1

Introduction to Computational Intelligence

Keep it simple:
As simple as possible,
But not simpler.
–Albert Einstein

1.1 Computational Intelligence

Much is unknown about intelligence and much will remain beyond human comprehension. The fundamental nature of intelligence is only poorly understood and even the definition of intelligence remains a subject of controversy. Considerable research is currently being devoted to the understanding and representation of intelligence. According to its dictionary definition, intelligence means the ability to comprehend, reason and learn. From this point of view, a definition of intelligence can be elicited whereby an intelligent system is capable of comprehending (with or without much *a priori* information) the environment or a process; reasoning about and identifying different environmental or process variables, their interrelationship and influence on the environment or process; and learning about the environment or process, its disturbance and operating conditions. Other aspects of intelligence that describe human intelligence are creativity, skills, consciousness, intuition and emotion.

Traditional artificial intelligence (AI) has tried to simulate such intelligent behaviour in systems requiring exact and complete knowledge representation (Turing, 1950). Unfortunately, many real-world systems cannot be described exactly with complete knowledge. It has been demonstrated that the use of highly complex mathematical description can seriously inhibit the ability to develop system models. Furthermore, it is required to cope with significant unmodelled and unanticipated changes in the environment or process and in the model objectives. This will involve the use of advanced decision-making processes to generate actions so that a certain performance level is maintained even though there are drastic changes in the operating conditions. Thus, the dissatisfaction with conventional modelling techniques is growing with

Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing, First Edition.
Nazmul Siddique and Hojjat Adeli.

the increasing complexity of dynamical systems, necessitating the use of more human expertise and knowledge in handling such processes. Computational intelligence techniques are thus a manifestation of the crucial time when human knowledge will become more and more important in system modelling and control as an alternative approach to classical mathematical modelling, whose structure and consequent outputs in response to external commands are determined by experimental evidence (i.e., the observed input/output behaviour of the system or plant). The system is then a so-called intelligent system. Intelligent techniques are properly aimed at processes that are ill-defined, complex, nonlinear, time-varying and stochastic. Intelligent systems are not defined in terms of specific algorithms. They employ techniques that can sense and reason without much *a priori* knowledge about the environment and produce control actions in a flexible, adaptive and robust manner.

1.2 Paradigms of Computational Intelligence

Many attempts have been made by different authors and researchers to define the term computational intelligence (CI). Despite the widespread use of the term, there is no commonly accepted definition of CI. The term was first used in 1990 by the IEEE Neural Networks Council. Bezdek (1994) first proposed and defined the term CI. A system is called computationally intelligent if it deals with low-level data such as numerical data, has a pattern-recognition component and does not use knowledge in the AI sense, and additionally when it begins to exhibit computational adaptivity, fault tolerance, speed approaching human-like turnaround and error rates that approximate human performance (Bezdek, 1994). At the same time, the birth of CI is attributed to the IEEE World Congress on Computational Intelligence in 1994. Since then there has been much explanation published on the term CI. The IEEE Computational Intelligence Society (formerly the IEEE Neural Networks Council) defines its subject of interest as neural networks (NN), fuzzy systems (FS) and evolutionary algorithms (EA) (Dote and Ovaska, 2001). Some authors argue that computational intelligence is a collection of heuristic algorithms encompassing techniques such as swarm intelligence, fractals, chaos theory, immune systems and artificial intelligence. There are also other approaches that satisfy the AI techniques. Marks (1993) clearly outlined the distinction between CI and AI, although both CI and AI seek similar goals. Based on three levels of analysis of system complexity, Bezdek (1994) argues that CI is a subset of AI.

Zadeh (1994, 1998) proposed a different view of machine intelligence, where he distinguishes hard computing techniques based on AI from soft computing techniques based on CI. In hard computing, imprecision and uncertainty are undesirable features of a system whereas these are the foremost features in soft computing. Figure 1.1 shows the difference between AI and CI along with their alliance with hard computing (HC) and soft computing (SC). Zadeh defines soft computing as a consortium of methodologies that provide a foundation for designing intelligent systems. Some researchers also believe that SC is a large subset of CI (Eberhart and Shui, 2007). The remarkable features of these intelligent systems are their human-like capability to make decisions based on information with imprecision and uncertainty.

Fogel (1995a) views adaptation as the key feature of intelligence and delineates the technologies of neural, fuzzy and evolutionary systems as the rubric of CI, denoting them as methods of computation that can be used to adapt solutions to new problems without relying on explicit human intervention. Adaptation is defined as the ability of a system to change or evolve its parameters or structure in order to better meet its goal. Eberhart and Shui (2007) believe that

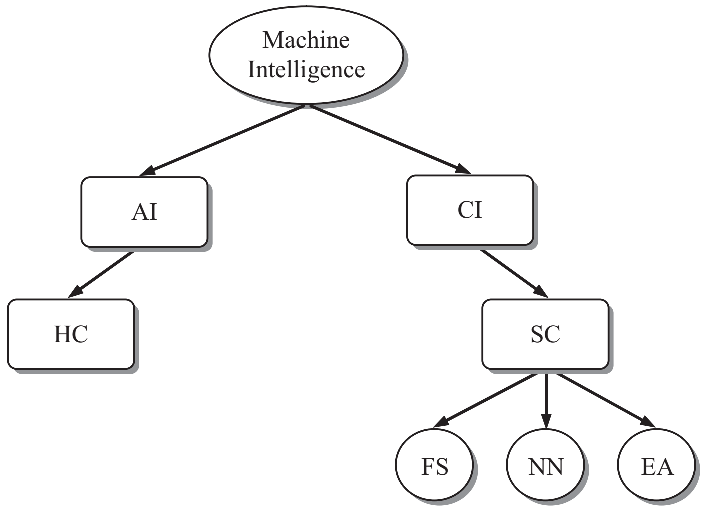

Figure 1.1 Difference between AI-HC and CI-SC

adaptation and self-organization play an important role in CI and argue that adaptation is central to CI, comprising the practical concept, paradigms, algorithms and implementations that facilitate intelligent behaviour. They argue further that CI and adaptation/self-organization are synonymous (Eberhart and Shui, 2007).

