Master of Technology

Computational Intelligence II

Introduction to Computational Intelligence and Evolutionary Computation

Dr. Zhu Fangming
Institute of Systems Science,
National University of Singapore
Email: isszfm@nus.edu.sg

Objectives

- Upon the completion of this lecture, the students will be able
  - To understand concepts of evolutionary computation.
  - To know different techniques under evolutionary computation.

Computational Intelligence (CI)

- A broad definition of CI is the study of
  - adaptive mechanisms to enable or facilitate "*intelligent*" behavior of machine in complex, uncertain, and changing environments
    - to learn or adapt to new situations, to generalize, abstract, discover and associate
- *CI* is understood as solving various problems of Artificial Intelligence with the use of computer to perform numerical calculations
- Such computations are connected with application of the techniques such as:
  - Neural networks, Fuzzy logic, Evolutionary algorithms, Rough sets, etc.

CI & Soft Computing (SC)

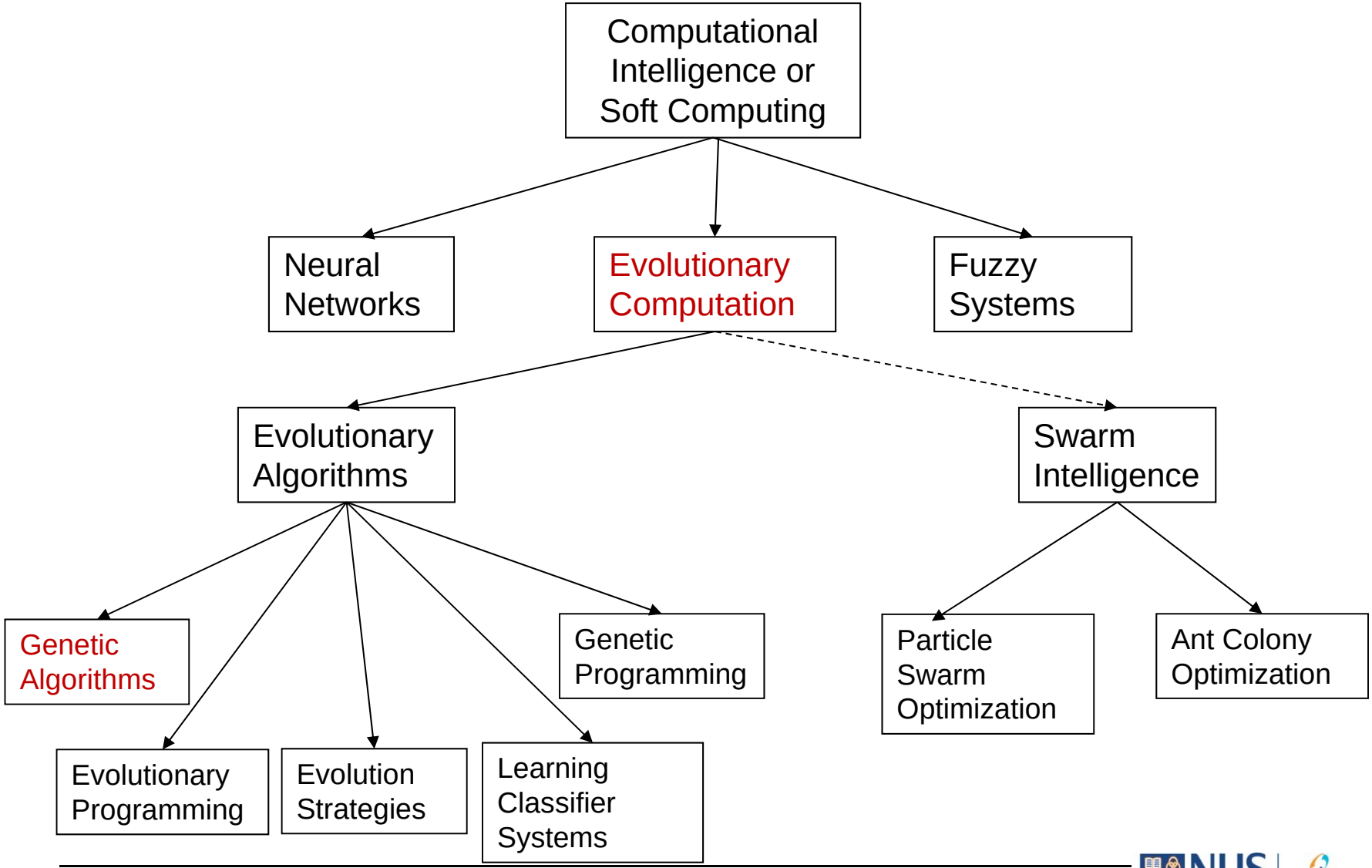
- SC includes principal members:
  - fuzzy logic (FL), neural networks (NN), evolutionary computation (EC)
    - Has neuro-fuzzy techniques as the back-bone
    - Has neural networks to help automatic learning and knowledge construction of knowledge based systems (KBS)
    - Has EC to enhance the performance of KBS
- CI and SC are sometimes used interchangeably, when indicating the major techniques of the field

