Computational Intelligence Assisted Design
In the Era of Industry 4.0

Yi Chen and Yun Li

Design evaluation

(Conventional CAD)

Design evolution

CRC Press
Taylor & Francis Group
A SCIENCE PUBLISHERS BOOK
Computational Intelligence Assisted Design
In Industrial Revolution 4.0

Yi Chen
Industry 4.0 Artificial Intelligence Laboratory
Dongguan University of Technology
Songshanhu, Dongguan 523808, China

and

Yun Li
Industry 4.0 Artificial Intelligence Laboratory
Dongguan University of Technology
Songshanhu, Dongguan 523808, China
Professor in Design, Manufacture and Engineering Management
Faculty of Engineering, University of Strathclyde
Glasgow G1 1XJ, UK
In recent years, many sectors of industry have experienced a paradigm shift in the way business is conducted, primarily due to the proliferating use of the internet and artificial intelligence. With this shift, Industry 4.0 (i4) is emerging as the first a-priori engineered 'industrial revolution' to bring about the 'Fourth Industrial Revolution', in which the factory floor is set to become a centre of innovation with where customization is possible at economical costs.

A common feature of all trends of i4 is the integration of several features in response to challenges of computerised decision making and big data, which are proliferated via cloud computing, and has led to a race in intelligent design, innovation, and creativity. However, i4 research and development (R&D) so far is mostly spearheaded by manufacturers, such as Siemens, and thus lacks an academic perspective and an 'intelligent design' tool commensurate with such a race. Clearly, 'market informatics' should be used to close the loop of i4 in customised smart manufacturing and smart products. Amongst the most innovative activities, design and engineering are essential to the global economy and manufacturing industries. Due to customization requirements, design efficiency, and responsiveness, demand for a smart design system has increased.

This book presents computational intelligence (CI) for smart design and manufacturing. It starts with the values and principles of CI, providing a brief history of various CI theories and the need for Computational Intelligence Assisted Design (CIAD) and engineering. Furthermore, the book introduces an important conceptual framework of a smart design and manufacturing platform for the future — CIAD and its multidisciplinary applications. Readers will find this book an easy to understand textbook for CIAD courses. It is written for senior students in science, technology, engineering and mathematics (STEM) as well as in the social sciences. Readers are expected to have fulfilled the following basic level of prerequisites: vector-matrix algebra, differential equations, control engineering, circuit analysis, mechanics, dynamics and elementary programming, e.g., MATLAB®, C/C++ or Python.

The book has been written to provide a basic understanding of CI theory and its applications. Highly mathematical arguments have been intentionally avoided. Statement proofs are provided whenever they contribute to the understanding of the subject matter presented. The book has 25 chapters with greater knowledge on
multidisciplinary applications. Special efforts have been made to provide examples at strategic points so that the reader will have a clear understanding of the subject matter discussed. The reader is encouraged to study all such worked problems carefully, which will allow him/her to obtain a deeper understanding of the topics discussed. The unsolved problems may be used as homework or quiz problems to further develop skills.

If this book is used as course text, relevant portions of the book may be used, while skipping the rest. Because of the abundance of model problems and worked examples that answer many questions, this book can also serve as a self-study book for practising engineers and industrial designers in the era of Industrial Revolution 4.0.
## Contents

_Dedication_ iii  
_Foreword_ v  
_Preface_ vii  
_Acknowledgements_ ix  
_Acronyms_ xxi

1 **Introduction** 1  
1.1 Introduction 1  
1.2 History of Computational Intelligence 1  
1.3 On the Way to Industry 4.0 13  
1.4 Need for Computational Intelligence in Design and Engineering 15  
1.5 Terms and Definitions 16  
1.6 Specialized and Application Areas 18  
1.7 Information Sources 19  
1.8 How to Use This Book 21

### Part I Hands-on Learning of Computational Intelligence

2 **Global Optimization and Evolutionary Search** 25  
2.1 Mimicking Natural Evolution 25  
2.1.1 Breading Engineering Solutions 26  
2.1.2 Conventional Computers 27  
2.1.3 Genetic Evolution—A Way to Solve Complex Optimisation Problems 27
2.2 Nondeterministic Methods for Optimization and Machine Learning
   2.2.1 Nondeterministic Hill-Climbing
   2.2.2 Simulated Annealing
   2.2.3 Parallelism C—An Essential Step Towards Global Search

2.3 The Simple Genetic Algorithm
   2.3.1 Mutation and Crossover
   2.3.2 Coding, Genes and Chromosomes
   2.3.3 The Working Mechanism
   2.3.4 GAs Transform Exponential Problems to NP-Complete Problems

2.4 Micro Genetic Algorithm

2.5 Genetic Algorithm Using Mendel’s Principles

2.6 Characteristics of Evolutionary Design Tools
   2.6.1 Advantages of Evolutionary Techniques
   2.6.2 Preparation and Conditions of Use

2.7 Tutorials and Coursework

2.8 Summary

3 Artificial Neural Networks and Learning Systems

3.1 Human Brain and Artificial Neural Networks
   3.1.1 Central Nervous System and Conventional Computer
   3.1.2 ‘Reproduce’ the Human Brain by Artificial Neural Networks
   3.1.3 Mathematical Models and Types of ANNs

3.2 ANN Design and Learning
   3.2.1 Three Steps in ANN Design
   3.2.2 Knowledge Representation
   3.2.3 Learning Process
   3.2.4 Learning Methods
   3.2.5 Learning Paradigms

3.3 Learning Algorithms
   3.3.1 Hebbian Learning
   3.3.2 Error-Correction Learning
   3.3.3 Competitive Learning
   3.3.4 Darwinian Selective Learning and Darwin Machine

3.4 Tutorials and Coursework
## 4 Fuzzy Logic and Fuzzy Systems

4.1 Human Inference and Fuzzy Logic
   4.1.1 Human Inference and Fuzzy Systems
   4.1.2 Fuzzy Sets
   4.1.3 Membership Functions

4.2 Fuzzy Logic and Decision Making
   4.2.1 Formation of Fuzzy Decision Signal
   4.2.2 Fuzzy Rule Base
   4.2.3 Measurements with Fuzzification and Scaling
   4.2.4 Defuzzification and Output Signals

4.3 Tutorial and Coursework

## Part II CIAD and Advanced Computational Intelligence Tools

5 CIAD—Computational Intelligence Assisted Design

5.1 Introduction

5.2 Optimization, Design and Intelligent Design Automation
   5.2.1 Optimal Engineering Design
   5.2.2 Difficulties with Conventional Optimization Methods
   5.2.3 Converting a Design Problem into a Simulation Problem
   5.2.4 Manual Design through Trial and Error
   5.2.5 Automate Design by Exhaustive Search?
   5.2.6 Further Requirements on a CAD Environment

5.3 Linking Intelligent Design with Manufacture

5.4 Computational Intelligence Integrated Solver

5.5 CIAD

5.6 CIAE

5.7 Intelligent Virtual Prototypes

5.8 Physical Prototyping

5.9 CIAM

5.10 System Integration

5.11 Applications

5.12 Cyber-Physical Design Integration for Industry 4.0
6 Extra-Numerical Multi-Objective Optimization 115

6.1 Introduction 115
6.2 History of Multi-Objective Optimization 117
   6.2.1 Aggregating Approaches 117
   6.2.2 Population-Based Approaches 118
   6.2.3 Pareto-Based Approaches 118
6.3 Theory and Applications 121
6.4 Multi-Objective Genetic Algorithm 122

7 Swarm Intelligence 125

7.1 Introduction 125
7.2 Particle Swarm Optimization 125
7.3 Ant Colony Optimization 127
7.4 Swarm Fish Algorithm 130
   7.4.1 Swarm Fish Algorithm with Variable Population 130
   7.4.2 Multi-Objective Artificial Swarm Fish Algorithm 133
   7.4.3 Case Study 135
   7.4.4 Conclusions 137
7.5 Swarm Bat Algorithm 138
7.6 Firefly Algorithm 140
7.7 Artificial Dolphin Swarm Algorithm 142
   7.7.1 Introduction 142
   7.7.2 Dynamic Behaviors of Dolphins 143
   7.7.3 k-Nearest Neighbor Classification 147
   7.7.4 Swarm Dolphin Algorithm 148

8 Evolving Artificial Neural Networks in a Closed Loop 151

8.1 Introduction 151
8.2 Directly Evolving a Neural Network in a Closed Loop 152
   8.2.1 Existing Architectures for Neural Control 152
   8.2.2 Architecture of a Neural Network Controller 152
   8.2.3 The Problem of Neurocontroller Design 154
8.3 Globally Optimized Design Through a Genetic Algorithm 155
   8.3.1 Difficulties with Conventional Neural Network Training Methods 155
   8.3.2 Training with a Genetic Algorithm 157
Part III CIAD for Science and Technology

11 Adaptive Bathtub-Shaped Curve

11.1 Introduction
11.2 Parameterization Method via Radial Basis Functions
11.3 Adaptive Bathtub-Shaped Failure Rate Function
11.4 Fitness Function Definition
11.5 Simulations and Discussion
11.6 Conclusions and Future Work

