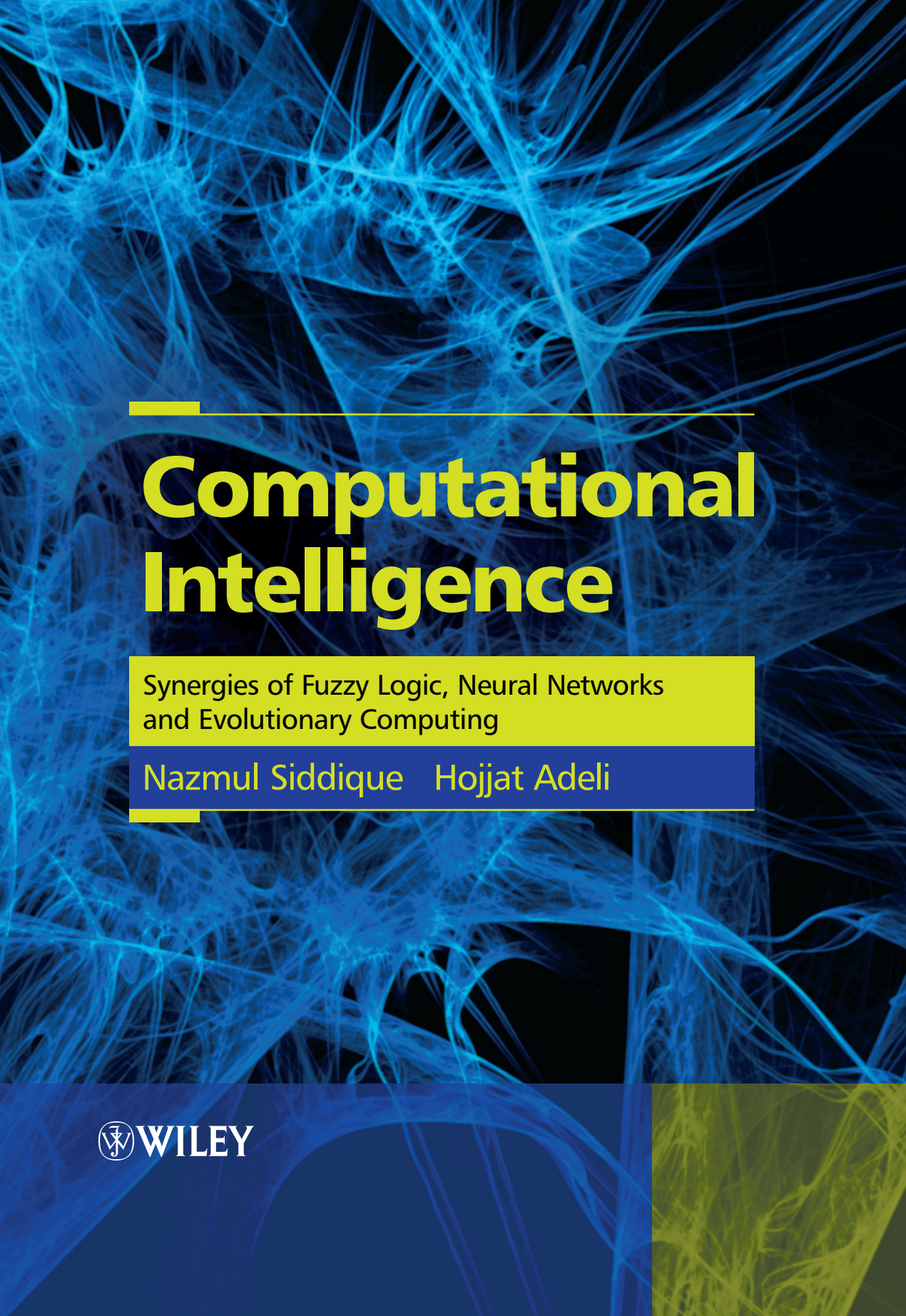
Computational Intelligence
Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing
Nazmul Siddique Hojjat Adeli
WILEY