Evolutionary Computation

"In computer science, **evolutionary computation** is a subfield of artificial intelligence (more particularly computational intelligence) that involves continuous optimization and combinatorial optimization problems. Its algorithms can be considered global optimization methods with a metaheuristic or stochastic optimization character and are mostly applied for black box problems (no derivatives known), often in the context of expensive optimization."

---Wikipedia

Evolutionary Computation Taxonomy



Darwinian Evolution

Inspired by Darwinian natural evolution:

- Survival of the fittest
- Selection on phenotype through environment
- Genotypic inheritance
- Reproduction
- Blind variation

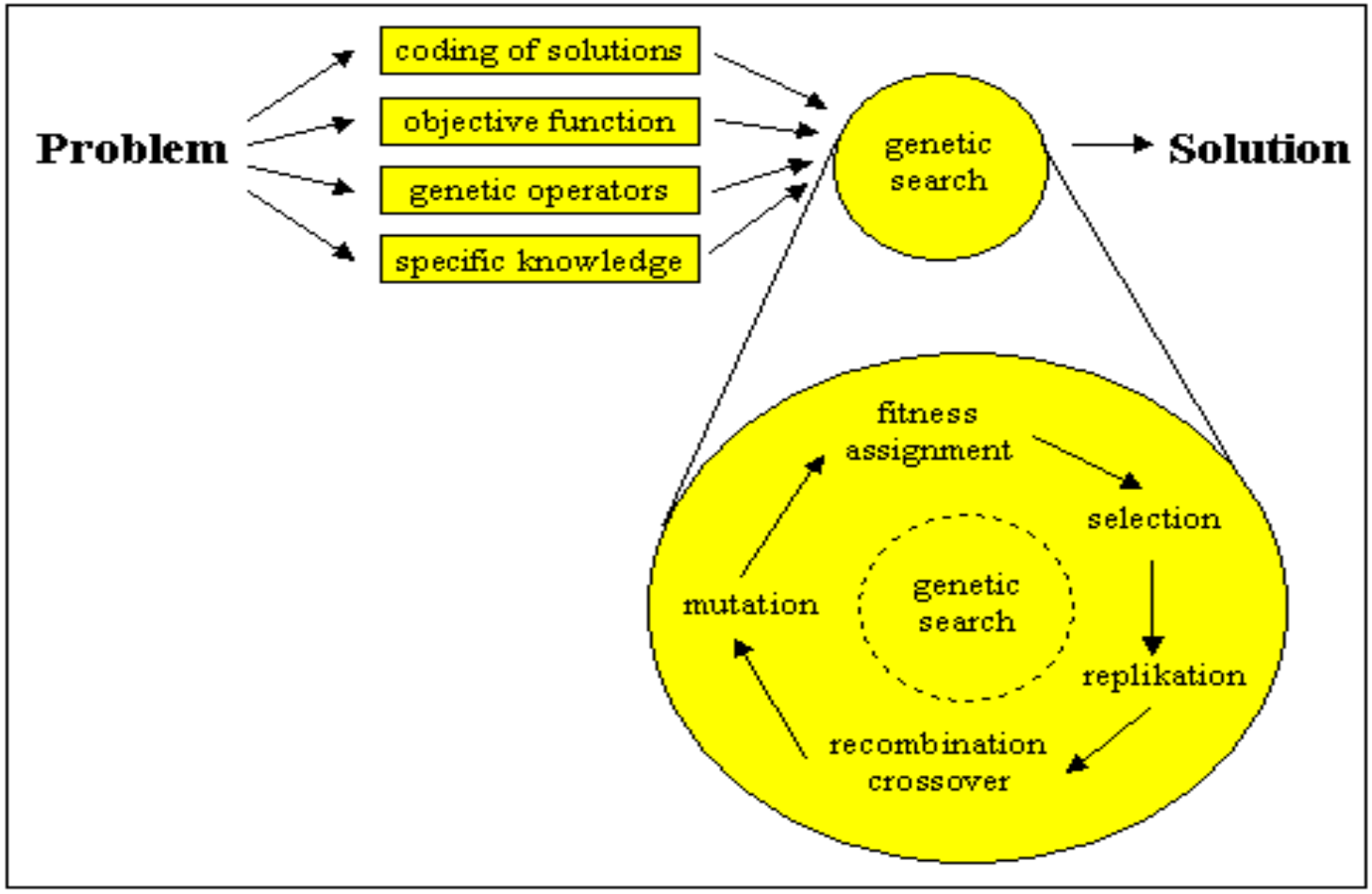
Evolutionary Computation Metaphor

EVOLUTION		PROBLEM SOLVING
Environment	⟷	Problem
Individual	⟷	Candidate Solution
Fitness	⟷	Quality

Evolutionary Algorithms History

- Evolutionary Programming
  - L. Fogel 1962 (San Diego, CA)
- Genetic Algorithms
  - J. Holland 1962 (Ann Arbor, MI)
- Evolution Strategies
  - I. Rechenberg & H.-P. Schwefel 1965 (Berlin, Germany)
- Genetic Programming
  - J. Koza 1989 (Palo Alto, CA)

Problem Solution
Using Evolutionary Algorithms



Evolutionary Algorithms

- Computer-based problem solving systems which use computational models that follow the principles of evolution and heredity in their design and implementation
- EAs maintain a population of structures, that evolve according to rules of selection, and other operators, that are referred to as “search operators”.
- Each individual in the population receives a measure of its fitness in the environment.
- Reproduction focuses attention on high fitness individuals.

Evolutionary Algorithms

- Evolutionary algorithms search a population of points in parallel, not a single point.
- Evolutionary algorithms do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Evolutionary algorithms are generally more straightforward to apply.

Evolutionary Algorithms

- Evolutionary algorithms use probabilistic transition rules, not deterministic ones.