12 Terahertz Spectroscopic Analysis

12.1 Introduction
12.2 THz-TDS Experimental Setup Sketch
12.3 Statement of Mixture Component Determination
12.4 Fitness Function Definition
12.5 Uncertainty Studies
12.6 Empirical Studies and Discussion
12.7 Conclusions and Future Work

13 Evolving a Sliding Robust Fuzzy System

13.1 Introduction
13.2 Application of Fuzzy Logic to Sliding Mode Control
  13.2.1 Fuzzy Switching Element for the SMC System
  13.2.2 Fuzzy Gain Scheduling for the Switching Element in the SMC System
  13.2.3 Fuzzy PD SMC System with Integral Equivalent Control
13.3 Fuzzy SMC System Designs Using a GA
  13.3.1 FSMC-I System with a Fuzzy Switching Element
  13.3.2 FSMC-II System with Fuzzy Gain-Scheduling
  13.3.3 Fuzzy PD SMC System with Integral Equivalent Control
13.4 Conclusion
## Contents

**14 Space Tether for Payload Orbital Transfer**  
14.1 Introduction  
14.2 Motorized Momentum Exchange Tether  
14.3 Payload Transfer  
14.4 Tether Strength Criterion  
14.5 Payload Transfer Objective Definition  
14.6 Simulations  
14.7 Conclusion and Future Work  

**15 Structural Design for Heat Sinks**  
15.1 Introduction  
15.2 Structural Modeling  
15.2.1 Total Thermal Resistance  
15.2.2 Pressure Drop  
15.3 Experimental Setup  
15.4 Optimal Design  
15.5 Fitness Functions  
15.6 Empirical Results  
15.6.1 Simulations  
15.6.2 Verification  
15.7 Conclusions and Future Work  

**16 Battery Capacity Prediction**  
16.1 Introduction  
16.2 Adaptive Bathtub-Shaped Functions  
16.3 Battery Capacity Prediction  
16.4 Fitness Function  
16.5 Simulation Results and Discussion  
16.6 Conclusion and Future Work  

**17 Parameter Determination for Fuel Cells**  
17.1 Introduction  
17.2 Analytical Modeling  
17.3 Fitness Function  
17.4 Empirical Results and Discussion  
17.5 Conclusion and Future Work
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>CIAD Towards the Invention of a Microwave-Ignition Engine</td>
<td>321</td>
</tr>
<tr>
<td>18.1</td>
<td>Introduction</td>
<td>321</td>
</tr>
<tr>
<td>18.2</td>
<td>HCMI Design Evaluation and Virtual Prototyping Through Simulation</td>
<td>322</td>
</tr>
<tr>
<td>18.2.1</td>
<td>Models of the Emitter and Cylinder</td>
<td>322</td>
</tr>
<tr>
<td>18.2.2</td>
<td>Coupled Constraint Optimization Problem</td>
<td>324</td>
</tr>
<tr>
<td>18.3</td>
<td>Heuristic Methods and Improved GA Search</td>
<td>325</td>
</tr>
<tr>
<td>18.3.1</td>
<td>Existing Heuristic Methods Tested</td>
<td>325</td>
</tr>
<tr>
<td>18.3.2</td>
<td>Improved GA Search</td>
<td>326</td>
</tr>
<tr>
<td>18.4</td>
<td>Case Studies</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Case 1: Coupled Resonant Frequencies and Emitter Lengths</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Case 2: Coupled Frequencies, Emitter Lengths, Emitter Heights and Emitter Widths</td>
<td>330</td>
</tr>
<tr>
<td>18.5</td>
<td>Virtual Prototyping Results and Comparison</td>
<td>330</td>
</tr>
<tr>
<td>18.5.1</td>
<td>Virtual Prototyping for Case 1 with a Default Emitter</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>Search Results of the NM Method</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>Search Results of the Generic GA</td>
<td>331</td>
</tr>
<tr>
<td></td>
<td>Search Results of the PGA</td>
<td>332</td>
</tr>
<tr>
<td>18.5.2</td>
<td>Virtual Prototyping for Case 2 with an Extended Emitter</td>
<td>333</td>
</tr>
<tr>
<td></td>
<td>Search Results of the NM Simplex</td>
<td>333</td>
</tr>
<tr>
<td></td>
<td>Search Results of the GA</td>
<td>334</td>
</tr>
<tr>
<td></td>
<td>Search Results of the PGA</td>
<td>335</td>
</tr>
<tr>
<td>18.6</td>
<td>Conclusion</td>
<td>336</td>
</tr>
<tr>
<td>19</td>
<td>Control for Semi-Active Vehicle Suspension System</td>
<td>339</td>
</tr>
<tr>
<td>19.1</td>
<td>Introduction</td>
<td>339</td>
</tr>
<tr>
<td>19.2</td>
<td>Two-Degree-of-Freedom Semi-Active Suspension System</td>
<td>342</td>
</tr>
<tr>
<td>19.3</td>
<td>Sliding Mode Control with Skyhook Surface Scheme</td>
<td>345</td>
</tr>
<tr>
<td>19.4</td>
<td>Fuzzy Logic Control</td>
<td>347</td>
</tr>
<tr>
<td>19.5</td>
<td>Fuzzy Sliding Mode Control with Switching Factor $\alpha$-FaSMC</td>
<td>351</td>
</tr>
<tr>
<td>19.6</td>
<td>Polynomial Function Supervising FaSMC—An Improvement</td>
<td>351</td>
</tr>
<tr>
<td>19.6.1</td>
<td>Multi-objective Micro-GA for the Offline Step</td>
<td>352</td>
</tr>
<tr>
<td>19.6.2</td>
<td>Offline Step</td>
<td>355</td>
</tr>
<tr>
<td>19.6.3</td>
<td>Online Step</td>
<td>356</td>
</tr>
<tr>
<td>19.7</td>
<td>Road Surface Profile—Modeling of the Source of Uncertainty</td>
<td>357</td>
</tr>
<tr>
<td>19.8</td>
<td>Uncertainty Studies</td>
<td>358</td>
</tr>
<tr>
<td>19.9</td>
<td>Simulations</td>
<td>360</td>
</tr>
<tr>
<td>19.10</td>
<td>Conclusion and Future Work</td>
<td>371</td>
</tr>
</tbody>
</table>
## Part IV CIAD for Social Sciences

### 20 Exchange Rate Modeling and Decision Support 375

- **20.1 Introduction** 375
- **20.2 Exchange Rate Determination Model** 376
- **20.3 Fitness Function of Regression Modeling** 377
- **20.4 Empirical Results and Discussion** 379
- **20.5 Conclusions and Future Work** 385

### 21 Quantitative Modeling of Electricity Consumption 387

- **21.1 Introduction** 387
- **21.2 Quantitative Modeling of National Electricity Consumption** 391
- **21.3 Fitness Function** 392
- **21.4 Numerical Results** 393
- **21.5 Social, Economic and Environmental Impacts** 397
- **21.6 Conclusions and Future Work** 401

### 22 CIAD Gaming Support for Electricity Trading Decisions 403

- **22.1 Introduction** 403
- **22.2 Modelling Intelligent Market Behaviors** 404
  - **22.2.1 NETA Market Price Formulation** 404
  - **22.2.2 Generator Gaming Strategies** 405
  - **22.2.3 Supplier Gaming Strategies** 408
- **22.3 Intelligent Agents and Modeling** 408
- **22.4 Model Analysis and Verification** 411
  - **22.4.1 Small-scale Model Simulation** 412
  - **22.4.2 Large-scale Model Simulation** 414
- **22.5 Applications of the Model** 415
  - **22.5.1 Competitive Strategy** 415
  - **22.5.2 Cooperative Strategy** 416
- **22.6 Conclusions** 419

### 23 Dynamic Behavior of Rural Regions with CO₂ Emission Estimation 421

- **23.1 Introduction** 421
- **23.2 CO₂ Emission Estimation of Productive Activity** 422
23.3 Hybrid Modeling of the Functional Region 426
23.4 Fitness Function 427
23.5 Empirical Results and Discussion 429
23.6 Conclusions and Future Work 433

24 Spatial Analysis of Functional Region of Suburban-Rural Areas 435

24.1 Introduction 435
24.2 Spatial Modeling of the Functional Regions 436
24.3 Sensitive Analysis to Functional Distance 437
24.4 Fitness Function 437
24.5 Empirical Results and Discussion 438
24.6 Conclusions and Future Work 443

25 CIAD for Industry 4.0 Predictive Customization 445

25.1 Introduction 445
25.2 Customization in Industry 4.0 446
   25.2.1 CPS with Data Analytics Framework for Smart Manufacturing 447
   25.2.2 Smart Products and Product Lifecycle for Industry 4.0 448
   25.2.3 Computational Intelligence for Customized Production 449
25.3 Methodology and CIAD Approaches 451
   25.3.1 Fuzzy c-Means Approach 452
   25.3.2 Framework for Predicting Potential Customer Needs and Wants 453
25.4 Case Study 455
25.5 Discussion and Conclusion 460

