Adaptation

Adaptive agent

An agent that can respond to its environment.

Four primary ways of adapting:

- **reaction** - a direct, predetermined response to a particular event or an environmental signal.
- **reasoning** - ability to make inferences.
- **learning** - change that occurs during the lifetime of an agent.
- **evolution** - change in a population that occurs over successive generations of agents.
Adapting by reaction

A direct, predetermined response to a particular event or environmental signal.

Typically expressed in the form

"WHEN event, IF condition(s), THEN action"

Examples:

- thermostats
- robotic sensors that can detect the presence of a nearby wall and activate a device for avoiding it
- washing machines and vacuum cleaners that use fuzzy logic

Adapting by reasoning

A reactive response that uses inference rules.

A more advanced form of reactive adaptation using a set of rules to infer.

Typically chains of rules in the form—

"WHEN event, IF condition(s), THEN action"

Typical kinds of techniques:

- patient diagnosis
- bulletin board or web foraging agents
- data mining
Adapting by learning

Typical kinds of techniques:
- credit assignment
- Bayesian (or probabilistic) rules
- neural networks
- classifier rules
- exploration of internal representations
- Q learning

Adapting by evolution

Typical kinds of strategies:
- Natural & artificial selection (i.e., “survival of the fittest”)
- Darwinian vs. Lamarckian evolution (e.g., genotype and phenotype)
- Differentiation into ecosystem roles and community formation
- Competition (e.g., increasing returns)
- Cooperation (e.g., multiagent composition)
- Coevolution & arms races
- Cultural transmission (e.g., Richard Dawkins’ “memes”, the Baldwin effect)
Four primary ways of adapting

- Reaction (minimum requirement)
- Reasoning
- Learning
- Evolution

Any or all in combination (e.g. learning Bayesian inference webs using memes and genes...)

Some common adaptive learning and evolution methods

- Neural networks
- Genetic algorithms
- Genetic programming
- Ant colony optimization

(all in insufficient detail...)
Neural networks

A network algorithm that identifies patterns using neuron-based techniques.

A Single Processing Element

Many inputs

Connection Weights

Input1 = I1 x W1
Input2 = I2 x W2
...\nInputn = In x Wn

Sum = Input1 + Input2 + ... + Inputn

If Sum > Threshold then PE "fires"

Output = (sum of Inputs x weights)

Insurane company example

Inputs

Input Layer

Hidden Layers

Output Layer

Outputs

Previous Medical History

Age

Occupation

Lifestyle (smoking, drinking, other recreational drug use, etc.)

Connection weights are adjusted - network output produces low error

Low Risk (Accept)

High Risk (Reject or refer to expert)
Neural networks

Some of the challenges:

- Steep learning curve
- No real development methodology
- Demanding preprocessing requirements
- Integration issues
- Interpretability of results

Genetic algorithms

Algorithms based on biogenetics and the principles of Darwinian evolution

GAs are parallel stochastic search algorithms that

- represent possible solutions as strings (chromosomes) comprised of input values (genes).
- create consecutive generations of solutions using operators inspired by real genetics—mainly crossover and mutation—in order to create new variation in the population.
- evaluate each solution (individual) based on fitness criteria determined by the specific problem being solved (the environment).
- use the fitness information to favor the best individuals (strings) when creating new generations.
Genetic algorithms

Two of the most common operations:
Crossover and mutation.

Others include operations such as
inversion, deletion, and duplication.

Example using steps in
genetic algorithm cycle
Genetic algorithms

Some of the challenges:

- Problem solving is concurrent (multiagent)
- Can result in counterintuitive solutions
- Unplanned results can be scary
- Is processing-intensive

Other evolutionary algorithms

- Evolution strategies (ES)
- Evolutionary programming (EP)
- GANNs & cellular encoding
- Evolvable hardware
- Genetic programming
Genetic programming

Algorithm production based on the principles of genetic algorithms.