1.3 Approaches to Computational Intelligence

Central to computational intelligence is the construction of a process or system model (King, 1999; Konar, 2005), which is not amenable to mathematical or traditional modelling because:

(i) the processes are too complex to represent mathematically;
(ii) the process models are difficult and expensive to evaluate;
(iii) there are uncertainties in process operation;
(iv) the process is nonlinear, distributed, incomplete and stochastic in nature.

The system has the ability to learn and/or deal with new or unknown situations and is able to make predictions or decisions about future events. The term computational intelligence, as defined by Zadeh, is a combination of soft computing and numerical processing. The area of computational intelligence is in fact interdisciplinary and attempts to combine and extend theories and methods from other disciplines, including modern adaptive control, optimal control, learning theory, reinforcement learning, fuzzy logic, neural networks and evolutionary computation. Each discipline approaches computational intelligence from a different perspective, using different methodologies and toolsets towards a common goal. The inter-relationship between these disciplines is illustrated in Figure 1.2.

Computational intelligence uses experiential knowledge about the process that generally produces a model in terms of input/output behaviour. The question is how to model this human

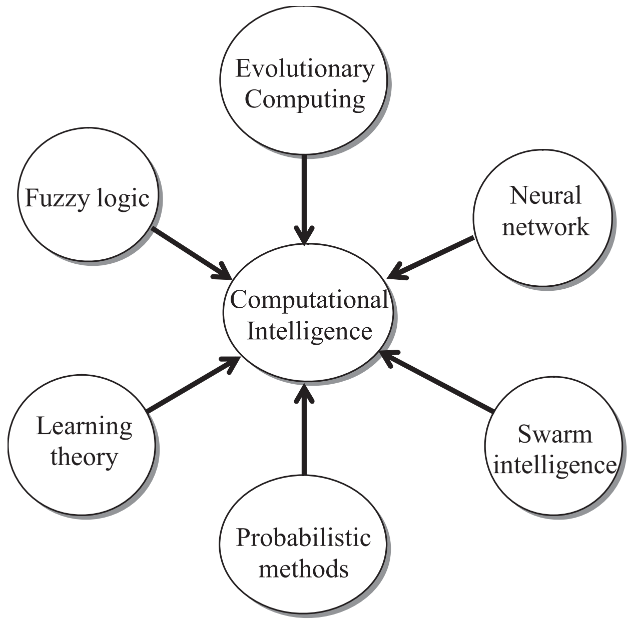

Figure 1.2 Periphery of computational intelligent methodologies

knowledge and represent it in such a manner as to be computationally efficient. Engelbrecht (2002) considers the following five basic approaches to computational intelligence:

(i) Fuzzy logic,
(ii) Neural networks,
(iii) Evolutionary computing,
(iv) Learning theory,
(v) Probabilistic methods,
(vi) Swarm intelligence.

In this book, the three methodologies of fuzzy logic, neural networks and evolutionary computing and their synergies will be covered and all other methodologies (such as swarm intelligence, learning theory and probabilistic methods) will be addressed as supportive methods in computational intelligence.

1.3.1 Fuzzy Logic

It has been suggested by researchers that measurements, process modelling and control can never be exact for real and complex processes. Also, there are uncertainties such as incompleteness, randomness and ignorance of data in the process model. The seminal work of Zadeh introduced the concept of fuzzy logic to model human reasoning from imprecise and incomplete information by giving definitions to vague terms and allowing construction of a rule base (Zadeh, 1965, 1973). Fuzzy logic can incorporate human experiential knowledge and give it an engineering flavour to model and control such ill-defined systems with nonlinearity and uncertainty. The fuzzy logic methodology usually deals with reasoning and inference on a higher level, such as semantic or linguistic.

1.3.2 Neural Networks

Neurons are the fundamental building blocks of the biological brain. Neurons receive signals from neighbouring neurons through connections, process them in the cell body and transfer the results through a long fibre called an axon. An inhibiting unit at the end of the axon, called the synapse, controls the signal between neurons. The axon behaves like a signal-conducting device. An artificial neural network is an electrical analogue of the biological neural network. Neural networks originated from the work of Hebb in the 1940s and more recently the work of Hopfield, Rumellhart, Grossberg and Widrow in the 1980s has led to a resurgence of research interest in the field (Hebb, 1949; Grossberg, 1982; Hopfield, 1982; Rummelhart *et al*., 1986; Widrow, 1987). Neural networks are biologically inspired, massively parallel and distributed information-processing systems. Neural networks are characterized by computational power, fault tolerance, learning from experiential data and generalization capability, and are essentially low-level computational algorithms that usually demonstrate good performance in processing numerical data. The learning takes place in different forms in neural networks, such as supervised, unsupervised, competitive and reinforcement learning. Research on neural network-based control systems has received considerable interest over the past several years, firstly because neural networks have been shown to be able to approximate any nonlinear function defined on a compact set of data to a specified accuracy and secondly because most control systems exhibit certain types of unknown nonlinearity, which suits neural networks as an appropriate control technology.

1.3.3 Evolutionary Computing

Evolutionary computing is the emulation of the process of natural selection in a search procedure based on the seminal work on evolutionary theory by Charles Darwin (Darwin, 1859). In nature, organisms have certain characteristics that influence their ability to survive in adverse environments and pass on to successive progeny with improved abilities. The genetic information of species can be coded into chromosomes that represent these characteristics. The species undergo reproduction and give birth to new offspring with features of capability to combat the adverse environment and survive. The process of natural selection ensures that the more fit individuals have the opportunity to mate most of the time, leading to the expectation that the offspring will have a similar or higher level of fitness. Evolutionary computation uses iterative progress and development in a population. This population is then selected in a guided random search using parallel processing to achieve the desired population of solutions. Such processes are often inspired by biological mechanisms of evolution. Nearly a century after Darwin's theory of evolution, Fraser (1957) was the first to conduct a simulation of genetic systems representing organisms by binary strings. Box (1957) proposed an evolutionary operation to optimize industrial production. Friedberg (1958) proposed an approach to evolve computer programs. The fundamental works of Lowrence Fogel (Fogel, 1962) in evolutionary programming, John Holland (Holland, 1962) in genetic algorithms, Ingo Rechenberg (Rechenberg, 1965) and Hans-Paul Schwefel (Schwefel, 1968) in evolutionary strategies have had significant influence on the development of evolutionary algorithms and computation as a general concept for problem solving and a powerful tool for optimization. Since the developmental years of the 1960s there have been significant contributions to the field by many people, including De Jong (1975), Goldberg (1989) and Fogel (1995b) to name a few. The 1990s saw another set of developments in evolutionary algorithms, for example Koza (1992) developed genetic programming, Reynolds (1994, 1999) developed cultural algorithms and Storn and Price (1997) developed differential evolution. Evolutionary algorithms have now found widespread application in almost all branches of science and engineering.