References 461
Glossary 501
Index 503
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABF</td>
<td>Adaptive Bathtub Failure Rate</td>
</tr>
<tr>
<td>AABF</td>
<td>Asymmetric ABF</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ANNs</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>BFR</td>
<td>Bathtub-shaped Failure Rate</td>
</tr>
<tr>
<td>CI</td>
<td>Computational Intelligence</td>
</tr>
<tr>
<td>CIAD</td>
<td>Computational Intelligence-assisted Design</td>
</tr>
<tr>
<td>CIAE</td>
<td>Computational Intelligence-assisted Engineering</td>
</tr>
<tr>
<td>CIAM</td>
<td>Computational Intelligence-assisted Manufacture</td>
</tr>
<tr>
<td>EC</td>
<td>Evolutionary Computation</td>
</tr>
<tr>
<td>FL</td>
<td>Fuzzy Logic</td>
</tr>
<tr>
<td>FRF</td>
<td>Failure Rate Function</td>
</tr>
<tr>
<td>GAs</td>
<td>Genetic Algorithms</td>
</tr>
<tr>
<td>i4</td>
<td>Industry 4.0</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>SABF</td>
<td>Symmetric ABF</td>
</tr>
<tr>
<td>SI</td>
<td>Swarm Intelligence</td>
</tr>
<tr>
<td>STEM</td>
<td>Science, Technology, Engineering and Mathematics</td>
</tr>
<tr>
<td>μGA</td>
<td>Micro-genetic Algorithm</td>
</tr>
</tbody>
</table>
1

Introduction

1.1 Introduction

Computational intelligence (CI) is a set of nature-inspired computational approaches that offer a wealth of capabilities in complex problem solving. Compared to traditional optimization methods, the first advantage of CI is that it does not need to reformulate a problem to search a non-linear or non-differentiable space, thereby providing effective or feasible solutions to many multidisciplinary applications with or without analytical representations of the real-world problems. Another advantage of CI is its flexibility in formulating the fitness function, which can be expressed as a function of the system output. This feature is particularly appealing when an explicit objective function is difficult to obtain, where traditional models often fail to address uncertainties in ever-changing conditions or surroundings.

As shown in Figure 1.1, this book considers four paradigms of the CI family: Fuzzy Logic theory (FL), Artificial Neural Networks (ANNs), Evolutionary Computation (EC) and Swarm Intelligence (SI). In recent years, there have been significant developments in various CI-related research works, including the newly developed methods such as Swarm Intelligence, Artificial Immune Systems (AIS), Quantum Computing (QC) and DNA Computing (DNAC).

1.2 History of Computational Intelligence

The terms artificial intelligence (AI), soft computing and natural computing have been widely employed for similar or even the same area in research. John McCarthy, known as the father of AI, has noted that ‘CI’ is a more suitable name for the subject of AI, which highlights the key role played by computers in AI [Andersen (2002)].

The term ‘Computational Intelligence’ was formally introduced by Bezdek in 1992 [Bezdek (1992), Bezdek (2013)] in an attempt to attach the term computational intelligence to several activities related to the IEEE Computational Intelligence Society (IEEE CIS); however, Bezdek did not consider himself as the originator of the term CI [Bezdek (2013)]. He also discussed the actual history of CI,
which came from various sources and was attributed to various creators; his work simply helped this term become more popular in various research areas.

Since then, CI has been widely studied and utilized in academic and industrial applications under increasing demands, with a few up-to-date keywords being related to it, such as artificial intelligence, Internet of Things (IoT), big data, cloud computing, intelligent robotics and industry 4.0. In this section, we talk about the history of the CI members as listed in Section 1.1 since every member’s history contributes to the history of the CI family.

- **Fuzzy Logic Theory**

The first research on fuzzy logic can be traced back from the 1920s to 1930s, then known as ‘*infinite-valued logic*’, proposed by Łukasiewicz and Tarski. In addition, there was a very strong resurgence of interest in the 1950s and early 1960s [Pelletier (2000)]. Fuzzy logic theory, first introduced by Lotfi A. Zadeh in 1965 [Zadeh (1965a)], was based on the concept of fuzzy sets. Zadeh performed his work on fuzzy mathematics and system theory a few years before he published the concept of fuzzy sets [Zadeh (1962)] in 1962.
In 2015, Zadeh published a novel paper to introduce fuzzy logic theory in a historical perspective [Zadeh (2015)], therein marking the 50th anniversary of the publication of his first paper on fuzzy sets in 1965. Table 1.1 summarizes Zadeh’s works (some with other researchers) on fuzzy logic theory from its beginning to the present day.

Table 1.1 Summary of Zadeh’s Work on Fuzzy Logic Theory [Zadeh (2015), Zadeh, et al. (1996), Zadeh (2008)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1962</td>
<td>Fuzzy mathematics and system theory</td>
<td>[Zadeh (1962)]</td>
</tr>
<tr>
<td>1964</td>
<td>Separation theorem</td>
<td>[Zadeh (1964), Zadeh (1973a), Zadeh (1976b)]</td>
</tr>
<tr>
<td>1965</td>
<td>Fuzzy sets, membership functions, ( \alpha )-cut</td>
<td>[Zadeh (1965a)]</td>
</tr>
<tr>
<td>1966</td>
<td>Fuzzy relation, shadows of fuzzy sets</td>
<td>[Zadeh (1966), Zadeh (1971b)]</td>
</tr>
<tr>
<td>1968</td>
<td>Probability theory with fuzzy sets</td>
<td>[Zadeh (1968b)]</td>
</tr>
<tr>
<td>1968</td>
<td>Fuzzy algorithm, fuzzy Turing machine</td>
<td>[Zadeh (1968a), Zadeh (1972a), Zadeh (1975a), Zadeh (1975c), Zadeh (1975d)]</td>
</tr>
<tr>
<td>1970</td>
<td>Fuzzy decision making, fuzzy dynamical programming</td>
<td>[Bellman and Zadeh (1970), Zadeh (1976a)]</td>
</tr>
<tr>
<td>1971</td>
<td>Fuzzy graph, fuzzy finite-state machines and ( n )-dimensional unit hypercubes</td>
<td>[Zadeh (1971a)]</td>
</tr>
<tr>
<td>1972</td>
<td>Fuzzy control</td>
<td>[Zadeh (1972b), Chang and Zadeh (1972)]</td>
</tr>
<tr>
<td>1973</td>
<td>Linguistic variables</td>
<td>[Zadeh (1973a), Zadeh (1975a), Zadeh (1975c), Zadeh (1975d)]</td>
</tr>
<tr>
<td>1974</td>
<td>Fuzzy logic</td>
<td>[Zadeh (1975a), Zadeh (1975c), Zadeh (1975d), Zadeh (1974a)]</td>
</tr>
<tr>
<td>1975</td>
<td>Extension principle</td>
<td>[Zadeh (1975a), Zadeh (1975c), Zadeh (1975d)]</td>
</tr>
<tr>
<td>1975</td>
<td>Information and restriction</td>
<td>[Zadeh (1975e), Zadeh (2013)]</td>
</tr>
<tr>
<td>1976</td>
<td>Fuzzy optimization</td>
<td>[Zadeh (1976a)]</td>
</tr>
<tr>
<td>1978</td>
<td>Possibility theory and fuzzy propositions</td>
<td>[Zadeh (1978b), Zadeh (1978a)]</td>
</tr>
<tr>
<td>1979</td>
<td>Granularity</td>
<td>[Zadeh (1979a), Zadeh (1997)]</td>
</tr>
<tr>
<td>2008</td>
<td>FL-generalization</td>
<td>[Zadeh (2008)]</td>
</tr>
</tbody>
</table>
**Artificial Neural Networks**

ANNs, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain, even though the relation between this model and the biological architecture of the brain remains under debate; it is not clear to what degree ANNs mirror the brain function [Russell (2012)].

The modern era of neural network research is credited to the work done by neuro-physiologists. In 1943, McCulloch and Pitts [McCulloch and Li (1943)] published the first paper on ANNs, in which they proposed their work on a computational model of Neural Networks (NNs) based on mathematics and algorithms, called ‘threshold logic’. Basically, ANNs are biologically-inspired computational models inspired by the morphological and biophysical properties of neurons in the human brain. Such networks can simulate the way in which the human brain processes information and that is in the following two ways: (1) an NN acquires knowledge by learning, and (2) the knowledge of an NN is stored in the connections between neurons, known as synaptic weights.

Researchers have been working on Alan Turing’s unorganized machines and ANNs since 1948, particularly in the context of their potential to demonstrate intelligent behavior [Webster (2012)]. In the late 1940s, Hebb [Hebb (1949)], a psychologist, created a hypothesis of learning based on the mechanism of neural plasticity, which is now known as Hebbian Learning (It is also called Hebb’s rule, Hebb’s postulate, or cell assembly theory). Hebbian learning is considered to be a ‘typical’ unsupervised learning rule and its later variants were the early models for long-term potentiation. These ideas started being applied to computational models in 1948 with Turing’s B-type machines [Webster (2012)].

