Automatic identification of oil spills on satellite images, Environ. Modell. Softw. 21 (2006) 640–652.

[94] L.P. Khoo, M.Y. Loi, A Tabu-Enhanced Genetic Algorithm approach to Agile manufacturing, Int. J. Adv. Manuf. Tech. 20 (2002) 692–700.

[95] G. Kim, A.P. Barros, Quantitative flood forecasting using multisensor data and neural networks, J. Hydrol. 246 (2001) 45–62.

[96] J.M.P. King, R. Banares-Alcantara, Z.A. Manan, Minimising environmental impact using CBR: an azeotropic distillation case study, Environ. Modell. Softw. 14 (1999) 359–366.

[97] G.B. Kingston, M.F. Lambert, H.R. Maier, Development of stochastic Artificial Neural Networks for hydrological prediction, in: Proceedings for Modsim, International Congress on Modelling and Simulation, Townsville, Queensland, 2003, pp. 837–842.

[98] T. Koutroumanidis, L. Iliadis, G.K. Sylaios, Time-series modelling of fishery landings using ARIMA models and Fuzzy Expected Intervals software, Environ. Modell. Softw. 21 (2006) 1711–1721.

[99] J.T. Kuo, Y.Y. Wang, W.S. Lung, A hybrid neural-genetic algorithm for reservoir water quality management, Water Res. 40 (2006) 1367–1376.

[100] B.H. Lee, M. Scholz, A. Horn, Constructed wetlands: treatment of concentrated storm water runoff (Part A), Environ. Eng. Sci. 23 (2006) 320–331.

[101] B.H. Lee, M. Scholz, A. Horn, A.M. Furber, Constructed Wetlands: prediction of performance with case-based reasoning (Part B), Environ. Eng. Sci. 23 (2006) 332–340.

[102] V.R. Lesser, Multiagent systems: an emerging subdiscipline of AI, ACM Comput. Surv. 27 (1995) 340–342.

[103] Y. Li, A.B. Chan Hilton, Optimal groundwater monitoring design using an ant colony optimization paradigm, Environ. Modell. Softw. 22 (2007) 110–116.

[104] W. Loibl, T. Toetzer, Modeling growth and densification processes in suburban regions—simulation of landscape transition with spatial agents, Environ. Modell. Softw. 18 (2003) 553–563.

[105] W.Z. Lu, H.Y. Fan, S.M. Lo, Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong, Neurocomputing 51 (2003) 387–400.

[106] B.D. Mahaman, P. Harizanis, I. Filis, E. Antonopoulou, C.P. Yialouris, A.B. Sideridis, A diagnostic expert system for honeybee pests, Comput. Electron. Agric. 36 (2002) 17–31.

[107] B.D. Mahaman, H.C. Passam, A.B. Sideridis, C.P. Yialouris, DIARES-IPM: a diagnostic advisory rule-based expert system for integrated pest management in Solanaceous crop systems, Agric. Syst. 76 (2003) 1119–1135.

[108] H.R. Maier, G.C. Dandy, M.D. Burch, Use of artificial neural networks for modelling cyanobacteria *Anabaena* spp. in the River Murray, South Australia, Ecol. Model. 105 (1998) 257–272.

[109] H.R. Maier, A.R. Simpson, A.C. Zecchin, W.K. Foong, K.Y. Phang, H.Y. Seah, C.L. Tan, Ant colony optimization for design of water distribution systems, J. Water Resour. Plan. Manage. 129 (2003) 200–209.

[110] H.R. Maier, A.C. Zecchin, L. Radbone, P. Goonan, Optimising the mutual information of ecological data clusters using evolutionary algorithms, Math. Comput. Model. 44 (2006) 439–450.

[111] A. Marchini, A.C. Marchini, A fuzzy logic model to recognise ecological sectors in the lagoon of Venice based on the benthic community, Ecol. Model. 193 (2006) 105–193.

[112] L.S. Matott, A.J. Rabideau, J.R. Craig, Pump-and-treat optimization using analytic element method flow models, Adv. Water Resour. 29 (2006) 760–775.