1.3.4 Learning Theory

Humans appear to be able to learn new concepts without much effort in a conventional sense. The mechanism of learning in humans is little known. In psychology, learning is the process of bringing together cognitive, emotional and environmental effects and experiences to acquire, enhance or change knowledge, skills, values and world views (Ormrod, 1995; Illeris, 2004). For any learning, it is also important how information is input, processed and stored. Learning theories provide explanations of such processes, and how exactly they occur (Vapnik, 1998).

Learning theories fall into three main philosophical frameworks: behaviourism, cognitive theories and constructivism. Behaviourism deals with the objectively observable aspects of learning. Cognitive theories look at how learning occurs in the brain. Constructivism views learning as a process in which the learner actively constructs or builds new ideas or concepts.

A new scientific discipline of machine learning (Samuel, 1959) has evolved based on the psychological learning theories. In machine learning, researchers use and apply four basic forms of learning. Supervised learning generates a function that maps inputs to desired outputs. Unsupervised learning models a set of input features and maps them to similar patterns, like clustering. Semi-supervised learning combines both labelled and unlabelled examples to generate an appropriate function or classifier. Reinforcement learning indicates how to act on a given observation from the environment. Every action has some impact on the environment, and the environment provides feedback in the form of rewards that guide the learning process. Learning mechanisms are an essential part of any intelligent system and hence are powerful tools for computational intelligence.

1.3.5 Probabilistic Methods

Probability theory has been viewed as the methodology of choice for dealing with uncertainty and imprecision. The probabilistic method involves considering an appropriate probability space over a wider family of structures, and proving that a sample point corresponding to the required structure has positive probability in this space. This method was introduced by Erdos and Spencer (1974) and has made major contributions in areas of mathematics and computer science such as combinatorics, functional analysis, number theory, topology, group theory, combinatorial geometry and theoretical computer science. Probabilistic behaviour or stochasticity (randomness) is also sometimes listed as an attribute of intelligent systems. A complex nonlinear dynamic system very often shows chaotic behaviour, that is, chaotic phenomena are features of complex dynamical systems (Grim, 1993). It is somewhat uncertain whether the attribute should be represented as randomness or chaos.

The term chaos refers to complicated dynamical behaviour. There is no uniform agreement as to the precise definition, but a significant body of literature uses the term to refer to systems of a particular type with a set of periodic points and an orbit which are dense in a closed invariant set Λ and these are very sensitive to initial conditions (Devaney, 1989). In principle, the future behaviour of a chaotic system is completely determined by the past, but in practice, any uncertainty in the choice of initial conditions grows exponentially with time. Chaotic behaviour has been observed in the laboratory in a variety of systems, such as electrical and electronic circuits, lasers, oscillating chemical reactions, fluid dynamics, mechanical and magneto-mechanical systems (Sumathi and Surekha, 2010). The dynamic behaviour of a chaotic system is predictable in the short term but impossible to predict in the long term. Chaos theory is essentially a recent extension of a larger field of mathematics which is part of complex nonlinear system dynamics. However, these theories seem to permeate many aspects of natural intelligent systems, from basic biology to behavioural intelligence, as well as most artificial intelligent processes and systems.

1.3.6 Swarm Intelligence

Swarm systems in nature are perhaps one of the most mesmerizing things to observe. A flock of birds twisting in the evening light, the V-shaped structure of migrating geese, winter birds hunting for food, the dancing of starlings in the evening light, ants marching to forage, the synchronized flashing of fireflies and mound building by termites are some of the fascinating examples of swarm systems. But how do they produce such well-choreographed collective behaviour without any central coordinator or leader? How do they communicate with each other? How does an ant which has found food tell other ants about the location of the food? How do the flocks of migrating geese maintain a V-shaped structure? How do fireflies know when to glow? Is there a central control or coordinator for the collective behaviours? Scientists and biologists have been researching for decades to answer some of these questions.

The collective behaviours of insects living in colonies (such as ants, bees, wasps and termites) have attracted researchers and naturalists for many years. Close observation of an insect colony shows that the whole colony is very organized, with every single insect having its own agenda. The seamless integration of all individual activities does not have any central control or any kind of supervision. Researchers are interested in this new way of achieving a form of collective intelligence, called swarm intelligence (SI) (Bonabeau *et al*., 1999; Kennedy and Eberhart, 2001). SI is widely accepted as a computational intelligence technique based around the study of collective behaviour in decentralized and self-organized systems typically made up of a population of simple agents interacting locally with one another and with their environment (Kennedy and Eberhart, 2001; Garnier *et al*., 2007). Although there is normally no centralized control structure dictating how individual agents should behave, local interactions between such agents often lead to the emergence of global behaviour. Examples of systems like this can be found in nature, including particle swarms, ant colonies, birds flocking, animals herding, fish schooling and bacterial foraging. Recently, biologists and computer scientists have studied how to model biological swarms to understand how such social insects interact, achieve goals and evolve.

Ants are social insects. They live in colonies and their behaviour is governed by the goal of colony survival rather than the survival of individuals. When searching for food, ants initially explore the surrounding area close to the nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants can smell the pheromone. When choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. It has been shown (Deneubourg *et al.*, 1990; Dorigo and Stützle, 2004) that the indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources.

Ant colonies or societies in general can be compared to distributed systems, which present a highly structured social organization in spite of simple individuals. The ant colonies can accomplish complex tasks far beyond their individual capabilities due to the structured organization of their society. The inspiring source of ant colony optimization (ACO) is the foraging behaviour of real ant colonies (Blum, 2005). Dorigo *et al.* (1996) were the first to propose a simple stochastic model that adequately describes the dynamics of the ants' foraging behaviour, and in particular, how ants can find shortest paths between food sources and their nest.

ACO is a meta-heuristic optimization algorithm that can be used to find approximate solutions to difficult combinatorial optimization problems and has been applied successfully to an impressive number of optimization problems. Applications of ACO include routing optimization in networks and vehicle routing, graph colouring, timetabling, scheduling and solving the quadratic assignment problem, the travelling salesman problem (Blum, 2005). Studies of the nest building of ants and bees have resulted in the development of clustering and structural optimization algorithms.