In 1954, Farley and Wesley [Farley and Clark (1954)] first proposed NN computational machines, called calculators, to simulate a Hebbian network at MIT. Other NN computational machines were created by Rochester et al. [Rochester, et al. (1956)] in 1956. In 1958, Rosenblatt [Rosenblatt (1958)] proposed the perceptron that an algorithm for pattern recognition is based on a two-layer computer learning network using simple addition and subtraction. With mathematical notations, Rosenblatt described circuitry not in the basic perceptron, such as the exclusive or circuit (XOR gate), but a circuit whose mathematical computation could not be processed until after the back-propagation algorithm was created by Werbos [Werbos (1974)] in 1974. This effectively solved the XOR problem.

In 1969, Minsky and Papert [Minsky and Papert (1969)] discovered two key issues with the computational machines that processed neural networks: (1) single-layer ANNs were incapable of processing the XOR problem, and (2) computers did not have sufficient processing power to effectively handle the long run times required by large ANNs.
Since the 1980s, research on ANNs has seen remarkable developments and have come to be widely applied. In 1986, parallel distributed processing became popular under the word ‘connectionism’. The book by Rumelhart and McClelland [Rumelhart and McClelland (1986)] provided a full exposition on the use of connectionism in computers to simulate neural processes. In 1989, a second generation of neurons integrated with a nonlinear activation function facilitated increased interest in ANNs [Hornik, et al. (1989)ANN], therein allowing nonlinear problems to be solved.

Support vector machines and other methods, such as linear classifiers, gradually overtook NNs in popularity for machine learning. However, the advent of deep learning in the late 2000s sparked renewed interest in ANNs. Table 1.2 summarizes the work on ANNs.

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1943</td>
<td>The first paper on ANNs</td>
<td>McCulloch and Pitts [McCulloch and Li (1943)]</td>
</tr>
<tr>
<td>1948</td>
<td>Alan Turing’s unorganized machines and ANNs</td>
<td>Webster [Webster (2012)]</td>
</tr>
<tr>
<td>1949</td>
<td>Hebbian learning</td>
<td>Donald Hebb [Hebb (1949)]</td>
</tr>
<tr>
<td>1954</td>
<td>Computational machines for Hebbian networks</td>
<td>Farley and Wesley [Farley and Clark (1954)]</td>
</tr>
<tr>
<td>1954</td>
<td>Neural-analog reinforcement systems</td>
<td>Minsky [MinskyReinforcement (1954)]</td>
</tr>
<tr>
<td>1956</td>
<td>Cell assembly theory</td>
<td>Rochester et al. [Rochester, et al. (1956)]</td>
</tr>
<tr>
<td>1958</td>
<td>Perceptrons</td>
<td>Rosenblatt [Rosenblatt (1958), Rosenblatt (1962)]</td>
</tr>
<tr>
<td>1982</td>
<td>Hopfield network</td>
<td>Hopfield [Hopfield (1982)]</td>
</tr>
<tr>
<td>1986</td>
<td>Connectionism</td>
<td>Rumelhart and McClelland [Rumelhart and McClelland (1986)]</td>
</tr>
<tr>
<td>1989</td>
<td>The second generation of neurons</td>
<td>Hornik et al. [Hornik, et al. (1989)ANN]</td>
</tr>
<tr>
<td>2000s</td>
<td>Neural networks in machine learning</td>
<td>Russell [Russell (2012)]</td>
</tr>
</tbody>
</table>

- **Evolutionary Computation**

In the 1950s, researchers started to work on evolutionary systems with the idea that evolution could be utilized as a tool for engineering problem optimizations, where the overall idea was to evolve a population of candidate solutions to a given problem using evolutionary operators inspired by natural genetic variation and natural selection. As also mentioned in the Fogel’s work [Fogel (2002)], a comprehensive series of simulations of evolutionary processes, such as simulation of genetic systems, varying effects of linkage, epistasis, rates of reproduction and additional factors on the rates of advance under selection, the genetic variability of a population and other statistics, were carried out by Fraser [Fraser (1957)],
Fraser (1957), Fraser (1958), Fraser (1960a), Fraser (1960b), Fraser (1960c), Fraser (1962), Fraser (1968), Fraser (1970) and collaborated with others in this area [Fraser and Burnell (1967), Fraser and Burnell (1967), Fraser, et al. (1966), Fraser and Hansche (1965), Barker (1958a), Barker (1958b), Gill (1963), Gill (1965), Martin and Cockerham (1960), Crosby (1960), Crosby (1963), Crosby (1973), Justice and Gervinski (1968), Fogel and Fraser (2000), Wolpert and Macready (1997)], such as computer models in genetics. In 1958 and 1959, Friedberg and his colleagues [Friedberg (1958), Friedberg, et al. (1959)] evolved a learning machine with language code. In the mid-1990s, Fogel’s work [Fogel (1998)] provided a historical review of the efforts in evolutionary algorithms from the early 1950s to the early 1990s.

Since then, evolutionary computation (EC), or evolutionary algorithms (EAs), have become popular for optimization, machine learning and solving design problems. EC utilizes simulated evolution to search for optimal solutions to determine optimal parameters of complex problems. In this book, we apply the term EC in all the chapters. There are many different types of evolutionary algorithms; and this book introduce 3 types: GAs, evolution strategies and evolutionary programming [De Jong, et al. (1997), Whitley (2001), Melanie (1998), Sumathi and Paneerselvam (2010)].

1. Genetic Algorithms

Genetic algorithms (GAs) are a family of computational methods inspired by Darwin’s evolutionary theory and initialized by Bremermann [Bremermann (1958)] in 1958. One of the most popular works of GA was Holland’s book Adaptation in Natural and Artificial Systems in the early 1970s [Holland (1975)]. His work originated with studies of cellular automata, which he conducted along with his colleagues and his students at the University of Michigan [De Jong (1975), Holland (1968), Goldberg (1989)]. Holland introduced a formalized framework for predicting the quality of the next generation of evolution, known as Holland’s Schema Theorem [Holland (1975), Holland (1968)]. Research in GAs remained largely theoretical until the middle of the 1980s, when the first international conference on GAs was held in Pittsburgh, Pennsylvania, USA. Since then, more studies have focused on GAs [Deb and Kumar (1995), Nix and Vose (1992), Vose and Liepins (1991), Rogers and Priddatugel-Bennett (1999)] and a few further developed GA algorithms have been proposed [Kotani, et al. (2001)], such as micro-GAs [Goldberg (1989), Krishnakumar (1989), Chen and Song (2012)] and Mendel-GAs [Chen and Zhang (2013)]. Table 1.3 provides a brief literature review on the research of Evolutionary Computation and GAs.
Table 1.3 Summary of Research on Evolutionary Computation and GAs

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1957</td>
<td>Introduction to simulation of genetic systems and effects of linkage</td>
<td>Fraser [Fraser (1957), Fraser (1957)]</td>
</tr>
<tr>
<td>1958–1959</td>
<td>Learning machine</td>
<td>Friedberg et al. [Friedberg (1958), Friedberg et al. (1959)]</td>
</tr>
<tr>
<td>1958</td>
<td>Monte Carlo analyses of genetic models</td>
<td>Fraser [Fraser (1958)]</td>
</tr>
<tr>
<td>1958</td>
<td>Initialization of GA</td>
<td>Bremermann [Bremermann (1958)]</td>
</tr>
<tr>
<td>1958</td>
<td>Selection between alleles at an autosomal locus</td>
<td>Barker [Barker (1958a)]</td>
</tr>
<tr>
<td>1958</td>
<td>Selection between alleles at a sex-linked locus</td>
<td>Barker [Barker (1958b)]</td>
</tr>
<tr>
<td>1960</td>
<td>5-linkage, dominance and epistasis</td>
<td>Fraser [Fraser (1960a)]</td>
</tr>
<tr>
<td>1960</td>
<td>Effects of reproduction rate and intensity of selection</td>
<td>Fraser [Fraser (1960c)]</td>
</tr>
<tr>
<td>1960</td>
<td>High speed selection studies</td>
<td>Martin and Cockerham [Fraser (1960c)]</td>
</tr>
<tr>
<td>1965</td>
<td>Major and minor loci</td>
<td>Fraser and Hansche [Fraser and Hansche (1965)]</td>
</tr>
<tr>
<td>1967</td>
<td>Inversion polymorphism</td>
<td>Fraser and Burnell [Fraser and Burnell (1967)]</td>
</tr>
<tr>
<td>1967</td>
<td>Models of inversion polymorphism</td>
<td>Fraser and Burnell [Fraser and Burnell (1967)]</td>
</tr>
<tr>
<td>1968</td>
<td>Simulation of the dynamics of evolving biological systems</td>
<td>Justice and Gervinski [Justice and Gervinski (1968)]</td>
</tr>
<tr>
<td>1968</td>
<td>The evolution of purposive behavior</td>
<td>Fraser [Fraser (1968)]</td>
</tr>
<tr>
<td>1968</td>
<td>Schema theory</td>
<td>Holland [Holland (1968), Holland (1975)], Fogel and Ghozeil [Fogel and Ghozeil (1997)]</td>
</tr>
<tr>
<td>1970</td>
<td>Computer models in genetics</td>
<td>Fraser [Fraser (1970)]</td>
</tr>
<tr>
<td>1975</td>
<td>The first book on GAs</td>
<td>Holland [Holland (1975)]</td>
</tr>
<tr>
<td>1975</td>
<td>Further study of GAs, block hypothesis</td>
<td>DeJong [De Jong (1975)], Goldberg [Goldberg (1989)]</td>
</tr>
<tr>
<td>1992</td>
<td>Modeling GAs with Markov chains</td>
<td>Nix and Vose [Nix and Vose (1992)]</td>
</tr>
<tr>
<td>1994</td>
<td>Co-operative co-evolutionary GAs</td>
<td>Potter and De Jong [Potter and De Jong (1994)]</td>
</tr>
<tr>
<td>1995</td>
<td>Simulated binary crossover</td>
<td>Deb [Deb and Kumar (1995)]</td>
</tr>
<tr>
<td>1997</td>
<td>No free lunch theorems for optimization</td>
<td>Wolpert and Macready [Wolpert and Macready (1997)]</td>
</tr>
<tr>
<td>1998</td>
<td>Evolutionary computation: the fossil record</td>
<td>Fogel [Fogel (1998)]</td>
</tr>
<tr>
<td>1999</td>
<td>Steady-state GAs</td>
<td>Rogers [Rogers and Priddatogel-Bennett (1999)]</td>
</tr>
<tr>
<td>2000</td>
<td>Running races with Fraser’s recombination</td>
<td>Fogel and Fraser [Fogel and Fraser (2000)]</td>
</tr>
<tr>
<td>2001</td>
<td>Variable-length-chromosome GAs</td>
<td>Kotani et al. [Kotani et al. (2001)]</td>
</tr>
<tr>
<td>2013</td>
<td>Mendel-GAs</td>
<td>Chen [Chen and Zhang (2013)]</td>
</tr>
</tbody>
</table>