COMPUTATIONAL INTELLIGENCE

COMPUTATIONAL INTELLIGENCE

SYNERGIES OF FUZZY LOGIC, NEURAL NETWORKS AND EVOLUTIONARY COMPUTING

Nazmul Siddique
University of Ulster, UK

Hojjat Adeli
The Ohio State University, USA

WILEY
A John Wiley & Sons, Ltd., Publication

This edition first published 2013

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

Library of Congress Cataloging-in-Publication Data

Siddique, N. H.
Computational intelligence : synergies of fuzzy logic, neural networks, and evolutionary computing / Nazmul Siddique, Hojjat Adeli.
pages cm
Includes bibliographical references and index.
ISBN 978-1-118-33784-4 (cloth)
1. Computational intelligence. I. Adeli, Hojjat, 1950– II. Title.
Q342.S53 2013
006.3–dc23

2012047736

A catalogue record for this book is available from the British Library

ISBN: 9781118337844

Typeset in 10/12pt Times by Aptara Inc., New Delhi, India

To Kaniz, Oyndrilla, Opala and Orla
– Nazmul

To Nahid, Amir, Anahita, Cyrus and Mona
– Hojjat

Contents

Foreword **xiii**

Preface **xv**

Acknowledgements **xix**

1 Introduction to Computational Intelligence **1**
1.1 Computational Intelligence 1
1.2 Paradigms of Computational Intelligence 2
1.3 Approaches to Computational Intelligence 3
1.3.1 Fuzzy Logic 4
1.3.2 Neural Networks 5
1.3.3 Evolutionary Computing 5
1.3.4 Learning Theory 6
1.3.5 Probabilistic Methods 6
1.3.6 Swarm Intelligence 7
1.4 Synergies of Computational Intelligence Techniques 11
1.5 Applications of Computational Intelligence 12
1.6 Grand Challenges of Computational Intelligence 13
1.7 Overview of the Book 13
1.8 MATLAB® Basics 14
References 15

2 Introduction to Fuzzy Logic **19**
2.1 Introduction 19
2.2 Fuzzy Logic 20
2.3 Fuzzy Sets 21
2.4 Membership Functions 22
2.4.1 Triangular MF 23
2.4.2 Trapezoidal MF 23
2.4.3 Gaussian MF 24
2.4.4 Bell-shaped MF 24
2.4.5 Sigmoidal MF 26
2.5 Features of MFs 27
2.5.1 Support 27
2.5.2 Core 27

	2.5.3 Fuzzy Singleton	27
	2.5.4 Crossover Point	28
2.6	Operations on Fuzzy Sets	29
2.7	Linguistic Variables	33
	2.7.1 Features of Linguistic Variables	33
2.8	Linguistic Hedges	35
2.9	Fuzzy Relations	37
	2.9.1 Compositional Rule of Inference	38
2.10	Fuzzy If–Then Rules	39
	2.10.1 Rule Forms	40
	2.10.2 Compound Rules	40
	2.10.3 Aggregation of Rules	41
2.11	Fuzzification	43
2.12	Defuzzification	44
2.13	Inference Mechanism	48
	2.13.1 Mamdani Fuzzy Inference	49
	2.13.2 Sugeno Fuzzy Inference	50
	2.13.3 Tsukamoto Fuzzy Inference	53
2.14	Worked Examples	54
2.15	MATLAB® Programs	61
	References	61
3	**Fuzzy Systems and Applications**	**65**
3.1	Introduction	65
3.2	Fuzzy System	66
3.3	Fuzzy Modelling	67
	3.3.1 Structure Identification	67
	3.3.2 Parameter Identification	70
	3.3.3 Construction of Parameterized Membership Functions	70
3.4	Fuzzy Control	75
	3.4.1 Fuzzification	75
	3.4.2 Inference Mechanism	76
	3.4.3 Rule Base	78
	3.4.4 Defuzzification	80
3.5	Design of Fuzzy Controller	81
	3.5.1 Input/Output Selection	82
	3.5.2 Choice of Membership Functions	82
	3.5.3 Creation of Rule Base	82
	3.5.4 Types of Fuzzy Controller	83
3.6	Modular Fuzzy Controller	97
3.7	MATLAB® Programs	99
	References	100
4	**Neural Networks**	**103**
4.1	Introduction	103
4.2	Artificial Neuron Model	106
4.3	Activation Functions	107