[113] G. Metternicht, S. Gonzalez, FUERO: foundations of a fuzzy exploratory model for soil erosion hazard prediction, Environ. Modell. Softw. 20 (2005) 715–728.

[114] J. Molofsky, D. Bever, A new kind of ecology? BioScience 54 (2004) 440–446.

[115] M. Monticino, M. Acevedo, B. Callicott, T. Cogdill, C. Lindquist, Coupled human and natural systems: a multi-agent-based approach, Environ. Modell. Softw., in press.

[116] N. Moreno, R. Quintero, M. Ablan, R. Barros, J. Davila, H. Ramirez, G. Tonella, M.F. Acevedo, Biocomplexity of deforestation in the Caparo tropical forest reserve in Venezuela: an integrated multi-agent and cellular automata model, Environ. Modell. Softw., in press.

[117] C.V. Negoita, Expert Systems and Fuzzy Systems, Benjamin/Cummings Publishing Co., California, 1985.

[118] K.C. Ng, B. Abramson, Uncertainty management in expert systems, IEEE Intell. Syst. Appl. 5 (1990) 29–47.

[119] S. Nissen, Implementation of a Fast Artificial Neural Network Library (FANN), http://fann.sourceforge.net/report/report.html, 2003.

[120] Y. Niv, D. Joel, I. Meilijson, E. Ruppin, Evolution of reinforcement learning in foraging bees: a simple explanation for risk averse behaviour, Neurocomputing 44–46 (2002) 951–956.

[121] H. Nunez, M. Sanchez-Marre, U. Cortes, J. Comas, M. Martinez, I. Rodriguez-Roda, M. Poch, A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situation, Environ. Modell. Softw. 9 (2004) 809–819.

[122] G. Onkal-Engin, I. Demir, S.N. Engin, Determination of the relationship between sewage odour and BOD by neural networks, Environ. Modell. Softw. 20 (2005) 843–850.

[123] C. Pahl-Wostl, Information, public empowerment, and the management of urban watersheds, Environ. Modell. Softw. 20 (2005) 457–467.

[124] D.C. Parker, S.M. Manson, M.A. Janssen, M.J. Hoffmann, P. Deadman, Multi-agent systems for the simulation of land-use and land-cover change: a review, Ann. Assoc. Am. Geogr. 93 (2003) 314–337.

[125] L. Parrott, R. Lacroix, K.M. Wade, Design considerations for the implementation of multi-agent systems in the dairy industry, Comput. Electron. Agric. 38 (2003) 79–98.

[126] S. Passone, P.W.H. Chung, V. Nassehi, Case-based reasoning for estuarine model design, Lect. Notes Artif. Int. 2416 (2002) 590–603.

[127] G.J. Pelletier, S.C. Chapra, H. Tao, QUAL2Kw—a framework for modelling water quality in streams and rivers using a genetic algorithm for calibration, Environ. Modell. Softw. 21 (2006) 419–425.

[128] R. Pongracz, I. Bogardi, L. Duckstein, Application of fuzzy rule-based modelling technique to regional drought, J. Hydrol. 224 (1999) 100–114.

[129] D. Pozdnyakov, R. Shuchman, A. Korosov, C. Hatt, Operational algorithm for the retrieval of water quality in the Great Lakes, Remote Sens. Environ. 97 (2005) 352–370.

[130] H. Purnomo, G.A. Mendoza, R. Prabhu, Y. Yasmi, Developing multi-stakeholder forest management scenarios: a multi-agent system simulation approach applied in Indonesia, For. Policy Econ. 7 (2005) 475–491.

[131] D.W. Roberts, Modelling forest dynamics with vital attributes and fuzzy systems theory, Ecol. Model. 90 (1996) 161–173.

[132] I.R. Roda, M. Sanchez-Marre, J. Comas, U. Cortes, M. Poch, Development of a case-based system for the supervision of an activated sludge process, Environ. Technol. 22 (2001) 477–486.

[133] D.M. Rodvold, D.G. McLeod, J.M. Brandt, P.B. Snow, G.P. Murphy, Introduction to artificial neural networks: taking the lid off the black box, Prostate 46 (2001) 39–44.