Particularly, in 1992, Koza [Koza (1992), Koza (1994)], utilized GAs to evolve programs to perform certain tasks in a process called genetic programming (GP). GP is a technique of enabling a GA to search a potentially infinite space of computer programs rather than a space of fixed-length solutions to a combinatorial optimization problems. These programs often take the form
of Lisp symbolic expressions, called *S-expressions*. The idea of applying GAs to S-expressions, rather than combinatorial structures, is due originally to the work of Fujiki and Dickinson [Fujiko and Dickinson (1987)] and was brought to prominence through the work of Koza [Koza (1992)]. Table 1.4 lists a few works on GP in recent years.

2. **Evolution Strategy**

In the 1960s, Rechenberg [Rechenberg (1965), Rechenberg (1973)] proposed the idea of evolution strategies (ES, *Evolutionsstrategie* in German), which were employed to optimize real-valued parameters for devices such as airfoils. This method was then further developed by Schwefel [Schwefel (1975), Schwefel (1977), Schwefel (1981), Schwefel, et al. (1991), Schwefel (1995)]. Since then, the field of ES has remained an active area of research and has been mostly developed independently from the other fields of EC; however, it has been interacting with the other fields of EC [Knowles and Corne (1999), Beyer and Schwefel (2002)]. Table 1.5 summarizes the works on ES.

3. **Evolutionary Programming**

In 1965, Fogel et al. [Fogel, et al. (1966)] proposed the technique of evolutionary programming (EP) in which candidate solutions were represented as finite-state machines and then evolved randomly by mutating state-transition diagrams and selecting the fittest solution. Since the middle of the 1980s, EP has been developed to solve more general tasks, including prediction problems, numerical and combinatorial optimizations and machine learning [Fogel and Ghozeil (1997), Schwefel (1981), Fogel (1991), Fogel and Atmar (1991), Fogel (1995)Machine, Fogel (1999)]. In 1992, the first annual conference on EP was held at La Jolla, CA. Since then, further conferences have been held annually. Table 1.6 summarizes the works on EP.

- **Swarm Intelligence**

In recent years, swarm intelligence has been attracting substantial attention of researchers and has been applied successfully in a variety of applications. In general, swarm intelligence addresses the modeling of the collective behaviors of simple agents interacting locally among themselves and their environment, leading to the emergence of a coherent functional global pattern [Kennedy and Eberhart (2001)]. A swarm can be defined as a group of agents cooperating with certain behavioral patterns to achieve certain goals. From the computational point of view, swarm intelligence models are computing
Table 1.4 Summary of Research on GP [GPbib (2017), GPorg (2007), Langdon and Gustafson (2010)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>GP, the first paper on GP</td>
<td>Koza [Koza (1992)]</td>
</tr>
<tr>
<td>1994</td>
<td>GP II</td>
<td>Koza [Koza (1994)]</td>
</tr>
<tr>
<td>1999</td>
<td>GP III</td>
<td>Koza et al. [Koza, et al. (1999)]</td>
</tr>
<tr>
<td>2001</td>
<td>Genetic-based machine learning (GBML) and GP</td>
<td>Sette and Boullart [Sette and Boullart (2001)]</td>
</tr>
<tr>
<td>2003</td>
<td>GP IV</td>
<td>Koza et al. [Koza, et al. (2003)]</td>
</tr>
<tr>
<td>2008</td>
<td>A field guide to GP</td>
<td>Poli et al. [Poli, et al. (2008)]</td>
</tr>
</tbody>
</table>

Table 1.5 Summary of Research on ES [Melanie (1998), Schwefel, et al. (1991), Beyer and Schwefel (2002)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1965</td>
<td>First proposed ES, (1+1)-ES, (µ+1)-ES</td>
<td>Rechenberg [Rechenberg (1965), Rechenberg (1973)]</td>
</tr>
<tr>
<td>1975</td>
<td>Further development, (µ+λ)-ES</td>
<td>Schwefel [Schwefel (1975), Schwefel (1977), Schwefel (1981), Schwefel, et al. (1991), Schwefel (1995)]</td>
</tr>
<tr>
<td>1996</td>
<td>Derandomized mutation step</td>
<td>Hansen et al. [Hansen and Ostermeier (1996)]</td>
</tr>
<tr>
<td>1996</td>
<td>Mutation of ES using Markov chains</td>
<td>Rudolph [Rudolph (1996)]</td>
</tr>
<tr>
<td>1996</td>
<td>Hybrid of GA and ES</td>
<td>Smith et al. [Smith and Fogarty (1996)]</td>
</tr>
<tr>
<td>1999</td>
<td>ES for multi-objective optimization</td>
<td>Knowles [Knowles and Corne (1999)]</td>
</tr>
<tr>
<td>2001</td>
<td>Success rule</td>
<td>Rechenberg [Rechenberg (1965), Rechenberg (1973)], Rudolph [Rudolph (2001)]</td>
</tr>
</tbody>
</table>

Table 1.6 Summary of Research on EP

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1966</td>
<td>The first book on EP</td>
<td>Fogel et al. [Fogel, et al. (1966)]</td>
</tr>
<tr>
<td>2008</td>
<td>Unbiased evolutionary programming (UEP)</td>
<td>MacNish and Yao [MacNish and Yao (2008)]</td>
</tr>
</tbody>
</table>

algorithms that are useful for addressing distributed optimization problems. As introduced in Table 1.7, a number of researchers have been investigating and proposing various models of swarm intelligence, among which the most widely known models include ant colony optimization (ACO) [Colorni, et al.(1992), Colorni, et al.(1992), Dorigo, et al. (1996)], particle swarm optimization (PSO) [Kennedy and Eberhart (1995), Del Valle, et al. (2008)] and artificial bee colony (ABC) [Seeley (1996), Teodorović, et al (2006)].
Table 1.7 Summary of Research on Swarm Intelligence [Lim and Jain(2009), Banks, et al. (2007)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>Modified particle swarm optimizer</td>
<td>Shi and Eberhart [Shi and Eberhart (1998)]</td>
</tr>
<tr>
<td>2000</td>
<td>Social insect behavior</td>
<td>Bonabeau [Bonabeau, et al. (2000)]</td>
</tr>
<tr>
<td>2002</td>
<td>Canonical particle swarm</td>
<td>Clerc et al. [Clerc and Kennedy (2002)]</td>
</tr>
<tr>
<td>2003</td>
<td>Artificial fish swarm algorithm</td>
<td>Li [Li (2003)]</td>
</tr>
<tr>
<td>2004</td>
<td>Fully informed particle swarm</td>
<td>Mendes [Mendes, et al. (2004)]</td>
</tr>
<tr>
<td>2005</td>
<td>Fuzzy bee system</td>
<td>Teodorovic [Teodorovic and Dell’orco ]</td>
</tr>
<tr>
<td>2005</td>
<td>Virtual bee algorithms</td>
<td>Yang [Yang (2005)]</td>
</tr>
<tr>
<td>2008</td>
<td>Firefly algorithm</td>
<td>Yang [Yang (2008)]</td>
</tr>
<tr>
<td>2010</td>
<td>Bat algorithm</td>
<td>Yang [Yang (2010)]</td>
</tr>
<tr>
<td>2010</td>
<td>Firework algorithm</td>
<td>Tan and Zhu [Tan and Zhu (2010)]</td>
</tr>
<tr>
<td>2015</td>
<td>Artificial dolphin swarm algorithm</td>
<td>Chapter 7.7</td>
</tr>
</tbody>
</table>

- **Artificial Immune Systems**

Immune systems are highly distributed, adaptive and self-organizing in nature; they also maintain a memory of past encounters and have the ability to continually learn about new encounters. From the computational point of view, the immune system has much to offer by means of inspiring researchers. Inspired by biological immune systems, artificial immune systems (AIS) have emerged during the last decade and are used by researchers to design and build immune-based models for a variety of applications. An AIS can be defined as a computational paradigm that is inspired by theoretical immunology, observed immune functions, principles and mechanisms [deCastro and Timmis (2003)].