Flocking is seen as a feature of coherent manoeuvring of a group of individuals in space. This is a commonly observed phenomenon in some animal societies. Flocks of birds, herds of quadrupeds and schools of fish are often shown as fascinating examples of self-organized coordination (Camazine *et al.*, 2001). Natural flocks maintain two balanced behaviours: a desire to stay close to the flock and a desire to avoid collisions within the flock (Shaw, 1975). Joining a flock or staying with a flock seems to be the result of evolutionary pressure from several factors, such as protecting and defending from predators, improving the chances of survival of the (shared) gene pool from attacks by predators, profiting from a larger effective search for food, and advantages for social and mating activities (Shaw, 1962). Reynolds (1987) was the first to develop a model to mimic the flocking behaviour of birds, which he described as a general class of polarized, non-colliding, aggregate motion of a group of individuals. Such flocking behaviours were simulated using three simple rules: collision avoidance with flock mates, velocity matching with nearby flock mates, and flock centring to stay close to the flock. Flocking models have numerous applications. Some include the simulation of traffic patterns, such as the flow of cars on a motorway which has a flock-like motion, animating troop movement in real-time strategy games and in simulating mobile robot movement (Momen *et al.*, 2007; Turgut *et al.*, 2008).

One of the interesting features in the behaviour of fishes is the fish school. About half of all fish species are known to form fish schools at some stage in their lives. Fish can form loosely structured groups called shoals and highly organized structures called fish schools. Fish schools are seen as self-organized systems consisting of individual autonomous agents (Shaw, 1962). Fish schools also come in many different shapes: stationary swarms, predator-avoiding vacuoles and flash expansions, hourglasses and vortices, highly aligned cruising parabolas, herds and balls (Parrish *et al.*, 2002). A fish school can be of various sizes, for example, a herring school often exceeds 5000 individuals and spreads over 700 square metres

(Mackinson, 1999). Modelling the behaviour of fish schools has been a subject of research for a long time. Niwa (1996) studied the collective behaviour of fishes and proposed a model based on Newtonian dynamics which results in emergent patterns. Couzin *et al*. (2002) proposed an alternative model where each fish is considered as an autonomous agent interacting with its local neighbours and producing a complex pattern by following three simple rules: (i) move away from very near neighbours; (ii) follow the same direction as close neighbours; (iii) avoid becoming isolated. Following the rules, each individual fish can have three zones: repulsion, alignment and attraction. Individuals are attracted to neighbours over a larger range than they align with in the attraction zone. Individuals always move away from neighbours in the repulsion zone. If the radius of the alignment zone is increased, individuals would go from a loosely packed stationary swarm to a torus where individuals circle round their centre of mass and, finally, to a parallel group moving in a common direction.

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart (1995) based on the social behaviour of swarms such as fish and birds in nature. PSO has similarities with evolutionary algorithms, but it is simpler in the sense that it does not apply any mutation or crossover operation, instead using real-number randomness and global communication among the swarming particles. Each particle, referring to an individual in the swarm representing a candidate solution to the optimization problem, is flown through the multidimensional search space, adjusting its position in the search space according to its own experience and that of its neighbouring particles. Particles make use of the best positions encountered and the best positions of their neighbours to position themselves towards an optimum solution. The performance of each particle is measured according to a predefined fitness function which is related to the problem being solved. Applications of PSO include function approximation, clustering, optimization of mechanical structures and solving systems of equations. There are now as many as about 20 different variants of PSO.

Rumours are a form of social communication. The way a rumour propagates within a population in society was first modelled by Daley and Kendal (1965) at the University of Cambridge. The spreading of a rumour often has severe consequences on the perception of celebrities, financial markets and even society (Nekovee *et al*., 2007). Rumours can also be manipulated intentionally to disrupt competitor organizations. They can cause panic during wars and can create disaster in stock markets.

The flashing of fireflies in the summer sky in tropical regions has been one of the most hypnotic and wonderful experiences for explorers and naturalists for many years. There are about 2000 firefly species, and most fireflies produce short and rhythmic flashes. The flashing light can be seen as a signalling system and the true function of such signalling system is not really known yet. However, two fundamental functions of such flashes are to attract mating partners and to attract potential prey; flashing may also serve as a warning mechanism. The rhythm, rate and duration of flashing form part of the signal system that brings two fireflies together. For example, females respond to a male's unique pattern of flashing. This unique feature of fireflies can be formulated in such a way as to make it possible to formulate new optimization algorithms.

Yang (2009) proposed a new heuristic algorithm, called the firefly algorithm (FA), based on three idealized rules: (i) fireflies attract one another with flashing lights; (ii) the level of attractiveness is proportional to their brightness and a less bright firefly will move towards a brighter one, otherwise it will move randomly; (iii) the brightness of a firefly is determined by the landscape of its objective function. For a maximization problem, a population of fireflies is generated and the brightness is simply proportional to the value of the objective function (fitness value). FA has found many applications in engineering and multi-objective optimization problems (Yang, 2008, 2010).

Quite a number of cuckoo species engage in obligate brood parasitism by laying their eggs in the nests of other host birds of different species. If the host bird discovers the eggs are not its own, it either throws the eggs away or abandons the nest. Some cuckoo species are specialized in the mimicry of colour and pattern of eggs of their chosen host species, thus reducing the chances of their eggs being thrown out or abandoned. Yang and Deb (2010) developed a new meta-heuristic optimization algorithm, called cuckoo search (CS), which is based on the interesting breed behaviour of certain cuckoo species. There have been many applications of cuckoo search reported in the literature (Yang, 2008, 2010).

MacArthur and Wilson (1967) began working together on mathematical models of biogeography in the 1960s. They were trying to develop mathematical models of biogeography that describe how species migrate from one island to another, how new species arise and how species become extinct. Since then biogeography has become a major area of research, which studies the geographical distribution of biological species. The concept of biogeography can be used to derive a new family of algorithms for optimization called biogeography-based optimization (BBO). BBO has been applied to benchmark functions and to a sensor-selection problem, providing performance on a par with other population-based methods (Simon, 2008).