[134] S.A. Rounds, Development of a neural network model for dissolved oxygen in the Tualatin River, Oregon, in: Proceedings of the Second Federal Interagency Hydrologic Modeling Conference, Las Vegas, NV, http://smig.usgs.gov/SMIG/features_0902/tualatin_ann.html, 2002.

[135] R. Sadiq, T. Husain, A fuzzy-based methodology for an aggregative environmental risk assessment: a case study of drilling waste, Environ. Modell. Softw. 20 (2005) 33–46.

[136] A.B. Salem, Intelligent Learning Systems: A Case-Based Reasoning Approach, http://rilw.itim-cj.ro/2000/tutorials/salem_tut.html, 2000.

[137] M. Sanchez-Marre, U. Cortes, I.R. Roda, M. Poch, Sustainable case learning for continuous domains, Environ. Modell. Softw. 14 (1999) 349–357.

[138] M. Sanchez-Marre, U. Cortes, I.R. Roda, M. Poch, J. Lafuente, Learning and adaptation in wastewater treatment plants through case-based reasoning, Microcomp. Civil Eng. 12 (1997) 251–266.

[139] J. San Pedro, F. Burstein, A. Sharp, A case-based fuzzy multicriteria decision support model for tropical cyclone forecasting, Eur. J. Oper. Res. 160 (2005) 308–324.

[140] M.S. Santiago Barros, V. Rodrigues, Nonlinear aspects of data integration for land-cover classification in a neural network environment, Adv. Space Res. 14 (1994) 265–268.

[141] M. Schluter, N. Ruger, Application of a GIS-based simulation tool to illustrate implications of uncertainties for water management in the Amudarya river delta, Environ. Modell. Softw. 22 (2007) 158–166.

[142] C. Schmid, Course on Dynamics of Multidisplicinary and Controlled Systems, http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/main.html, 2005.

[143] B. Schönfisch, M. Kinder, A fish migration model, Lect. Notes Comput. Sci. 2493 (2002) 210–219.

[144] P.S. Shelokar, P. Siarry, V.K. Jayaraman, B.D. Kulkarni, Particle swarm and ant colony algorithms hybridized for improved continuous optimization, Appl. Math. Comput., in press.

[145] T. Slini, A. Kaprara, K. Karatzas, N. Moussiopoulos, PM_{10} forecasting for Thessaloniki, Greece, Environ. Modell. Softw. 21 (2006) 559–565.

[146] K. Socha, J. Knowles, M. Samples, A Max-Min Ant System for the university course timetabling problem, Lect. Notes Comput. Sci. 2463 (2002) 1–13.

[147] S.I.V. Sousa, F.G. Martins, M.C.M. Alvim-Ferraz, M.C. Pereira, Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations, Environ. Modell. Softw. 22 (2007) 97–103.

[148] J. Spate, A.J. Jakeman, Environ. Modell. Softw., submitted for publication.

[149] S. Srinivasula, A. Jain, A comparative analysis of training methods for artificial neural network rainfall-runoff models, Appl. Soft Comput. 6 (2006) 295–306.

[150] W.-H. Steeb, The Nonlinear Workbook, World Scientific Publishing, Singapore, 1999.

[151] C. Stergiou, D. Siganos, Neural Networks, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html, 1996.

[152] D. Stevens, S. Dragicevic, K. Rothley, iCity: a GIS-CA modelling tool for urban planning and decision making, Environ. Modell. Softw., in press.

[153] D.R.B. Stockwell, J.H. Beach, A. Stewart, G. Vorontsov, D. Vieglais, R.S. Pereira, The use of the GARP genetic algorithm and Internet grid computing in the Lifemapper world atlas of species biodiversity, Ecol. Model. 195 (2002) 139–145.

[154] T. Stützle, H.H. Hoos, MAX-MIN Ant System, Future Gener. Comp. Syst. 16 (2000) 889–914.

[155] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, http://www.cs.ualberta.ca/%7Esutton/book/ebook/the-book.html, 1998.