The AIS emerged in the middle of the 1980s through articles authored by Farmer et al. [Farmer and Packard (1986)] and Bersini and Varela [Bernstein and Vazirani (1991)] on immune networks. However, it was only in the middle of the 1990s that AIS became an independent field [deCastro and Timmis (2003), de Castro and Von Zuben (1999)]. Forrest et al. and Kephart et al. [Kephart (1994)] published their first papers on AIS in 1994 and Dasgupta conducted extensive studies on negative selection algorithms. Hunt and Cooke [Hunt and Cooke (1996)] began working on immune network models in 1995. The first book on AIS was edited by Dasgupta in 1999 [Dasgupta (1999)]. In 2000, Timmis et al. [Timmis, et al. (2000)] developed an immune network theory-inspired AIS based on work undertaken by Hunt and Cooke [Hunt and Cooke (1996)], in which the proposed AIS consisted of a set of B-cells, links between those B-cells and cloning and mutation operations performed on the B-cell objects. This AIS was tested on the Fisher Iris dataset.
1.2 History of Computational Intelligence

[Fisher (1936)], with some encouraging results. Timmis and Neal continued this work and made some improvements [Timmis and Neal (2001)]. In 2002, De Castro and Timmis proposed a framework for AIS [deCastro and Timmis (2002)] that provides a representation to create abstract models of immune organs, cells and molecules; a set of affinity functions to quantify the interactions of these ‘artificial elements’; and a set of general-purpose algorithms to govern the dynamics of the AIS. In 2008, Dasgupta and Nino [Dasgupta and Nino (2008)] published a textbook on immunological computation that presents a compendium of up-to-date work related to immunity-based techniques and describes a wide variety of applications.

Currently, the new ideas along AIS lines, such as danger theory and algorithms inspired by the innate immune system, are also being explored. However, some believe that these new ideas do not yet offer any truly ‘new’ abstract over and above the existing AIS algorithms. This, however, is hotly debated and the debate provides one of the main driving forces for AIS development at the moment. Other recent developments involve the exploration of degeneracy in AIS models [Andrews and Timmis (2006), Mendao, et al. (2007)], motivated by its hypothesized role in open-ended learning and evolution [Edelman and Gally (2001), Whitacre (2010)]. Table 1.8 lists a few works on AIS.

Table 1.8 Summary of Research on AIS [deCastro and Timmis (2002), Timmis, et al. (2008), Timmis, et al. (2004), Al-Enezi, et al. (2010)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>The first paper on AIS</td>
<td>Kephart (1994)</td>
</tr>
<tr>
<td>1994</td>
<td>Negative selection</td>
<td>Forrest, et al. (1994)</td>
</tr>
<tr>
<td>2000</td>
<td>AIS consisting of a set of B-cells</td>
<td>Timmis, et al. (2000)</td>
</tr>
<tr>
<td>2001</td>
<td>Sparse distributed memory (SDM) using immune system metaphor</td>
<td>HartRoss and Ross (2001)</td>
</tr>
</tbody>
</table>

• Quantum Computing

Quantum computing (QC) can be defined as a paradigm that exploits a computational model relying on the principles of quantum mechanics [Furia (2006)]. The introduction of quantum mechanical computational models was based on the computational difficulty of simulating quantum systems with a classical computer. The first works were performed by Manin [Manin (1980), Manin (1999)]
and Feynman [Feynman (1982)] in the early 1980s. Table 1.9 gives a brief review of the research on QC.

The QC concerns theoretical computation systems that directly utilize quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. It seems obvious that QC will largely contribute to the engineering goals of CI by applying it to various CI systems to speed up the computational process. However, it is indeed very difficult to design quantum algorithms for solving certain CI problems that are more efficient than existing classical algorithms for the same purpose.

Table 1.9 Summary of Research on QC [Furia (2006), Nielsen and Chuang (2010), Ying (2010)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>early 1980s</td>
<td>The first work on QC</td>
<td>Manin [Manin (1980), Manin (1999)], Feynman</td>
</tr>
<tr>
<td>mid 1980s</td>
<td>Quantum mechanical computational devices</td>
<td>Feynman [Feynman computers (1985), Feynman computers (1986)], Deutsch [Deutsch (1985)]</td>
</tr>
<tr>
<td>mid 1980s</td>
<td>Applying quantum models to cryptography</td>
<td>Bennett and Brassard [Bennett and Brassard (1984)], Wiesner [Wiesner (1983)]</td>
</tr>
<tr>
<td>early 1990s</td>
<td>Deutsch-Jozsa algorithm</td>
<td>Deutsch and Jozsa [Deutsch and Jozsa (1992)]</td>
</tr>
<tr>
<td>1993</td>
<td>Quantum complexity theory</td>
<td>Bernstein and Vazirani [Bernstein and Vazirani (1993), Bernstein and Vazirani (1997)]</td>
</tr>
</tbody>
</table>

• DNA Computing

DNA (deoxyribonucleic acid) can be found in every cellular organism as the storage medium for genetic information. It is composed of units called nucleotides, which are distinguished by the chemical group, or base, attached to them. The four bases are adenine, guanine, cytosine and thymine, abbreviated as A, G, C, and T. The single nucleotides are linked together end-to-end to form DNA strands.

DNA computing (DNAC) is an area of natural computing based on the idea that molecular biology processes can be used to perform arithmetic and logic operations on information encoded as DNA strands [Kari, et al. (2010)]. As a new computing paradigm, DNAC has advantages for addressing complex problems: (1) DNA codes and DNA molecular operations are suitable for
representing complex information, (2) DNAC can reduce computation time due to its parallel nature, and (3) DNA molecules have a huge storage capacity that provides a benefit for solving large-scale, multi-variable problems. However, in comparison with conventional approaches, DNAC has certain disadvantages, such as being inconvenient, un-scalable and expensive. Therefore, it is difficult to solve practical engineering problems using DNA molecules. Hence, with the help of digital computers, Garzon et al. simulated a virtual test tube and reproduced Adleman’s experiment using electronic DNA [Garzon, et al. (1999)].

On the other hand, theoretical research on DNAC includes designing potential experiments for solving various problems through DNA manipulation. Descriptions of such experiments include the satisfiability problem [Lipton (1995)], breaking the data encryption standard [Boneh, et al. (1996)], expansions of symbolic determinants [Leete, et al. (1999)], matrix multiplication [Oliver (1996)] and the bounded post correspondence problem [Kari, et al. (2000)]. Table 1.10 summarizes the work on DNAC.

Table 1.10 Summary of Research on DNAC [Pisanti (1997), Xu and Tan (2007), Bakar and Watada (2008)]

<table>
<thead>
<tr>
<th>Year</th>
<th>Work</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>The first DNA computing experiment</td>
<td>Adleman [Adleman (1994), Adleman (1998)]</td>
</tr>
<tr>
<td>1995</td>
<td>DNA solution of hard computation problems</td>
<td>Lipton [Lipton (1995)]</td>
</tr>
<tr>
<td>2004</td>
<td>Genetic algorithms in DNA computing</td>
<td>Li et al. [Li, et al. (2004)]</td>
</tr>
</tbody>
</table>

1.3 On the Way to Industry 4.0

The industrial revolution began in Great Britain and most of the important technological innovations were British. The earliest use of ‘industrial revolution’ was in a letter written on 6 July 1799 by French envoy Louis-Guillaume Otto, announcing that France had entered the race to industrialise [Francois (1996)]. Figure 1.2 shows the brief history of the industrial revolution.

- The first industrial revolution marked the transition to new manufacturing processes in the period from about 1760 to sometime between 1820 and 1840. The commencement of the first industrial revolution is closely linked to a small number of innovations, such as textiles, steam power, iron making and the invention

- The first industrial revolution evolved into the second industrial revolution in the transition years between 1840 and 1870, when technological and economic progress continued with the increasing adoption of steam transport (telephone, light bulb, phonograph, the internal combustion engine, railways, boats and ships, etc.), the large-scale manufacture of machine tools and the increasing use of machinery in steam-powered factories [Wikipedia (2018)].

- The third industrial revolution is the digitisation of design and manufacturing in a sustainable era of distributed capitalism, ushering the technologies of the internet, green electricity and 3-D printing, etc. [Rifkin (2015)].

- The fourth industrial revolution, also known as Industry 4.0 (i4), is the current trend of automation and data exchange in manufacturing technologies, which include cyber-physical systems, the Internet of things, cloud computing, robotics, artificial intelligence, nanotechnology, biotechnology, etc. The i4 has begun and offers attractive opportunities to industrial companies [Wikipedia (2018)].