Passino (2002) pointed out, in a seminal paper, how individual and groups of bacteria forage for nutrients and how to model this as a distributed optimization process; the natural foraging strategy can lead to optimization and the idea can be applied to solve real-world optimization problems. Based on this concept, Passino (2002) proposed an optimization technique known as the bacterial foraging optimization algorithm (BFOA). To date, BFOA has been applied successfully to real-world problems such as optimal controller design, harmonic estimation, transmission loss reduction, active power filter synthesis and learning of artificial neural networks (Das *et al*., 2009).

Several new meta-heuristic optimization algorithms inspired by nature have been introduced in recent years. Among them are a galaxy-based search algorithm (Hosseini, 2011) and spiral dynamics-inspired optimization (Tamura and Yasuda, 2011).

EA and SI together form a broader class of search and optimization paradigm termed global search and optimization (GSO). The classification of the different algorithms and techniques of GSO is shown in Figure 1.3.

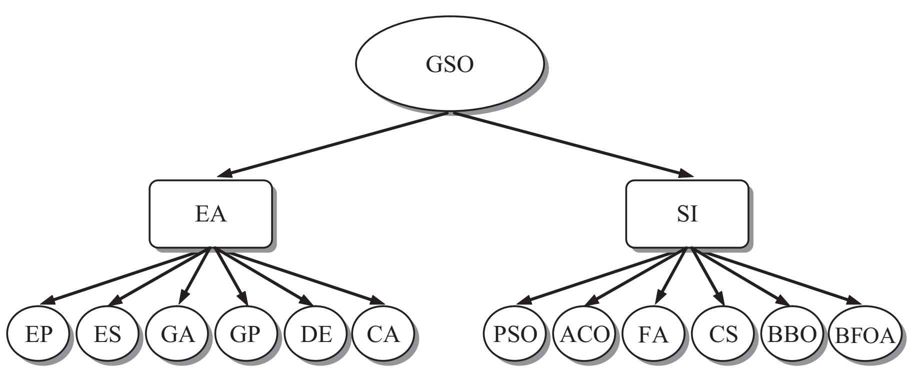

Figure 1.3 Classification of GSO

1.4 Synergies of Computational Intelligence Techniques

The synergistic combination of all the methodologies has a very rational basis for applications and designing intelligent systems. An individual method can be excellent in approximate reasoning and modelling uncertainty but may not be good at learning with experiential data or may not be good at adapting in an unknown environment. Thus, a combined approach with computational intelligence techniques and their implementation is of importance for overall performance, computation cost and convenience of application. This combination is called a hybrid intelligent system by many researchers. Zadeh (1994) thinks hybrid intelligent systems are definitely the way of the future.

Fuzzy logic is good at approximate reasoning but does not have any learning ability or adaptive capacity. Neural networks, on the other hand, have efficient mechanisms in learning from experiential data. Evolutionary algorithms enable a system to adapt behaviour or optimize structure. The synergistic combination of these methodologies can provide better computational models that will complement the limitations of any single method. Depending on the compatibility of the individual methodologies, the synergism can be classified into two types: strongly coupled and weakly coupled. In strongly coupled synergism, the individual methodologies are hybridized in such as way as to be inseparable and each individual methodology loses most of its identity in the combined structure. In weakly coupled synergism, each individual methodology plays its own part by upholding the structural identity and working towards a common goal.

The different forms of synergisms of fuzzy logic, neural networks and evolutionary algorithms are shown in Figure 1.4. The common forms of synergism of fuzzy systems and evolutionary algorithms include tuning, optimization and learning of membership functions,

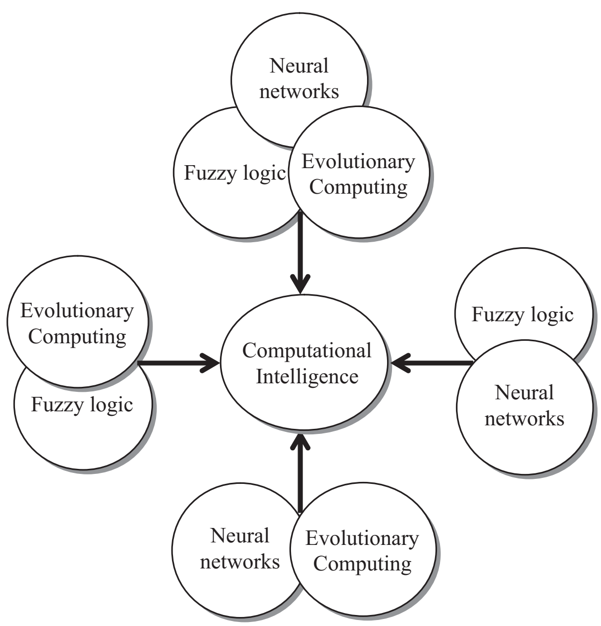

Figure 1.4 Synergies of computational intelligent methodologies

parameters and rule-based fuzzy systems using evolutionary algorithms. Another form of synergism is the control of different parameters of evolutionary algorithm by a fuzzy controller. Both forms of fuzzy evolutionary synergism are weakly coupled.

The common forms of weakly coupled synergisms of neural networks and evolutionary algorithms include training, designing, optimizing architecture and parameters of neural networks and feature selection, transformation and scaling of training data for neural networks using evolutionary algorithms. Also, neural networks are being used to control parameters of evolutionary algorithms. There is also a strongly coupled synergism between the two methodologies where the genetic operations are represented in the form of a neural network and the training epochs are meant to be the generations of evolution.

Synergisms of neural networks and fuzzy systems are the most common and have proved to be very powerful tools for system modelling and control. In a weakly coupled synergism, neural networks and fuzzy systems work independently towards a common goal, where neural networks assist fuzzy systems to acquire knowledge and rules, tuning or adjusting membership functions. In strongly coupled synergism, a fuzzy system is represented in the form of a neural network, which can learn from experiential data. The literature is rich in this type of synergism.

The final type of synergism is a combination of the three methodologies. The most common synergism is the training or optimizing structure of a hybrid neuro-fuzzy system using an evolutionary algorithm. A strongly coupled synergism of the three methodologies may not be possible. There are other types of synergisms possible between swarm intelligence, fuzzy systems, evolutionary algorithms and neural networks.