4.4	Network Architecture	108
	4.4.1 Feedforward Networks	109
4.5	Learning in Neural Networks	124
	4.5.1 Supervised Learning	124
	4.5.2 Unsupervised Learning	138
4.6	Recurrent Neural Networks	149
	4.6.1 Elman Networks	150
	4.6.2 Jordan Networks	152
	4.6.3 Hopfield Networks	153
4.7	MATLAB® Programs	155
	References	156
5	**Neural Systems and Applications**	**159**
5.1	Introduction	159
5.2	System Identification and Control	160
	5.2.1 System Description	160
	5.2.2 System Identification	160
	5.2.3 System Control	161
5.3	Neural Networks for Control	163
	5.3.1 System Identification for Control Design	164
	5.3.2 Neural Networks for Control Design	165
5.4	MATLAB® Programs	179
	References	180
6	**Evolutionary Computing**	**183**
6.1	Introduction	183
6.2	Evolutionary Computing	183
6.3	Terminologies of Evolutionary Computing	185
	6.3.1 Chromosome Representation	185
	6.3.2 Encoding Schemes	186
	6.3.3 Population	191
	6.3.4 Evaluation (or Fitness) Functions	193
	6.3.5 Fitness Scaling	194
6.4	Genetic Operators	194
	6.4.1 Selection Operators	195
	6.4.2 Crossover Operators	198
	6.4.3 Mutation Operators	206
6.5	Performance Measures of EA	208
6.6	Evolutionary Algorithms	209
	6.6.1 Evolutionary Programming	209
	6.6.2 Evolution Strategies	213
	6.6.3 Genetic Algorithms	218
	6.6.4 Genetic Programming	223
	6.6.5 Differential Evolution	230
	6.6.6 Cultural Algorithm	233
6.7	MATLAB® Programs	234
	References	235

7	**Evolutionary Systems**	**239**
7.1	Introduction	239
7.2	Multi-objective Optimization	243
	7.2.1 Vector-Evaluated GA	246
	7.2.2 Multi-objective GA	247
	7.2.3 Niched Pareto GA	247
	7.2.4 Non-dominated Sorting GA	248
	7.2.5 Strength Pareto Evolutionary Algorithm	249
7.3	Co-evolution	250
	7.3.1 Cooperative Co-evolution	253
	7.3.2 Competitive Co-evolution	255
7.4	Parallel Evolutionary Algorithm	256
	7.4.1 Global GA	257
	7.4.2 Migration (or Island) Model GA	258
	7.4.3 Diffusion GA	259
	7.4.4 Hybrid Parallel GA	261
	References	262
8	**Evolutionary Fuzzy Systems**	**265**
8.1	Introduction	265
8.2	Evolutionary Adaptive Fuzzy Systems	267
	8.2.1 Evolutionary Tuning of Fuzzy Systems	268
	8.2.2 Evolutionary Learning of Fuzzy Systems	281
8.3	Objective Functions and Evaluation	287
	8.3.1 Objective Functions	287
	8.3.2 Evaluation	289
8.4	Fuzzy Adaptive Evolutionary Algorithms	290
	8.4.1 Fuzzy Logic-Based Control of EA Parameters	292
	8.4.2 Fuzzy Logic-Based Genetic Operators of EA	302
	References	303
9	**Evolutionary Neural Networks**	**307**
9.1	Introduction	307
9.2	Supportive Combinations	309
	9.2.1 NN-EA Supportive Combination	309
	9.2.2 EA-NN Supportive Combination	310
9.3	Collaborative Combinations	318
	9.3.1 EA for NN Connection Weight Training	319
	9.3.2 EA for NN Architectures	326
	9.3.3 EA for NN Node Transfer Functions	338
	9.3.4 EA for NN Weight, Architecture and Transfer Function Training	341
9.4	Amalgamated Combination	343
9.5	Competing Conventions	345
	References	351
10	**Neural Fuzzy Systems**	**357**
10.1	Introduction	357
10.2	Combination of Neural and Fuzzy Systems	359