- Evolutionary algorithms can provide a number of potential solutions to a given problem. The final choice is left to the user.
- However, evolutionary algorithms are less efficient compared to, say, hill-climbing techniques when the problem space is very well-behaved.

When to Use Evolutionary Algorithms

- Problems that are difficult to describe mathematically
- When space to be searched is large
- Approach to solving a problem not well understood
- Problems with many parameters that need to be simultaneously optimized

Application Domains

- Evolutionary Computation (EC) techniques are mainly used in optimisation, learning and design.
  - Numerical & Combinatorial Optimization
  - Engineering Design
  - Interactive Creative Design
  - Machine Learning
  - Scheduling and Control
  - Etc…

References

1) Hitch-Hiker's Guide to Evolutionary Computation
   - http://www.aip.de/~ast/EvolCompFAQ/

2) Forgel, L.J., Owens, A.F., & Walsh, M.J., "Artificial Intelligence through Simulated Evolution" New York: John Wiley (1966).

3) Holland, J.H., "Adaptation in Natural and Artificial Systems" , Ann Arbor, MI: The University of Michigan Press (1975).

4) Goldberg, D.E., "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley (1989).

5) Davis, L., "Handbook of Genetic Algorithms", Van Nostrand Reinhold, New York (1991).

6) Michalewicz, Z., "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Varlag, New York (1992).

References

7) Koza, J.R., “Genetic Programming: on the programming of computers by means of natural selection”, Cambridge, MA:MIT press (1992).

8) Genetic Algorithms & Grouping Problems, Emanuel Falkenauer, Wiley, 1998

9) Mitchell, M. “An Introduction to Genetic Algorithms”, MIT Press, Cambridge, MA, 1996

10) Koza, J. “Genetic Programming II”, MIT Press, Cambridge, MA, 1994

11) Baeck, Th. “Evolutionary Algorithms in Theory and Practice”, Oxford University Press, New York, 1996

12) Eiben, A.E., Smith, J.E., Introduction to Evolutionary Computing, Springer, 2010

13) De Jong, K. A., Evolutionary computation: a unified approach. MIT Press, Cambridge MA, 2006