1.5 Applications of Computational Intelligence

The essence of systems based on computational intelligence is the process that interprets information and data of various natures. The other feature of computational intelligence is that where processing of information in algorithms becomes difficult. Developed theories have been quickly applied to various fields of computer science, engineering, data analysis and biomedicine. Each component methodology of the CI has its application areas. Certainly, more than one technology can be applied to the same application. For example, data clustering can be performed using neural networks and fuzzy logic but the difference would be in the accuracy of the performance. The application areas of neural networks can be categorized into five groups, such as data analysis and classification, associative memory, clustering, generation of patterns and control. Neural networks have been applied to analyse and classify medical data and images, for example, EEG, cancer data, etc. Neural networks have also been widely used for face detection, fraud detection and pattern analysis. The use of neural networks in nonlinear control applications is the most successful area. The inherent advantage of neural networks is that they can deal with nonlinearities of a system and model such systems when sufficient data are available. The application areas of evolutionary algorithms are optimization and multi-objective optimization. Since traditional mathematical optimization techniques are difficult or too costly to apply to many problem domains such as robot tract determination, scheduling problems, DNA analysis, optimization of large structural parameters, etc., evolutionary algorithms are becoming popular for these problems. Fuzzy logic and fuzzy systems have found a wide range of applications such as control, image processing and decision making. Fuzzy logic control has been applied to many household appliances such as washing machines, microwave ovens, toasters, vacuum cleaners, etc. One application of a fuzzy controller is well known: its implementation on a video camera to stabilize the image while holding the camera unsteadily. Fuzzy expert systems have been applied to many areas of medical diagnostics, scheduling, foreign exchange trading and business strategy selection.

1.6 Grand Challenges of Computational Intelligence

Though the CI techniques have been applied successfully to scientific, engineering, economic, business and industrial problems, CI seriously lacks efficient knowledge acquisition, representation and retrieval structures. The grand challenges for the CI community would be to propose more efficient knowledge representation and retrieval mechanisms. Feigenbaum (2003) thinks the grand challenge would be to build a large knowledge base by simply reading text and thus reducing the knowledge engineering effort by one order of magnitude. Some researchers argue that CI should be more human-centric, helping humans to formulate their goals and solve their problems, leading to personal fulfilment (Duch, 2007). A long-term goal for CI would be to create cognitive systems that can compete with humans in a large number of problems. A good part of CI research is concerned with low-level cognitive functions such as perception, object recognition, signal analysis, finding structure in data and association tasks. Despite great progress in CI, artificial systems designed to solve lower-level cognitive tasks are far behind simple natural systems. From this point of view, CI needs to focus on higher-level cognitive systems using symbolic knowledge representation. CI is more than the study of the design of intelligent systems; it includes all non-algorithmizable processes which humans can perform with various degrees of competence. In that sense, Goldberg and Harik (1996) see CI more as a way of thinking about problems rather than a solution to problems using specific techniques.

1.7 Overview of the Book

Chapter 2 describes the concepts of fuzzy logic, fuzzy sets and the description of fuzzy sets by membership functions, different types of membership functions and their features. To apply fuzzy logic one needs to understand the operations of fuzzy sets and fuzzy relations, which are discussed in the chapter with examples. The chapter also describes the interesting features of linguistic variables and hedges. The chapter shows the fascinating features of fuzzy if–then rules and inference mechanisms that will help in developing applications. The chapter also provides a set of worked examples.

Chapter 3 presents an investigation into different types of fuzzy systems, fuzzy modelling and fuzzy control methods and techniques in general. These include simple Mamdani-, Sugeno- and Tsukomoto-type fuzzy modelling and control techniques. A comparative study of the suitability of different methods for applications is made. The chapter also presents different types of fuzzy controllers, namely PD, PI and PID. Different approaches to rule reduction are investigated and analysed as well.

Chapter 4 presents an introduction to biological neurons, different models of neurons, activation functions and basics of neural networks. The chapter then introduces different feedforward architectures such as the multilayer perceptron, radial-basis function, regression networks, probabilistic, belief and stochastic networks and recurrent architectures such as Elman, Jordan and Hopfield networks. The chapter describes different learning algorithms of neural networks, such as supervised and unsupervised.

Chapter 5 presents neural systems with application to nonlinear systems. Different techniques of identification and modelling of nonlinear systems using neural networks have been discussed. Application of neural networks to control problems is a very popular and widely used technique. Different schemes of neuro-control, such as direct, indirect, backpropagation through time and inverse control, have been discussed. Neural networks have also been popular for predictive and adaptive control. Different schemes of predictive and adaptive neuro-control with applications have also been discussed. This chapter introduces different application developments with MATLAB® as well.

Chapter 6 presents evolutionary computing and algorithms. Basic to any evolutionary computing is the chromosome representation and genetic operators. This chapter describes different types of encoding scheme, selection and crossover and mutation operators. Finally, it introduces different evolutionary algorithms such as genetic algorithms, genetic programming, evolutionary programming, evolutionary strategies, differential evolution and cultural algorithms.

Chapter 7 presents an investigation into different evolutionary systems and their applications. Multi-objective optimization is one of the promising areas of application of evolutionary algorithms. This chapter also investigates co-evolution of populations and different symbiotic relationships between species. Another aspect in evolutionary algorithms is parallelism, where multiple populations work together with a common goal. This chapter presents an account of these techniques widely used in evolutionary computing.

Chapter 8 presents combinations of fuzzy systems and evolutionary computing. Different kinds of combination are possible, such as controlling parameters of evolutionary algorithms by fuzzy logic and optimizing parameters of a fuzzy system by evolutionary algorithms. This chapter presents the optimization of fuzzy systems, especially membership functions, rule-based or both using evolutionary algorithms and also highlights the fuzzy control of genetic operators in limited applications.

Chapter 9 presents combinations of evolutionary algorithms and neural networks. Mainly two types of combination, supportive and collaborative, between evolutionary algorithms and neural networks have been reported in the literature. In supportive combinations, one of the two technologies is the primary problem solver and the other plays a supporting role, such as setting up initial conditions or parameters. In collaborative combinations, both of the technologies act together as a problem solver. This chapter will explore these combinations in designing and training neural networks, learning control parameters, activation functions and setting up initial conditions.

Chapter 10 presents combinations of neural networks and fuzzy systems, the most important of which are cooperative and hybrid combinations. In cooperative combinations, fuzzy systems or neural networks are used to control parameters, initial conditions and/or structures of neural networks or fuzzy systems. This chapter covers the detailed description, architectures and use of possible combinations of these two technologies. In hybrid combinations, each of these technologies loses its identity and presents a new single system to address the problem at hand. The most successful and widely used hybrid system is the ANFIS. This chapter will introduce ANFIS and different variants as well as other hybrids.