10.3 Cooperative Neuro-Fuzzy Systems 360
10.3.1 Cooperative FS-NN Systems 361
10.3.2 Cooperative NN-FS Systems 362
10.4 Concurrent Neuro-Fuzzy Systems 369
10.5 Hybrid Neuro-Fuzzy Systems 369
10.5.1 Fuzzy Neural Networks with Mamdani-Type Fuzzy Inference System 370
10.5.2 Fuzzy Neural Networks with Takagi–Sugeno-type Fuzzy Inference System 372
10.5.3 Fuzzy Neural Networks with Tsukamoto-Type Fuzzy Inference System 373
10.5.4 Neural Network-Based Fuzzy System (Pi–Sigma Network) 377
10.5.5 Fuzzy-Neural System Architecture with Ellipsoid Input Space 380
10.5.6 Fuzzy Adaptive Learning Control Network (FALCON) 382
10.5.7 Approximate Reasoning-Based Intelligent Control (ARIC) 384
10.5.8 Generalized ARIC (GARIC) 388
10.5.9 Fuzzy Basis Function Networks (FBFN) 393
10.5.10 Fuzzy Net (FUN) 396
10.5.11 Combination of Fuzzy Inference and Neural Network in Fuzzy Inference Software (FINEST) 397
10.5.12 Neuro-Fuzzy Controller (NEFCON) 400
10.5.13 Self-constructing Neural Fuzzy Inference Network (SONFIN) 401
10.6 Adaptive Neuro-Fuzzy System 404
10.6.1 Adaptive Neuro-Fuzzy Inference System (ANFIS) 404
10.6.2 Coactive Neuro-Fuzzy Inference System (CANFIS) 407
10.7 Fuzzy Neurons 409
10.8 MATLAB® Programs 411
References 412

Appendix A: MATLAB® Basics 415

Appendix B: MATLAB® Programs for Fuzzy Logic 433

Appendix C: MATLAB® Programs for Fuzzy Systems 443

Appendix D: MATLAB® Programs for Neural Systems 461

Appendix E: MATLAB® Programs for Neural Control Design 473

Appendix F: MATLAB® Programs for Evolutionary Algorithms 489

Appendix G: MATLAB® Programs for Neuro-Fuzzy Systems 497

Index 507

Foreword

Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing, or CIS for short, is a true *magnum opus*. Authored by Dr Nazmul Siddique and Professor Hojjat Adeli, CIS is a profoundly impressive work. It breaks new ground on many levels and in many directions. It contains a wealth of information which is new, and if not new, hard to find elsewhere.

In recent years, computational intelligence (CI) has been growing rapidly in visibility and importance. CIS's coverage of CI is very thorough, very authoritative, very insightful and very reader-friendly. CIS paves the way for making courses on CI a requirement in engineering curricula.

What is computational intelligence? A bit of history is in order. The core of CI is the conception, design and utilization of intelligent systems. The concept of an intelligent system began to crystallize in the 1980s, at a time when AI was undergoing an identity crisis, moving from logic to probability theory. There were many competing methodologies, among them traditional AI, fuzzy logic, neuro-computing and evolutionary computing. Each of these methodologies had a community, with each community claiming superiority over the others. In that climate, I came to the conclusion that what was needed was a unification of the methodologies which were competing with AI, gaining strength through unity. This was the genesis of the concept of soft computing (SC) as a partnership of fuzzy logic, neuro-computing, evolutionary computing and probabilistic computing. The guiding principle that underlies SC is that generally, superior performance can be achieved when the constituent methodologies of SC are employed in combination rather than in stand-alone mode. The Berkeley Initiative in Soft Computing (BISC) was launched in 1991 with very lukewarm support of my colleagues but strong backing from the Dean of the College of Engineering, David Hodges. Today, there are 20 journals with 'soft computing' in the title.