GP is a population-based evolutionary algorithm in which

- the desired solution is a function or algorithm itself, not just a set of numerical values.
- a non-brittle language is used to represent individual prospective solutions in the population.
- crossover, mutation, and the other search operators occur between programs, not just swapping values but exchanging arbitrary snippets of code.

\[
\pi \div x_1 + x_1 \div \sqrt{x_1}
\]

\[
\text{IF}(\log x_1 \leq 2.5) \quad \text{THEN} \quad (x_1 \cdot \min(x_1, x_3)) \quad \text{ELSE} \quad (\infty)
\]
Genetic programming

\[ \frac{\pi}{x_1} + \min(x_1, x_3) \]

IF \(\log x_1 \leq 2.5\) THEN \((x_1 \cdot x_1^{x_2/3})\) ELSE \(\infty\)

Genetic programming

Familiar formula:

\[ ax^2 + bx + c = 0 \quad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

Given: \(T = \{A, B, C\}\) \(F = \{+, -, \ast, \%\%, \sqrt{\text{}}\}\)

(function set)

Generated code after 30 generations:

\[
\text{(LIST2 (\% (- (- (B (+ A A))) (\% (\sqrt{\%} (\ast (*) (- (\sqrt{\%} B))) (* (+ (- C A) A) (\% A B))) (* (\sqrt{\%} A)) (\% (\sqrt{\%}) (\% (- (\sqrt{\%} B) (\% B)))) (\% (+ (- (+ B -5)) (+ (* (- (- (+ B -5) 1) ((\% A B)) + (\% B)) (-2 C) (- B))) (\% A A)) (\% C -3))))))
\]

\[
(\% (\sqrt{\%} (+ + B -5) (+ (- (- (+ B -5) 1) (* (+ B 4)) (+ (* C A) (- (+ (\% (- B B)) 4) -5) 1)) (\% (+ 1 (+ B B)) 4))) A))
\]
**Genetic programming**

Some of the challenges:

- Problem solving is concurrent (for multiagents)
- Can create novel, counterintuitive solutions
- Can result in uninterpretable structures—*even more easily than other evolutionary algorithms*
- Uninterpretable algorithms can be *very* scary
- Is *extraordinarily* processing-intensive

**Swarm Intelligence**

Technique of using the models of social insect behavior to design complex systems

This social insect analogy for problem solving has become a hot topic in the last few years.

- It is inspired by the mechanisms that generate collective behavior in insects.
- Individual behavior modifies the environment, which in turn modifies the behavior of other individuals.
- It emphasizes distributedness, indirect interaction among agents, flexibility, and **robustness**
Swarm intelligence methods

Experimental setup and drawings of the short brach selection of the *Linepithema humile* ant

---

Swarm intelligence methods

Network connection application

<table>
<thead>
<tr>
<th></th>
<th>Average call failures</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shortest path</td>
<td>12.53</td>
<td>2.04</td>
</tr>
<tr>
<td>Mobile agents</td>
<td>9.24</td>
<td>0.80</td>
</tr>
<tr>
<td>Improved mobile agents</td>
<td>4.41</td>
<td>0.85</td>
</tr>
<tr>
<td>ABC without noise</td>
<td>2.72</td>
<td>1.24</td>
</tr>
<tr>
<td>ABC with noise</td>
<td>2.56</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Interconnection structure of the British Telecom SDH network
Swarm intelligence methods

Traveling salesman application

An example of a trail configuration found in a 50-city problem

Swarm intelligence methods

Sorting application

An example of sorting seven kinds of items (800 items total)
Swarm intelligence methods

Pheromonal template application

When the pheromones of an Acantholepsis custodiens queen are detected, workers respond with template-based behavior.

---

Some of the challenges:

- Problem solving is concurrent and massively parallel
- Poor at finding novel, counterintuitive solutions
- Must hand-design internal ant dynamics and population size
- Mapping from ant algorithm to emergent dynamics is poorly understood