![The Industrial Revolution](image)

**Fig. 1.2** The industrial revolution

Today, i4 refers to the industrial value chain and technological evolution towards smart manufacturing, with associated concepts of networked embedded systems, cyber-physical systems (CPS) [Sha and Gopalakrishnan (2009), Kim and Kumar (2012)], smart factory, the Internet of Things (IoT), the Internet of Services (IoS), ‘Internet+’, and ‘5G’ telecommunications, to name but a few [Flores, et al. (2015)], as depicted in Figure 1.3. With i4, design and manufacture are currently shifting to such a new paradigm, targeting innovation, lower costs, better responses to customer needs, optimal solutions, intelligent systems and alternatives towards on-demand production. All these trends have in common the integration of several features at the same place as a response to challenges of computerised decision making and big data, which are proliferated by Internet and cloud com-
1.4 Need for Computational Intelligence in Design and Engineering

Research shows that manufacturers using digital prototyping build half the number of physical prototypes as the average manufacturer, get to market 58 days faster than average and experience 48 per cent lower prototyping costs [Aberdeen (2006)]. This is achieved even without using the techniques of CI-assisted design and engineering (CIAD). This book introduces the conceptual CIAD frameworks and their multidisciplinary applications, with their objectives of shorter times and reduced costs of potentially 10–50 per cent. This book contributes to a driving force in the
In the era of industry 4.0, reflecting one of the most innovative activities in businesses, design and creation are essential to the global economy and manufacturing industries. A rising demand of a smart design system is seen in increased requirements of customization and flexibility, design efficiency and responsiveness, and cost-effectiveness of products and their manufacturing. These requirements become urgent in the international race to the next ‘industrial revolution’, as highlighted by the German ‘Industrie 4.0’ initiative that aims at upgrading the entire manufacturing value chain by turning the factory floor into an innovation centre capable of mass customization.

“The term ‘Industrie 4.0’ was revived in 2011 at the Hannover Fair. In October 2012, the Working Group on Industry 4.0 presented a set of Industry 4.0 implementation recommendations to the German federal government. The Industry 4.0 workgroup members are recognized as the founding fathers and driving force behind Industry 4.0.” [Wikipedia (2018)]

As this book spans both theoretical and applicable research, we foresee not only a clear potential impact on the ever-expanding needs of the intelligent design, engineering and manufacturing fields but also a methodological impact on a wide range of multi-disciplinary fields studied by universities and industrial partners such as science, technology, engineering and mathematics (STEM); life science; and social science. Beneficiaries in academia include researchers and students working on robotics, engineering design and manufacture, dynamics and control, big data systems and related professional circles in a variety of disciplines.

This book provides new insights into economic growth through the innovation of design and manufacture automation and into possible ways of increasing economic activities. It will contribute to developments in knowledge integration with industrial partners and end users, thus helping to maximize the impact of the research for knowledge sharing and transfer for potential long-term collaborations.

1.5 Terms and Definitions

• **Agents:**
  In CI, an agent is an autonomous entity that acts in a given environment; the agent observes through sensors, acts upon its environment using actuators, and directs its activity towards achieving goals through learning and pre-requisite knowledge [Poole and Mackworth (2010)].

• **Chromosome:**
  All living organisms consist of cells and each cell contains a set of chromosomes (strings of DNA) that serve as a ‘blueprint’ for an organism. In CI, the term
chromosome typically refers to a candidate solution to a problem, often encoded as a bit string.

- **Alleles:**
The different possible settings for a trait (e.g., blue, brown and hazel) are called alleles. In CI, an allele in a chromosome is either 0 or 1.

- **Gene:**
A chromosome can be conceptually divided into genes, each of which encodes a particular protein. Basically, we can consider a gene as the encoding of a trait, such as hair color. Each gene is located at a particular locus (position) on the chromosome. In CI, the 'genes' are either single bits or short blocks of adjacent bits that encode a particular element of the candidate solution. Each gene controls a particular characteristic of the individual; similarly, each bit in the string represents a characteristic of the solution.

- **Fuzzy Logic:**
A type of logic using graded or qualified statements rather than statements that are strictly true or false. The results of fuzzy reasoning are not as definite as those derived by strict logic, but they cover a larger field of discourse [Zadeh (1984a)].

- **Fuzzy Modifiers:**
Fuzzy modifiers are operations that change the membership function of a fuzzy set by spreading out the transition between full membership and non-membership, by sharpening that transition, or by moving the position of the transition region [Zadeh (1984a)].

- **Fuzzy Sets:**
Fuzzy sets are sets that do not have a crisply defined membership but rather allow objects to have grades of membership from 0 to 1 [Zadeh (1965a)].

- **Linguistic Variables:**
Linguistic variables are ordinary-language terms that are used to represent a particular fuzzy set in a given problem such as ‘large’, ‘small’, ‘medium’ or ‘OK’ [Zadeh (1984a)].

- **Ultra-fuzzy Sets:**
Ultra-fuzzy sets are those whose membership function is itself fuzzy. Here, an object in the set, rather than being given a membership grade between 0 and 1, is assigned a range of possible membership grades, for example, 0.4 to 0.8, instead of 0.55 [Zadeh (1984)].

- **Genotype:**
A genotype is the specific genetic make-up of an individual in the form of DNA. Together with the environmental variations that influence the individual, it codes
for the phenotype of that individual. In CI, a genotype is a basic data structure or type [Fogel (1995)].

• **Phenotype:**
  A phenotype is the total physical appearance and constitution or a specific manifestation of a trait, such as size, eye color, or behavior that varies between individuals. In CI, a phenotype represents a solution to a problem.

• **Genotypic Algorithms:**
  Genotypic algorithms operate on strings representing the system.

• **Phenotypic Algorithms:**
  Phenotypic algorithms operate directly on the parameters of the system itself. Both EP and Evolutionary Strategies are known as Phenotypic Algorithms, whereas the GA is a Genotypic Algorithm.

• **Industry 4.0:**
  “INDUSTRIE 4.0 is identified by the German government as one of ten ‘Future Projects’ as part of its High-Tech Strategy 2020 Action Plan. The INDUSTRIE 4.0 project (aka the fourth industrial revolution) began as a marketing opportunity for Germany to establish itself as an industry lead market and technology provider. It has now subsumed into the business lexicon as a catchall covering the automation of manufacturing, machine-to-machine and machine-to-product communication, the industrial internet and technology needed for mass customisation of production.” [BDO Industry 4.0 Report - IMechE (2016)]

### 1.6 Specialized and Application Areas

Ever since the inception of CI research, CIAD have branched into many specialized and application areas such as those listed below (but not limited to):

• Energy [Chen, et al. (2013)]
• Drug Development [Xu, et al. (2012), Liu et al. (2012)]
• Economics and Finance [Chen and Zhang (2013)]
• Sustainable Development [Chen, et al. (2013), Chen, et al. (2012)]
• Societal Application [Chen and Song (2012)]
1.7 Information Sources

• Journals

1. Advanced Engineering Informatics
2. Applied Mathematics and Optimization
3. Applied Soft Computing
4. Artificial Intelligence
5. Artificial Intelligence in Medicine
6. Annals of Mathematics and Artificial Intelligence
7. Computer Sciences in Engineering
8. Computers & Industrial Engineering
9. Computers and Mathematics with Applications
10. Computational Statistics and Data Analysis
11. Engineering Applications of Artificial Intelligence
12. Evolutionary Computation
13. Evolutionary Intelligence
14. Expert Systems
15. Expert Systems with Applications
16. Fuzzy Sets and Systems
17. IERI Procedia
18. International Journal of Approximate Reasoning
19. International Journal of Intelligent Systems
20. Information Sciences
21. Information Processing and Management
22. Information and Computation
23. Journal of Automated Reasoning
24. Journal of Information Science
25. Journal of Machine Learning Research
26. Journal of Statistical Planning and Inference
27. Journal of Supercomputing
29. Knowledge-Based Systems
30. Mathematics and Computers in Simulation
31. Mathematical and Computer Modelling
32. Machine Learning
33. IEEE Transactions on Knowledge and Data Engineering
34. IEEE Transactions on Neural Networks and Learning Systems
35. IEEE Computational Intelligence Magazine
36. IEEE Transactions on Evolutionary Computation
37. IEEE Transactions on Fuzzy Systems
38. IEEE Transactions on Autonomous Mental Development
39. IEEE/ACM Transactions on Computational Biology and Bioinformatics
40. IEEE Transactions on Computational Intelligence and AI in Games
41. IEEE Transactions on Nano Bioscience
42. *IEEE Transactions on Information Forensics and Security*
43. *IEEE Transactions on Affective Computing*
44. *IEEE Transactions on Smart Grid*
45. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
46. *Natural Computing*
47. *Neural Computation*
48. *Neural Networks*
49. *Neurocomputing*
50. *Pattern Recognition*
51. *Parallel Computing*
52. *Theoretical Computer Science*

**Conferences and Workshop**
1. UK Workshop on Computational Intelligence
2. Annual IEEE Congress on Evolutionary Computation
4. IEEE Congress on Evolutionary Computation (CEC)
5. International Joint Conference on Neural Networks (IJCNN)
6. International Workshop on DNA-based Computers
7. Genetic and Evolutionary Computation (GECCO) Conference
8. International Conference on Artificial Immune Systems

**Books**
2. Evolutionary Multiobjective Optimization—Theoretical Advances and Applications Abraham and Goldberg (2005)
3. Multi-objective Optimization using Evolutionary Algorithms [Deb (2001)]

**Organizations**
1. The International Fuzzy Systems Association (IFSA)
2. The IEEE Computational Intelligence Society
3. The Online Home of Artificial Immune Systems [The Online Home of Artificial Immune Systems (2015)]
4. Quantum Artificial Intelligence Laboratory (QuAIL), NASA
5. Ant Colony Optimization [Antorg (2015)]
1.8 How to Use This Book

This textbook can be used in many different areas, including computer science, control systems and various other STEM areas. The following list gives suggested contents for different courses at the undergraduate and graduate levels based on the syllabi of many universities around the world, such as the University of Glasgow.