A few years after the launch of BISC, Jim Bezdek used his influence to create a Computational Intelligence Society within the IEEE. The concept of computational intelligence is closely related to the concept of soft computing. The principal difference between SC and CI is that SC is a partnership of fuzzy logic, neuro-computing, evolutionary computing and probabilistic computing, whereas CI is a partnership of fuzzy logic, neuro-computing and evolutionary computing. There is no competition between CI and SC, but with the backing of the IEEE, CI has been growing rapidly in visibility and acceptance. An important factor in the growth of CI is that today the principal concepts and techniques which are employed in the conception, design and utilization of intelligent systems are drawn, in large measure, from CI rather than from AI. What is worthy of note is that in 1995, at about the time when CI came into existence, Professors Adeli and Hung published a book entitled *Machine Learning –*

Neural Networks, Genetic Algorithms, and Fuzzy Systems, which in effect was the first treatise on CI. In retrospect, the importance of this seminal treatise is hard to exaggerate.

CIS is a remarkable work. The introductory chapter on CIS presents the authors' perception of what CI is and what it has to offer. The introductory chapter is followed by two chapters on the basics of fuzzy set theory and fuzzy logic. The authors' exposition is succinct, insightful and reader-friendly. I am highly impressed by their exposition of fuzzy logic and its applications.

The concepts of a linguistic variable, fuzzy if–then rules and fuzzy control received a great deal of attention, with the stress on applications. The authors' discussion of the concept of a linguistic variable reminds me of the hostile criticism of what I wrote at the time, 1973, about the concept of a linguistic variable. The criticism reflected a deep-seated tradition within science and engineering – the tradition of respect for numbers and disrespect for words. What my critics did not understand is that the use of words in place of numbers opens the door to exploitation of tolerance for imprecision, and thereby reduces cost and achieves simplicity. Today, almost all applications of fuzzy logic employ the concept of a linguistic variable. Several important applications of fuzzy logic, among them fuzzy control, are discussed in detail. As in other chapters of CIS, the exposition concludes with MATLAB® programs and references.

Following their masterly exposition of the basics of fuzzy logic, the authors turn to neuro-computing and neural systems. In the three decades since its debut, neuro-computing has become a highly important body of concepts and techniques, with wide-ranging applications in system identification, simulation and adaptation. A critical event in the evolution of neuro-computing was the invention of the backpropagation algorithm, originally due to Paul Werbos in the early 1970s and independently reinvented and developed by David Rumelhart in the early 1980s. The backpropagation algorithm opened the door to a wide variety of applications of neuro-computing. Many examples of such applications are described in CIS.

The exposition of neuro-computing is followed by an equally masterly exposition of evolutionary computing. Evolutionary computing is rooted in the pioneering work of John Holland on genetic algorithms, in combination with the seminal work of Larry Fogel. In essence, evolutionary computing is systematized random search. What is surprising is that systematized random search can be so effective in optimization, and especially in global maximization. In CIS, one finds insightful expositions of various non-standard approaches to optimization, including multi-objective optimization and Pareto optimization. But what is really fascinating is what remarkable results can be achieved through the employment of John Koza's genetic programming.

I noted earlier that the guiding principle of computational intelligence is that, in general, superior performance is achieved when fuzzy logic, neuro-computing and evolutionary computing are used in combination rather than in stand-alone mode. The last part of CIS is motivated by this guiding principle. There are very informative discussions of neuro-fuzzy systems, evolutionary fuzzy systems, evolutionary neural systems and evolutionary fuzzy neural systems. Much of the information in these chapters is hard to find elsewhere. There is much that is original to the authors.