[156] K.P. Sycara, Multiagent systems, AI Mag. 19 (2) (1998) 79–92.

[157] T. Takeuchi, Y. Yazu, A. Sakuma, Integrated environmental management process applying genetic algorithm, Int. J. Prod. Econ. 60/61 (1999) 229–234.

[158] R.R. Tan, Rule-based life cycle impact assessment using modified rough set induction methodology, Environ. Modell. Softw. 20 (2005) 509–513.

[159] R.R. Tan, Hybrid evolutionary computation for the development of pollution prevention and control strategies, J. Cleaner Prod., in press.

[160] M. Termansen, C.J. McClean, C.D. Preston, The use of genetic algorithms and Bayesian classification to model species distributions, Ecol. Model. 192 (2006) 410–424.

[161] A. Tsoularis, J. Wallace, Reinforcement learning for a stochastic automaton modelling predation in stationary model-mimic environments, Math. Biosci. 195 (2005) 76–91.

[162] E. Van Broekhoven, V. Adriaenssens, B. De Baets, P.F.M. Verdonschot, Fuzzy rule-based macroinvertebrate habitat suitability models for running waters, Ecol. Model. 198 (2006) 71–84.

[163] A. Vellido, E. Marti, J. Comas, I. Rodriguez-Roda, F. Sabater, Exploring the ecological status of human altered streams through Generative Topographic Mapping, Environ. Modell. Softw., in press.

[164] F. Verdenius, J. Broeze, Generalised and instance-specific modelling for biological systems, Environ. Modell. Softw. 14 (1999) 339–348.

[165] Q.J. Wang, Using genetic algorithms to optimise model parameters, Environ. Modell. Softw. 12 (1997) 27–34.

[166] I. Watson, F. Marir, Case-Based, Reasoning: a review, Knowl. Eng. Rev. 9 (1994) 327–354.

[167] P.H. Winston, Artificial Intelligence, second ed., Addisin-Wesley Publishing Co., Reading, MA, 1984.

[168] S. Wolfram, Universality and complexity in cellular automata, Physica D 10 (1984) 1–35.

[169] L. Xiong, A.Y. Shamseldin, K.M. O'Connor, A non-linear combination of the forecasts of rainfall-runoff models by the first-order Takagi-Sugeno fuzzy system, J. Hydrol. 245 (2001) 196–217.

[170] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.

[171] P.Y. Yin, J.Y. Wang, Ant colony optimization for the nonlinear resource allocation problem, Appl. Math. Comput. 174 (2006) 1438–1453.

[172] P.Y. Yin, J.Y. Wang, A particle swarm optimization approach to the nonlinear resource allocation problem, Appl. Math. Comput., in press.

[173] M. Yong, P. Yong-Zhen, W. Xiao-Lian, W. Shu-Ying, Intelligent control aeration and external carbon addition for improving nitrogen removal, Environ. Modell. Softw. 21 (2006) 821–828.

[174] S.H. Zahiri, S.A. Seyedin, Swarm intelligence based classifiers, J. Franklin Inst., in press.

[175] A.C. Zecchin, A.R. Simpson, H.R. Maier, M. Leonard, A.J. Roberts, M.J. Berrisford, Application of two Ant Colony Optimisation algorithms to water distribution system optimisation, Math. Comput. Model. 44 (2006) 451–468.

[176] A.C. Zecchin, A.R. Simpson, H.R. Maier, J.B. Nixon, Parametric study for an Ant Algorithm applied to water distribution system optimization, IEEE Trans. Evol. Comput. 9 (2005) 175–190.

[177] F. Zetian, X. Feng, Z. Yun, Z. XiaoShuan, Pig-vet: a web-based expert system for pig disease diagnosis, Expert Syst. Appl. 29 (2005) 93–103.

[178] Q. Zhang, S.J. Stanley, Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modelling, Water Res. 31 (1997) 2340–2350.

[179] H. Zhang, H. Li, C.M. Tam, Particle swarm optimization for resource-constrained project scheduling, Int. J. Proj. Manage. 24 (2006) 83–92.