Preface

The term evolutionary computing refers to the study of the foundations and applications of certain heuristic techniques based on the principles of natural evolution; thus the aim of designing evolutionary algorithms (EAs) is to mimic some of the processes taking place in natural evolution. These algorithms are classified into three main categories, depending more on historical development rather than major functional techniques. In fact, their biological basis is essentially the same. Hence

$$\mathrm{EC} = \mathrm{GA} \cup \mathrm{GP} \cup \mathrm{ES} \cup \mathrm{EP}$$

EC = Evolutionary Computing
GA = Genetic Algorithms, GP=Genetic Programming
ES = Evolution Strategies, EP = Evolutionary Programming

Although the details of biological evolution are not completely understood (even nowadays), there is some strong experimental evidence to support the following points.

- Evolution is a process operating on chromosomes rather than on organisms.
- Natural selection is the mechanism that selects organisms which are well-adapted to the environment to reproduce more often than those which are not.
- The evolutionary process takes place during the reproduction stage that includes mutation (which causes the chromosomes of offspring to be different from those of the parents) and recombination (which combines the chromosomes of the parents to produce the offspring).

Based upon these features, the previously mentioned three models of evolutionary computing were independently (and almost simultaneously) developed.

An evolutionary algorithm(EA) is an iterative and stochastic process that operates on a set of individuals (called a population). Each individual represents a potential solution to the problem being solved. Initially, the population is randomly generated. Every individual in the population is assigned, by means of a fitness function, a measure of its goodness with respect to the problem under consideration, which guides the search. The whole process is sketched as:

$Generate[P(0)]$
$t \leftarrow 0$
$WHILENOT \;\; Termination - Criterion$
DO
$Evaluate \;\; [P(t)]$
$P'(t) \leftarrow Select \; [P(t)]$
$P''(t) \leftarrow \;\; Apply - Reproduction - Operatorson \; [P'(t)]$
$P(t+1) \leftarrow Replaceby \; [P(t), P''(t)]$
$t \leftarrow t + 1$
END
$RETURN \;\; Best - Solution$
Skeleton of an Evolutionary Algorithm

It can be seen that the algorithm comprises three major stages: selection, reproduction, and replacement. During the selection stage, a temporary population is created in which fitter individuals have a higher number of instances than less fit ones (natural selection). The reproductive operators are applied to these individuals in this population yielding a new population. Finally, individuals of the original population are substituted by the newly created individuals. This replacement usually tries to keep the best individuals deleting the worst ones. The whole process is repeated until a termination criterion is achieved. It should be noted that EAs are heuristics, and thus they do not ensure an optimal solution.

Major differences between GA, ES and EP come from the operators they use and in general from the way they implement the three stages: selection, reproduction, and replacement. EP is closely related to ES. Unlike GA, no crossover operator is used in ES/EP. Moreover, more emphasis is placed on behavioral changes rather than on the modification of the genetic material. For this reason, the genotype in ES/EP is usually very different (e.g., real numbers for ES, and a finite automaton for EP), and mutation operators are prepared to deal with such representations.

Any EA is composed of a set of common elements in spite of its differences from other EAs as follows.

- A population of trial solutions/strings. Typically strings are composed of binary, float, or some complex structure (e.g., a tree) genes.
- A fitness function to be optimized for evaluating strings.
- Some selection/replacement mechanism in order to simulate the survival of the fittest individuals for future generations.
- Nature-inspired operators (like recombination and mutation) for changing a string into a new string.

A large number of researchers, all over the world, have been engaged in developing EC methodologies for designing intelligent decision-making systems for a variety of real-world problems. However, research articles on such topics are sparse.

The present book provides a collection of 40 articles, divided into two parts, containing new material on the theoretical aspects of EC, and demonstrating the usefulness/success of EC for various kinds of large real-world problems. Each chapter represents an article in its own right. Part I contains 23 articles dealing with various theoretical aspects of EC, while Part II contains 17 articles demonstrating the success of EC methodologies. These articles are written by leading experts from many countries.

Part I starts with the article of Vassilev et al. who studied structures of fitness landscapes to provide a suitable mathematical framework for investigating the evolvability of complex systems. Xin Yao and colleagues (in a related article in Chapter 2) demonstrated the role of search step size in approximating the landscape by using different hybrid EAs. Chapter 3 is intended to provide an introductory review of the existing work done on visualizing EAs, and to identify some of the key issues for future research.

New schemes of EAs or designing their operators are described in Chapters 4 and 5. A parameter-free GA (called PfGA) is proposed by Sawai and colleagues in Chapter 4 based on the disparity theory of evolution which exploits different mutation rates and variable population size. In Chapter 5 Droste and Wiesmann suggest guidelines for the design of genetic operators and the representation of phenotypic space to solve specific types of problem. The applicability of this concept is shown by a systematic design of a GP system for finding Boolean functions. This system is the first GP system that has reportedly found the 12 parity function.