• Teaching Suggestions
  The material in this book can be adapted for a one-quarter or one-semester course. The organization is flexible, therein allowing the instructor to select the material that best suits the requirements and time constraints of the class.

• Before You Use This Book
  A number of chapters in this book assume at least a basic knowledge of numerical simulation, programming, dynamics, control engineering, applied mathematics, engineering design and computational intelligence. If you are unfamiliar with MATLAB®, C/C++, pseudo-code and program flowcharts, we strongly recommend that the reader attempts to access the above-mentioned pre-requisite knowledge. Our book attempts to allow researchers who want more information on more specific topics and on the numerous new methods that are constantly being developed to take advantage of emerging research opportunities. The goal of this volume is to present the varied methods in short, practically-oriented chapters that will allow readers to integrate the methods into their own research.

• Overview of Book Structure
  This book is organized into four parts:
  Part I introduces three core technologies of computational intelligence: fuzzy logic theory, ANNs and GAs.
  Part II introduces some advanced computational intelligence algorithms, such as swarm algorithms as optimization tools and a few metric indices that can be employed to assess their performance.
  Part III introduces the conceptual framework of CIAD and its applications in science and technology, such as control systems and battery capacity prediction.
  Part IV introduces some CIAD applications in social science, for example, exchange rate investigation, electricity consumption and spatial analysis for urban studies.

• Problems, Tutorials, Laboratory and Coursework
  This book provides Problems in Chapter 1, Tutorials in Chapters in Part I, and Laboratory and Coursework in the Appendices.

• Web Companion
  There is a companion website to this book. Students and practitioners are encouraged to visit http://www.i4ai.org/EA-demo/. Additional resources
will be posted on this website as they become available, including more examples of CIAD definitions, case studies, related CIAD material, illustrations and useful links.

**Problems**

1. Write an essay on the history of CI.
2. Discuss the need for CI in smart design, smart manufacture and industry 4.0.
3. Discuss at least three well-publicized systems employing CI approaches and the reasons for their use.
4. What are the CI-related tasks performed during the smart design process?
5. Define the following terms: CI, soft computing and nature-inspired computing.
Part I

Hands-on Learning of Computational Intelligence
Part I introduces three core technologies of computational intelligence: *fuzzy logic theory, artificial neural networks* and *genetic algorithms*.

This section is about learning computational intelligence and its applicability to enabling design automation within a modern engineering context. The major aims are:

- Master the powerful technology of evolutionary computing that borrows the principles of natural evolution to help with global optimization through virtual generations of digital prototyping;
- Master neural computing technology that mimics how human learning and training can achieve optimal results;
- Master fuzzy systems techniques that human beings utilize in gathering information and making decisions;
- Understand the industrial relevance of the technology in solving many traditionally practically unsolvable engineering problems, in ‘intelligently’ refining the solutions, in optimizing engineering designs and in design automation.

*Part I Learning Outcomes:*

1. Knowledge: Methods of neural network and evolutionary computing, fuzzy logic and their applications in solving smart design and engineering optimization problems.
2. Understanding: The process of natural evolution of the human species, the process of human learning in the eye-brain and central nervous system and information gathering and inference in a fuzzy manner. Why evolutionary methods are usually more thorough but usually a slower adaptation and optimization tool than the neural learning methods. How evolutionary algorithms can be used to find solutions to practical engineering problems with a global optimality. Why neural networks can be used to store and retrieve information and mimic the behavior of many engineering systems. Training strategies. How neural networks can be used both to represent and control systems; how fuzzy systems to help with learning and decision making and neural and evolutionary computing help with devising the fuzzy systems. Use and abuse of computational intelligence techniques.
References


References


Brian Saunders. 2001. NETA: A dramatic change. ELECTRA.


References


Conn, A., Scheinberg, R.K. and Vicente, L.N. 2009. Introduction to Derivative-Free Optimization, SIAM.


S.A.), Department of Applied and Computational Mathematics, University of Sheffield, Sheffield S10 2TN.


FlexTool(GA) http://www.aip.de/~ast/EvolCompFAQ/Q20_FLEXTOOL.htm.


References


References


techniques. In emph 36th Intersociety Energy Conversion Engineering Conference, July 29–August 2, Savannah, Georgia.


References


Mangeney, J. and Crozat, P. 2008. Ion-irradiated In0.53Ga0.47As photoconductive antennas for Thz generation and detection at 1.55 mm wavelength. Comptes Rendus Physique 9(2): 142–152.
References


References


References


References


References


References

Swann, P. and Prevezer, M. 1996. A comparison of the dynamics of industrial clustering in
Tam, V.K., Chen, Y.g. and Lui, K.S. 2006. Using micro-genetic algorithms to improve localization
Tan, K.C., Li, Y., Murray-Smith, D.J. and Sharman, K.C. 1995. System identification and
Tan, K.C., Li, Y., Murray-Smith, D.J. and Sharman, K.C. 1995. System identification and
linearisation using genetic algorithms with simulated annealing. First Int. Conf. on GA in Eng.
Sys.: Innovations and Appl. 1. Un. of Sheffield (accepted).
Y.H. and Tan, K.C. (eds.). Advances in Swarm Intelligence, Volume 6145 of Lecture Notes in
Computer Science. Springer Berlin Heidelberg.
and Systems 45: 135–156.
Teodorović, D. and Dell'orco, M. 2005. Bee colony optimization—A cooperative learning approach
to complex transportation problems. Advanced OR and AI Methods in Transportation,
pp. 51–60.
Teodorović, D., Lučić, P., Marković, G. and Dell Orco, M. 2006. Bee colony optimization:
Principles and Applications. pp. 151–156. 8th Seminar on Neural Network Applications in
Electrical Engineering, Neurel-2006, Belgrade.
Test Functions Index, 2017. http://infinity77.net/global_optimization/test_functions.html#.
The National Academy of Sciences (NAS). 2004. Teaching about Evolution and the Nature of
Science, National Academy of Sciences Press.
docs/reta3.pdf, July.
The Online Home of Artificial Immune Systems. 2015. www.artificial-immune-systems.org[ BDO
imeche.org/policy-and-press/reports/detail/industry-4.0-report.
Thorndike, E.L. 1898. Animal intelligence: An experimental study of the associative processes
details/animalintelligen00thoruoft.
55: 143–150.
Knowledge Based Systems 14: 121–130.
Computation in Cells and Tissues: Perspectives and Tools for Thought. Natural Computation
Series. Springer.
References

immune systems. Evolutionary Intelligence 1: 5–26.
3rd IEEE Int. Conf. on Fuzzy Systems 3: 1620–1625.
Tso, G.K.F. and Yau, K.K.W. 2007. Predicting electricity energy consumption: A comparison of
Turing, A.M. 1952. The chemical basis of morphogenesis. Phil. Trans. R. Soc. London B
Turkakar, G. and Okutucu-Ozyurt, T. 2012. Dimensional optimization of micro-channel heat sinks
systems. 2nd. IEEE Int. Conf. on Fuzzy Systems 2: 845–850.
Utkin, V.A. 1978. Sliding Modes and Their Application in Variable Structure Systems Moscow:
Nauka (in Russian) (also Moscow: Mir, 1978, in English).
Survey Paper 22(2).
Van Veldhuizen, D.A. and Lamont, G.B. 1998. Multiobjective evolutionary algorithm research:
Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-
Patterson AFB, OH.
Van Veldhuizen, D.A. 1999. Multiobjective Evolutionary Algorithms: Classifications, Analyses,
and New Innovations. Ph.D. Thesis, Department of Electrical and Computer Engineering,
Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, OH,
May 1999.
grouping problem with multiple objectives. Computers and Industrial Engineering
Victoria Furió, André Moya and Rafael Sanjuán. 2005. The cost of replication fidelity in an RNA
virus. PNAS 102: 10233–10237.
IEEE Int. Conf. on Fuzzy Systems 1: 388–393.
5: 31–44.


Webster, C.S. 2012. Evolutionary Intelligence Alan Turing’s unorganized machines and artificial neural networks: his remarkable early work and future possibilities 5: 35–43.


References


References