In sum, CIS is a major contribution to the literature – it is authoritative, thorough, up-to-date, insightful and reader-friendly. CIS should be on the desk of anybody who is interested in the conception, design and utilization of intelligent systems. Professors Nazmul Siddique and Hojjat Adeli (and their publisher, John Wiley & Sons Ltd) deserve our compliments and loud applause.

Lotfi A. Zadeh
UC Berkeley
December 2012

Preface

Creating intelligent systems has been of interest to scientists for many years. In the early days of science, scientists developed systems which imitated the behaviour of living organisms. A famous example is Jacque De Vaucanson's mechanical duck from 1735, which could move its head, tail and wings as well as swallow food. This gave the illusion of intelligence and delighted and amused people at the time. The whole control mechanism was based on rotating cylinders, with gudgeons used in music boxes to control the timely execution of different behaviours. The behaviour was mechanistic as the duck always showed the same behaviour according to the mechanical control system used. To generate a new behaviour, the mechanical control system had to be changed. That means, to generate different behaviours in different situations and environments, the mechanical duck needed different control systems. Rather than using different control systems for different behaviours, scientists attempted to provide different behaviours with a single control system. This posed the challenge of developing adaptive and learning systems. In the beginning, the hope of success was based only on the belief that some general laws of adaptation should exist. The endeavour went through different stages of development, such as deterministic, stochastic and adaptive. In the happy days of determinism, various mathematical and analytical tools were developed to describe and analyse systems. These methodologies were successful, especially for linear systems. Difficulties arose as soon as nonlinear factors had to be considered. New tools had to be developed for nonlinear systems.

Uncertainty was an issue to be avoided at all costs and was not addressed until the late nineteenth century. Newtonian mechanics was replaced by statistical mechanics to describe uncertainty with the help of probability theory, developed by Thomas Bayes in the eighteenth century, which continued until the late twentieth century. The gradual evolution of probability theory for the expression of uncertainty was challenged by new theories of vagueness and fuzzy set theory, developed by Lotfi Zadeh, which came into being in the latter part of the twentieth century as a measure of uncertainty. Fuzzy systems theory has proved to be a powerful tool for the approximation of nonlinear and complex systems where traditional analytic functions or numeric relations are unable to manage.

The long-suffering stage is the time taken for model development of unknown systems, which cannot even be determined experimentally. This is evidenced by the emergence of new theories of adaptivity. The possibility of developing system models under incomplete and very little *a priori* information is based on adaptation and learning. That is, a system capable of adapting and learning is to be considered intelligent. Among the many interesting mathematical and non-traditional apparatuses for adaptation and learning, neural networks are the most widely used. Various learning algorithms have been developed since the 1960s.

The problem of developing systems to satisfy specific criteria appeared at some stage due to design, technology and development constraints. The problem of optimality then became one of the key issues in developing models or systems. In fact, the problem of optimality is a central issue in science, engineering, economy and everyday life. In deterministic or stochastic processes, the criterion of optimality (i.e., the functional) should be known explicitly *a priori* with sufficient information. The conditions of optimality only define local extrema. If the number of such extrema is large, the problem of finding the global extremum becomes a complex one. Various conventional mathematical and derivative-based optimization techniques have been developed over the past decades. Unfortunately, very often these methods cannot be applied to a wide range of problems since the functional (or the objective functions) are not analytically treatable or even not available in closed form. Further, many real-life optimization problems have constraints that either cannot be defined mathematically or are highly nonlinear implicit and discontinuous functions of design variables. These led researchers to seek stochastic methods such as evolutionary and bio-inspired optimization techniques that are capable of searching a high-dimensional space.

The inherent capability and appeal of such traditional approaches diminished as the complexity of systems grew. There arose a need for non-traditional approaches inspired by nature, such as human thinking, perception and reasoning, biological neural networks and evolution in nature.

