CI TECHNOLOGIES

CITS4404
Artificial Intelligence & Adaptive Systems
Key technologies

• Evolutionary algorithms
• Particle swarm optimisation
• Ant colony optimisation
• Artificial neural networks
• Learning classifier systems
• Fuzzy reasoning
• Market-based algorithms
• Bayesian reasoning
• Artificial immune systems
Evolutionary algorithms

- Based on the principle of evolution by natural selection
- The algorithm maintains a population of **encodings**
  - The structure of an encoding captures what the algorithm is allowed to vary in its search for a good solution
- Each encoding represents a solution
- Each solution has a corresponding **fitness** that describes how good that solution is
- In each **generation**
  - The fitnesses are used to decide which solutions **survive**
  - The survivors become **parents** and they spawn new encodings, generated via **mutation** and **crossover**
  - The **children** also represent solutions with fitnesses…
EAs contd.

- Note that an encoding/solution, once created, never changes
  - Its descendants will be different
- The general idea is that good solutions generate “similar” solutions, some of which may be an improvement
  - Parents generate children either singly or in combination
- The initial population is generated either randomly, or using some domain knowledge
- Termination can be determined in several ways
  - A fixed number of generations
  - Until improvement ceases
  - Until a certain fitness is obtained
Particle swarm optimisation

- Based on the behaviour of a flock of birds searching for food
- The algorithm maintains a population of particles
- Each particle moves continually over the landscape
  - At any moment, each particle has a position, representing a solution; and a velocity, representing its momentum
- Each particle remembers the best solution it has ever seen, its personal best pbest
  - The algorithm also remembers the global best gbest
- In each generation
  - Each particle’s velocity is updated, favouring pbest and gbest
  - Each particle’s position is updated using its new velocity
  - The pbests and gbest are updated as appropriate
• Population dimension = 4
• Delay between iterations = 500
PSO contd.

• Each particle explores different solutions in different generations
  • Collectively the swarm explores the landscape
• The updating mechanisms mean that particles favour areas of the landscape known to have good solutions
  • Good solutions the particle has seen
  • Good solutions other particles have seen
• As time proceeds, the swarm focuses on a smaller and smaller area
  • Eventually, the swarm will converge on the area surrounding $gbest$
Ant colony optimisation

• Based on groups of ants communicating via pheromones
• Given a problem structured as a network, the algorithm maintains a population of ants that traverse the network
• An ant selects each step through the network probabilistically
  • It will favour “good” steps
  • It will favour steps with more pheromone
• When an ant completes a traversal, it lays pheromone on the path that it used
  • The amount of pheromone laid will be proportional to the quality of the path
• Pheromone evaporates over time, to allow for adaptation in the steps selected
ACO contd.

• The key points are that
  • When one ant discovers something good, every ant benefits
  • Initially-random choices improve over time
• ACO applies naturally to problems involving spatial networks
  • Travelling salesman
  • Vehicle routing
  • Electronic messaging
  • etc.
• But many other problems can be cast as networks
  • Scheduling
  • Timetabling
  • Image processing
  • etc.
Artificial neural networks

- Based on the structure of the brain and its processing ability
- ANNs act mainly as
  - Function approximators
  - Pattern recognisers
- An ANN is composed of one or more layers of neurons
  - Each neuron is very simple
  - Power and intelligence emerges from their (usually vast!) numbers, and from the interconnections between them
- Data is fed into one end of the network (the input layer), it passes through the hidden layers of the network, and it emerges from the output layer
  - The various layers generate progressively higher-level information
\[ a_i = g \left( \sum_{j=0}^{n} W_{j, i} \cdot a_j \right) \]
ANNs contd.

- The number of hidden layers required is determined by the complexity of the problem being solved
  - Zero hidden layers – can represent only linearly-separable functions
  - One hidden layer – can represent any continuous function
  - Multiple hidden layers – can represent any function

- ANNs can be
  - Acyclic (feed-forward) – stateless processing
  - Cyclic (recurrent) – supports short-term memory

- ANNs learn by fine-tuning the weights on their links, usually by one of two mechanisms
  - Back-propagation
  - Evolution or similar
Learning classifier systems

- Based on how “experts” solve problems and acquire skills
- The algorithm maintains a database of “condition-action-prediction” rules
  - The condition defines when the rule applies
  - The action states what the system should do
  - The prediction indicates the expected reward
- Given a problem instance, the algorithm
  - Forms a match set of rules whose conditions are satisfied
  - Chooses the action $A$ with the best predicted performance
  - Forms the action set of rules that recommend $A$
  - Executes $A$ and observes the actual performance, which is fed back to update the action set
updated periodically by evolution, covering, and subsumption

(Diagram adapted from a seminar on using LCSs for fraud detection, by M. Behdad)
LCS contd.

- As well as direct feedback, the rule set is periodically updated
  - By subsumption, generalisation, and covering
  - By evolution or similar
- Feedback can be based on either
  - The performance obtained by using the action
  - The accuracy of a rule’s prediction
- Sometimes the database is divided into semi-permanent “teams” of rules that are known to work well together
Fuzzy reasoning

• Based on human processing of noisy/imprecise/partial data

• Two key concepts
  • **Granulation**: everything is “clumped”, e.g. a person can be “young”, or “middle-aged”, or “old”
  • **Graduation**: everything is a matter of degree, e.g. a day can be “not cold”, or “a bit cold”, or “a lot cold”, or …

• Instead of saying that a state is either “cold” or “not cold”, we assign a degree of truth: e.g. a state is “0.8 cold”

• Operators are changed accordingly, e.g.
  • \( v(\text{not}(p)) = 1 - v(p) \)
  • \( v(p \text{ and } q) = \min \{v(p), v(q)\} \)
  • There are several alternative formulations for **and**
Fuzzy contd.

- A fuzzy control system is a collection of rules
  - IF X [AND Y] THEN Z
  - e.g. IF cold AND not warming-up THEN increase heating slightly
- These rules attempt to mimic human-style logic
- Granulation means that the exact values of constants are unimportant

- In each cycle, the system
  - Takes a set of observations and fuzzifies them
  - Applies all of the rules that match, generating a set of fuzzy results
  - Defuzzifies the results to get a precise output
Example from http://www.faqs.org/docs/fuzzy/

- temp is 0.48 cool
- pressure is 0.57 low and 0.25 ok
- Rule 2 gives 0.48 P2
- Rule 3 gives 0.25 Z
Market-based algorithms

• Based on the greedy operation of trading markets
Bayesian reasoning

- Based on probabilistic reasoning with learning
Artificial immune systems

- Based on the learning mechanisms of body-defense systems