1.8 MATLAB® Basics

MATLAB® is a high-level language for scientific and engineering computation. MATLAB® is an integrated software environment and provides numeric computation, data analysis, graphics visualization and system simulation. The integrated environment of MATLAB® is shown in

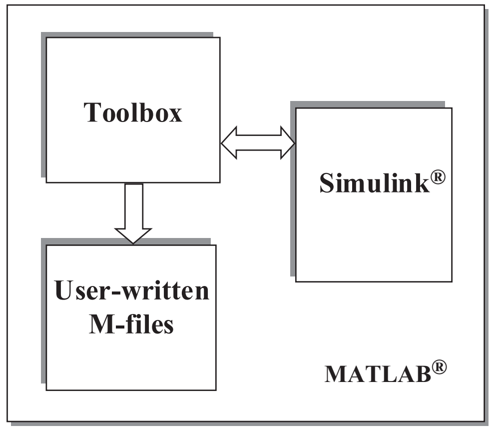

Figure 1.5 MATLAB® environment

Figure 1.5. The language, tools and built-in maths functions enable users to explore multiple approaches and reach faster solutions than with other programming languages. It provides tools for creating customized toolboxes or harnessing with other toolboxes such as fuzzy logic, direct search and genetic algorithms or neural network toolboxes. Applications are developed by writing M-files and running at the command prompt. Simulink® is a software package for modelling, simulating and analysing dynamical systems under MATLAB®. It supports linear and nonlinear systems, modelled in continuous time, sampled time or a hybrid of the two. Different parts of the system can have different rates. Simulink® provides a graphical user interface (GUI) for building models as block diagrams using click-and-drag mouse operations. Simulink® also includes a comprehensive block library of sinks, sources, linear and nonlinear components and connectors.

This section provides a brief introduction to different command-line functions of MATLAB®. A brief introduction to MATLAB®, different functions, control statements, writing M-files and plot functions are discussed, with examples in Appendix A.

References

Bezdek, J.C. (1994) What is computational intelligence? In *Computational Intelligence Imitating Life*, J.M. Zurada, R.J. Marks II and C.J. Robinson (eds), IEEE Press, New York, pp. 1–12.

Blum, C. (2005) Ant colony optimisation: introduction and recent trends, *Physics of Life Reviews*, 2, 353–373.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York.

Box, G.E.P. (1957) Evolutionary operation: a method for increasing industrial productivity, *Applied Statistics*, 6(2), 81–101.

Camazine, S., Deneubourg, J.-L., Franks, N.R., Sneyd, J., Theraulaz, G. and Bonabeau, E. (2001) *Self-Organization in Biological Systems*, Princeton University Press, Princeton, NJ.

Couzin, I.D., Karause, J., James, R., Ruxton, G.D. and Franks, N.R. (2002) Collective memory and spatial sorting in animal groups, *Journal of Theoretical Biology*, 218, 1–11.

Daley, D.J. and Kendal, D.G. (1965) Stochastic rumours, *Journal of the Institute of Mathematics and its Applications*, 1, 42–55.

Darwin, C. (1859) *The Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*, Mentor, New York (reprint 1958).

Das, S., Dasgupta, S., Biswas, A., Abraham, A. and Konar, A. (2009) On the stability of the chemotactic dynamics in bacterial-foraging optimisation algorithm, *IEEE Transactions on Systems, Man and Cybernetics A*, 39(3), 670–679.

Dote, Y. and Ovaska, S.J. (2001) Industrial applications of soft computing: a review, *Proceedings of the IEEE*, 89(9), 1243–1265.

De Jong, K.A. (1975) *Analysis of the behaviour of a class of genetic adaptive systems*, PhD Thesis, University of Michigan, Ann Arbor, MI.

Deneubourg, J.-L., Aron, S., Goss, S. and Pasteels, J.-M. (1990) The self-organizing exploratory pattern of the Argentine ant, *Journal of Insect Behaviour*, 3, 159–168.

Devaney, R.L. (1989) *An Introduction to Chaotic Systems*, 2nd edn, Addison Wesley, New York.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996) Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on System, Man and Cybernetics B*, 26(1), 29–41.

Dorigo, M. and Stützle, T. (2004) *Ant Colony Optimization*, MIT Press, Cambridge.

Duch, W. (2007) Towards comprehensive foundations of computational intelligence. In *Challenges for Computational Intelligence*, W. Duch and J. Mandziuk (eds), Springer-Verlag, Berlin.

Eberhart, R.C. and Shui, Y. (2007) *Computational Intelligence – Concepts to Implementations*, Elsevier, Amsterdam.

Engelbrecht, A.P. (2002) *Computational Intelligence: An Introduction*, John Wiley & Sons, New York.

Erdos, P. and Spencer, J. (1974) *Probabilistic Methods in Combinatorics*, Academic Press, New York.

Feigenbaum, E.A. (2003) Some challenges and grand challenges for computational intelligence, *Journal of the ACM*, 50(1), 32–40.

Fogel, D.B. (1995a) Review of computational intelligence: imitating life, *IEEE Transactions on Neural Networks*, 6, 1562–1565.

Fogel, D.B. (1995b) *Evolutionary Computation – Toward a New Philosophy of Machine Intelligence*, IEEE Press, New York.

Fogel, L.J. (1962) Autonomous automata, *Industrial Research*, 4, 14–19.

Fraser, A.S. (1957) Simulation of genetic systems by automatic digital computers, I. Introduction, *Australian Journal of Biological Sciences*, 10, 484–491.

Friedberg, R.M. (1958) A learning machine: Part I, *IBM Journal of Research and Development*, 2(1), 2–13.

Garnier, S., Gautrais, J. and Theraulaz, G. (2007) The biological principles of swarm intelligence, *Swarm Intelligence*, 1, 3–31.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, New York.

Goldberg, D.E. and Harik, G. (1996) A case study in abnormal CI: the design manufacturing and others anthropocentric systems, *International Journal of Computational Intelligence and Organisations*, 1, 78–93.

Grim, P. (1993) Self-reference and chaos in fuzzy logic, *IEEE Transactions on Fuzzy Systems*, 1(4), 237–253.