In 1995 H. Adeli and S.L. Hung published a seminal book, *Machine Learning – Neural Networks, Genetic Algorithms, and Fuzzy Systems* (John Wiley & Sons) as the first treatise to present the three main fields of computational intelligence in a single book and demonstrate that through integration of the three emerging computing paradigms, intractable problems could be solved more effectively. Since then, research on computational intelligence has grown exponentially and the field of computational intelligence is now well established. That seminal book has inspired the current book. Computational intelligence schemes are presented in this book with the development of a suitable framework for fuzzy, neural and evolutionary computing, evolutionary/fuzzy systems, evolutionary neural systems, neuro-fuzzy systems and finally hybridization of the three basic paradigms. Applications to linear and nonlinear systems are discussed, with examples and MATLAB® exercises.

This book is designed for final-year undergraduate, postgraduate, research students and professionals. It is written at a comprehensible level for students who have some basic knowledge in calculus, differential equations and some exposure to optimization theory. Owing to the emphasis on systems and control, the book should be appropriate for electrical, control, computer, industrial and manufacturing engineering students as well as computer and information science students. With mathematical and programming references and applications in each chapter, the book is self-contained. It should also serve already practicing engineers and scientists who intend to study the field of computational intelligence and system science. In particular, it is assumed that the reader has no experience in fuzzy logic, neural networks or evolutionary computing.

The final goal of the authors is the adroit integration of three different computational intelligence technologies and problem-solving paradigms: fuzzy systems, neural networks and evolutionary computing. The book is organized in ten chapters. It includes three introductory chapters (Chapters 2, 4 and 6) on basic techniques of fuzzy logic, neural networks and evolutionary computing in order to introduce the reader to these three different computing paradigms. It then presents different applications of the three technologies in a wide range of application domains in Chapters 3, 5 and 7. Hybridization of the three technologies is an

interesting feature, which has been presented in Chapters 8, 9 and 10. Most of the book covers applications in systems modelling, control and optimization.

There have been many texts, research monographs and edited volumes published since the 1990s. There are a few books that cover some topics on the combination of the three basic technologies. They are all referenced in each chapter of this book, which the reader may find useful in further reading for research. There is no single book covering all topics on fuzzy, neural networks or evolutionary computing or their combinations that is well suited for such a wide-ranging audience, especially undergraduate, postgraduate and research students. This book is an attempt to attract all groups, putting the emphasis on a combination of the three methodologies. The book has not been written for any specific course, however, it could be used for courses in computational intelligence, intelligent control, intelligent systems, fuzzy systems, neural computing, evolutionary computing and hybrid systems. As such, the book is appropriate for beginners in the field of computational intelligence. The book is also applicable as prescribed material for a final-year undergraduate course. The book is written based on the experience of many years following pedagogical features with illustrations, step-by-step algorithms, worked examples and MATLAB® code for real-world problems. The intention of the book is not to provide a thorough discussion of all computational intelligence paradigms and methods, but to give an overview of the most popular and frequently used methods.

Acknowledgements

It is necessary to thank a number of people who have helped in many ways (unknown to them) in the preparation of this book. First of all, the authors would like to thank all staff at the learning resource centre of the Magee Campus of the University of Ulster. Special thanks go to Mr Lewis Childs, who was very kind in finding the latest and rare literature from different sources across the UK. The initial material was developed and used for an MSc course at the School of Computing and Intelligent Systems, University of Ulster. Useful feedback was received from many postgraduate students: Dr Neil Glackin, Dr Leo Galway, Dr Patric Gormley, Dr Michael McBride, Dr Julie Wall, Mr Brian McAlister, Mr Jai Verdhan Singh, Dr Erich Michols and Dr Faraz Hasan among them. Thanks go to Dr Tom Lunney, who as the MSc course director communicated many helpful suggestions for improving the course material. Professor Robert John, as an external examiner of the MSc course, was very encouraging in developing the material as a book. Thanks to Professor Liam Maguire, Head of the School of Computing and Intelligent Systems, who was very supportive of the first author during the spring semester of 2012, allowing him to be dedicated full time to manuscript preparation.

