52 Selected Aspects of Natural Computing

David W. Corne[1] · *Kalyanmoy Deb*[2] · *Joshua Knowles*[3] · *Xin Yao*[4]
[1]School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK
dwcorne@macs.hw.ac.uk
[2]Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India
deb@iitk.ac.in
[3]School of Computer Science and Manchester Interdisciplinary Biocentre (MIB), University of Manchester, UK
j.knowles@manchester.ac.uk
[4]Natural Computation Group, School of Computer Science, University of Birmingham, UK
x.yao@cs.bham.ac.uk

G. Rozenberg et al. (eds.), *Handbook of Natural Computing*, DOI 10.1007/978-3-540-92910-9_52,

Abstract

In this chapter we will discuss a selection of application areas in which natural computation shows its value in real-world enterprises. For the purposes of demonstrating the significant impact and potential of natural computation in practice, there is certainly no shortage of documented examples that could be selected. We present just ten applications, ranging from specific problems to specific domains, and ranging from cases familiar to the authors to highlights known well in the general natural computation community. Each displays the proven promise or great potential of nature-inspired computation in high-profile and important real-world applications, and we hope that these applications inspire both students and practitioners.

1 Introduction

The study of natural computation has borne several fruits for science, industry, and commerce. By providing exemplary strategies for designing complex biological organisms, nature has suggested ways in which design spaces can be explored and developed into innovative new products. By exhibiting examples of effective cooperation among organisms, nature has hinted at new ideas for search and control engineering. By showing how highly interconnected networks of simple biological processing units can learn and adapt, nature has paved the way for the development of computational systems that can discriminate between complex patterns and improve their abilities over time. And the list goes on.

It is instructive to note that the methods used that have been inspired by nature are far more than simply "alternative approaches" to the problems and applications that they address. In many domains, nature-inspired methods have broken through barriers in the erstwhile achievements and capabilities of "classical" computing. In many cases, the role of natural inspiration in such breakthroughs can be viewed as that of a strategic pointer, or a kind of "tiebreaker." For example, there are many, many ways that one might build complex multiparameter statistical models for general use in classification or prediction; however, nature has extensive experience in a particular area of this design space, namely neural networks – this inspiration has guided much of the machine learning and pattern recognition community toward exploiting a particular style of statistical approach that has proved extremely successful. Similar can be said of the use of immune system metaphors to underpin the design of techniques that detect anomalous patterns in systems, or of evolutionary methods for design.

Moreover, it seems clear that natural inspiration has in some cases led to the exploration of algorithms that would not necessarily have been adopted, but have nevertheless proven significantly more successful than alternative techniques. Particle swarm optimization, for example, has been found enormously successful on a range of optimization problems, despite its natural inspiration having little to do with solving an optimization problem. Meanwhile, evolutionary computation, in its earliest days, was subjected to much skepticism and general lack of attention – why should a method be viable for real-world problems when that method, in nature, seems to take millions of years to achieve its ends? What need is there for slow methods that rely on random mutation, when classical optimization has a mature battery of sophisticated techniques with sound mathematical bases? Nevertheless, evolutionary methods are now firmly established, thanks to a long series of successful applications in which their performance is unmatched by classical techniques.

The idea of this chapter is to present and discuss a collection of exemplars of the claims made in this introduction. A handful of selected applications of natural computation will be looked at, each chosen for a subset of reasons, such as level of general interest, or impact. Some classic applications will be considered, which still serve as inspirational to current practitioners, and some newer areas will be looked at, with exciting or profound prospects for the future.

The applications are loosely clustered into four themes as follows. The first theme deals with applications under the banner of "Strategies," where three examples in which natural computing methods have been used to produce novel and useful strategies for different enterprises are studied in detail. These include an evolutionary/neural hybrid method that led to the generation of an expert checkers player, the use of genetic programming (GP) to discover rules for financial trading, and the exploitation of a learning classifier system to generate novel strategies for fighter pilots. The next theme is "science and engineering" where applications that have wider significance for progress in one or more areas of science and engineering, in areas (or in ways) that may not be traditionally associated with natural computing, are considered. The two exemplars in this area are the use of multi-objective evolutionary computation for a range of areas (often in the bio and analytical sciences) for *closed-loop* optimization, and the concept of *innovization*, which exploits multi-objective evolutionary computation in a way that leads to generic design insights for mechanical engineering (and other) problems. The next theme, "logistics," exemplifies how natural computing (largely, learning classifier systems and evolutionary computation) has provided us with successful ways to address difficult logistics problems (the case of a real-world truck scheduling problem is looked at), as well as a way to design new fast *algorithms* for a range of logistics and combinatorial problems, via approaches referred to as "super-heuristics" and/or "hyper-heuristics." Finally, the theme of "Design" is considered and three quite contrasting examples are discussed. These are, in turn, antenna design, Batik pattern design, and the emerging area of software design using natural computing methods.

2 Strategies: Generating Expert Pilots, Players, and Traders

Many problems in science and industry can be formulated as an attempt to find a good strategy. A strategy is, for practical purposes, a set of rules (or an algorithm, or a decision tree, and so forth) that sets out what to do in a variety of situations. Expert game players are experts, presumably, because they use good strategies. Similar is true for good pilots, and successful stock market traders, as well as myriad other professionals who are experts in their particular domains. It may well come as a surprise to some that humans do not have the last word on good strategy – strategies can be discovered by software, which, in some cases, can outperform most or even all human experts in particular fields. In this section, three examples of applications in which strategies have been developed via natural computing techniques, respectively, for piloting fighter aircraft, for playing expert-level checkers, and for trading on the stock market will be looked at.

2.1 Discovery of Novel Combat Maneuvers

In the early 1990s and beyond, building on funding support from NASA and the USAF, a diverse group of academics and engineers collaborated to explore the automated development of

http://www.springer.com/978-3-540-92909-3

Handbook of Natural Computing
Rozenberg, G.; Baeck, Th.; Kok, J.N. (Eds.)
2012, LII, 2052 p. 332 illus., 60 illus. in color. In 4 volumes, not available separately., Hardcover
ISBN: 978-3-540-92909-3