Grossberg, S. (1982) *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, Reidell Press, Boston, MA.

Hebb, D.O. (1949) *The Organization of Behaviour: A Neuropsychological Theory*, John Wiley, New York.

Holland, J.H. (1962) Outline for a logical theory of adaptive systems, *Journal of ACM*, 3, 297–314.

Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Sciences*, 79, 2554–2558.

Hosseini, H.S. (2011) Principal component analysis by galaxy-based search algorithm: a novel metaheurisitc for continuous optimisation, *International Journal of Computational Science and Engineering*, 6(1&2), 132–140.

Illeris, K. (2004) *Three Dimensions of Learning*, Krieger Publishing, Malabar, FL.

Kennedy, J. and Eberhart, R.C. (2001) *Swarm Intelligence*, Morgan-Kaufmann, New York.

King, R.E. (1999) *Computational Intelligence in Control Engineering*, Marcel Dekker, New York.

Konar, A. (2005) *Computational Intelligence: Principles, Techniques and Applications*, Springer-Verlag, Berlin.

Koza, J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA.

MacArthur, R. and Wilson, E. (1967) *The Theory of Biogeography*, Princeton University Press, Princeton, NJ.

Mackinson, S. (1999) Variation in structure and distribution of pre-spawning Pacific herring shoals in two regions of British Columbia, *Journal of Fish Biology*, 55, 972–989.

Marks, R. (1993) Intelligence: computational versus artificial, *IEEE Transactions on Neural Networks*, 4(5), 737–739.

Momen, S., Amavasai, B.P. and Siddique, N.H. (2007) Mixed species flocking for heterogenous robotic swarms. In *The International Conference on Computer as a Tool* (EUROCON 2007), Piscataway, NJ. IEEE Press, New York, pp. 2329–2336.

Nekovee, M., Moreno, Y., Bianconi, G. and Marsili, M. (2007) Theory of rumour spreading in complex social networks, *Journal of Physica A*, 374, 457–470.

Niwa, H.S. (1996) Newtonian dynamical approach to fish schooling, *Journal of Theoretical Biology*, 181, 47–63.

Ormrod, J.E. (1995) *Human Learning*, Prentice Hall, Englewood Cliffs, NJ.

Parrish, J.K., Viscido, S.V. and Grunbaum, D. (2002) Self-organized fish schools: an examination of emergent properties, *The Biological Bulletin*, 202, 296–305.

Passino, K.M. (2002) Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control System Magazine*, 22(3), 52–67.

Rechenberg, I. (1965) *Cybernetic Solution Path of an Experimental Problem*, Royal Aircraft Establishment, Library Translation No. 1122, Farnborough, UK.

Reynolds, C. (1987) Flocks, herds, and schools: a distributed behavioural model, *Computer Graphics*, 21(4), 25–34.

Reynolds, R.G. (1994) Introduction to cultural algorithms. In *Proceedings of the Third Annual Conference on Evolutionary Programming*, A.V. Sebald and L.J. Fogel (eds), World Scientific Press, Singapore, pp. 131–139.

Reynolds, R.G. (1999) *An Overview of Cultural Algorithms: Advances in Evolutionary Computation*, McGraw-Hill, New York.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning representations by back-propagation errors, *Nature*, 323, 533–536.

Samuel, A.L. (1959) Some studies in machine learning using game checkers, *IBM Journal of Research and Development*, 3, 211–229.

Schwefel, H.-P. (1968) *Projekt MHD-Strausstrhlrohr: Experimentelle Optimierung einer Zweiphasenduese*, Teil I, Technischer Bericht 11.034/68, 35, AEG Forschungsinstitute, Berlin.

Shaw, E. (1962) The schooling of fishes, *Scientific American*, 206, 128–138.

Shaw, E. (1975) Fish in schools, *Natural History*, 84(8), 40–46.

Simon, D. (2008) Biogeography-based optimization, *IEEE Transactions on Evolutionary Computation*, 12(6), 702–713.

Storn, R. and Price, K. (1997) Differential evolution – a simple and efficient heuristic for global optimisation over continuous space, *Journal of Global Optimisation*, 11(4), 431–459.

Sumathi, S. and Surekha, P. (2010) *Computational Intelligence Paradigms – Theory and Applications Using MATLAB*, CRC Press/Taylor & Francis, Boca Raton, FL.

Tamura, K. and Yasuda, K. (2011) Primary study of spiral dynamics inspired optimisation, *IEEJ Transactions on Electrical and Electronic Engineering*, 6(S1), 98–100.

Turgut, A.E., Çelikkanat, H., Gökçe, F. and Sahin, E. (2008) Self-organized flocking in mobile robot swarms, *Swarm Intelligence*, 2, 97–120.

Turing, A.M. (1950) Computing machinery and intelligence, *Mind*, 59, 433–460.

Vapnik, V.N. (1998) *Statistical Learning Theory*, John Wiley & Sons, New York.

Widrow, B. (1987) Adaline and Madaline, *Proceedings of the IEEE First International Conference on Neural Networks*, 1, 145–157.

Yang, X.S. (2008) *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Bristol.

Yang, X.S. (2009) Firefly algorithms for multimodal optimization. In *Stochastic Algorithms: Foundations and Applications*, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, Springer-Verlag, Berlin, pp. 169–178.

Yang, X.S. (2010) *Engineering Optimisation: An Introduction with Metaheuristic Applications*, John Wiley & Sons, New York.

Yang, X.S. and Deb, S. (2010) Engineering optimisation by cuckoo search, *International Journal of Mathematical Modelling and Numerical Optimisation*, 1(4), 330–343.

Zadeh, L.A. (1965) Fuzzy sets, *Information and Control*, 8(3), 338–353.

Zadeh, L.A. (1973) Outline of a new approach to the analysis of complex systems and decision process, *IEEE Transactions on System, Man and Cybernetics*, 3, 28–44.

Zadeh, L.A. (1998) Roles of soft computing and fuzzy logic in the conception, design and deployment of information/intelligent systems. In *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*, O. Kaynak, L.A. Zadeh, B. Tiirksen and I.J. Rudas (eds), Springer-Verlag, Berlin, pp. 10–37.

Zadeh, L.A. (1994) Fuzzy logic, neural networks and soft computing, *Communications of the ACM*, 37, 77–84.