The authors would like to thank many of their collaborators: Dr Bala Amavasai, Dr Richard Mitchell, Dr Michael O'Grady, Dr Mourad Oussalah, Dr John St. Quinton, Dr Osman Tokhi, Professor Alamgir Hossain, Professor Ali Hessami, Dr Takatoshi Okuno, Professor Akira Ikuta, Professor Hydeuki Takagi, Dr Filip Ponulak, Dr David Fogel and Professor Bernard Widrow. The authors would like to thank all the staff at John Wiley & Sons Ltd associated with the publication of this book, especially Tom Carter, Anne Hunt, Eric Willner and Genna Manaog for their support and help throughout the preparation of the manuscript and the production of this book.

The first author's eldest sister passed away during the preparation of the manuscript; she would have been happy to see the book published. The first author would like to thank his wife Kaniz for her love and patience during the entire endeavour of the book, without which it would never have been published, and his daughters Oyndrilla, Opala and Orla for making no complaints during this time.

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

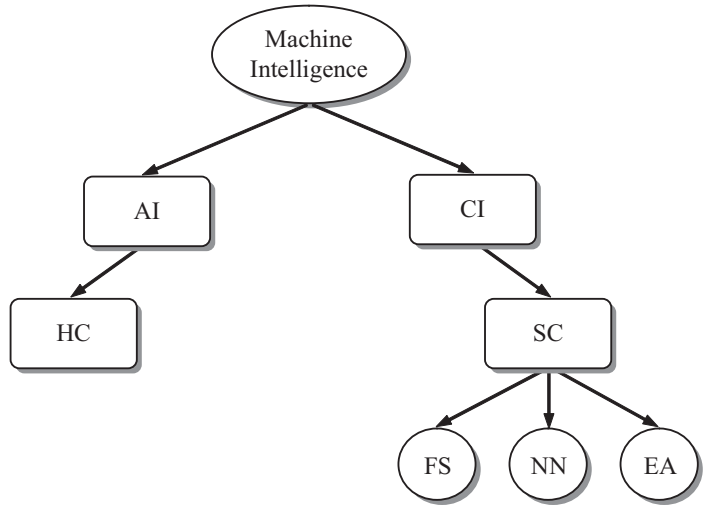

Figure 1.1 Difference between AI-HC and CI-SC

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

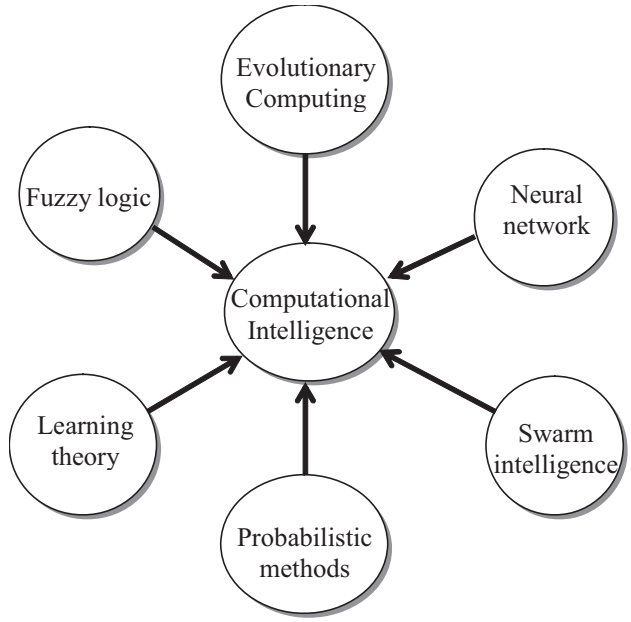

Figure 1.2 Periphery of computational intelligent methodologies

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