Eiben demonstrates the utility of multi-parent reproduction with successful results in Chapter 6. The traditional debate of mutation and crossover is also considered in the light of multi-parent reproduction.

In Chapter 7 Michalewicz and Schmidt propose a test case generator which is capable of creating various test problems with different characteristics, including the dimensionality of the problem, number of local optima, number of active constraints at the optimum, topology of the feasible search space, etc. Such a test case generator is useful for analyzing and comparing different constraint-handling techniques.

Handling real-coded parameters in GAs is an important research topic now-a-days. In Chapter 8 Ono et al. propose a new crossover operator named the unimodal normal distribution crossover for real-coded GAs which works the efficiently for optimization problems with epistasis among parameters.

Branke and Schmeck provide a good survey of the literature on EC in Chapter 9 for dynamic optimization problems; and offer a classification of the same set of problems. They also suggest a new technique for this task using a multi-population structure.

In the next chapter Deb proposes a few classical techniques to identify a preferred or compromise solution by introducing a biased sharing technique to find a biased distribution of Pareto-optimal solutions in multi-objective GAs.

The results are encouraging for more complex multi-objective optimization problems.

The utility of gene expression in scalable genetic search is studied by Kargupta in Chapter 11.

Knjazew and Goldberg present an ordering messy GA in Chapter 12 that is able to solve difficult permutation problems efficiently according to the experimental results.

Global optimal solutions are not always acceptable, if they are sensitive to perturbations in the environment. In Chapter 13 Tsutsui and Ghosh suggest ways of detecting robust solutions thereby extending the utility of GAs.

In Chapter 14, EC is used by Spears and Gordon to evolve finite-state machines having an optimal number of states for better performance in resource allocation.

Chapters 15-16 try to link EC with statistical inferencing. In Chapter 15 Zhang tries to view EC as a Bayesian inference that iteratively updates the posterior distribution of a population from the prior knowledge and observation of new individuals to find an individual with the maximum posterior probability. Chapter 16 by Aizawa is an attempt to combine experimental design and EC into a single search strategy using a specific type of recombination function called a deterministic crossover operator.

The next three chapters deal with the theoretical understanding of biology and its simulation. Maley's article in Chapter 17 aims to use EAs to extend our theoretical understanding of biology, and to reunite theoretical biology with experimental biology. Kumar and Bentley present a brief survey on using embryology and genetics in developmental biology in Chapter 18. An application of two embryological techniques is also shown on evolving certain predefined letters. In Chapter 19 Ray describes an evolutionary approach to synthetic biology which inoculates the process of natural evolution in an artificial medium, and finds the natural form of the living organisms in the artificial medium. He also suggests a possible means of harnessing the evolutionary process for the production of complex computer software.

Chapters 20-22 deal with other heuristic algorithms closely related to EAs, simulating some other natural phenomena. The main goal of Chapter 20 by Glover et al. is to demonstrate the development of scatter search procedures by illustrating how they may be applied to a class of non-linear optimization problems of bounded variables. They conclude the chapter by highlighting the key ideas and research issues that offer the promise of yielding future advances. In Chapter 21 the application of several ant colony optimization techniques is demonstrated by Carbonaro and Maniezzo on a number of hard optimization problems with specific attention to a new algorithm called ANTS. In recent years, considerable interest and enthusiasm have been generated by the prospect of widespread use of intelligent agent-based systems. A co-evolutionary optimization approach for evolving agent groups for multiagent systems and an adaptive system approach are suggested by Sen et al. in Chapter 22.

Schmidhuber studies in Chapter 23 an embedded active learner that can limit its prediction to arbitrary computable aspects of spatio-temporal events using probabilistic algorithms.

In Chapter 24, the first article of Part II, Ku and colleagues demonstrate a method to combine local search and evolutionary search techniques for neural network learning to reduce the computational time.

Chapters 25-27 deal with designing analog circuits using EC techniques. Analog circuits are evolved using variable length genetic algorithms in Chapter 25 by Iba et al. Koza suggests a technique in Chapter 26 for applying genetic programming techniques for the automatic synthesis of topology and sizing for analog electrical circuits, synthesis of placement and routing for circuits, and that of synthesis of both the topology and tuning of controllers. In Chapter 27 Cohoon et al. have used EC for a physical design problem where the input to the physical design set is a logical representation of the system under design, and the output of this step is the layout of a physical package that optimally or nearly optimally realizes the logical presentation. They also discuss important requirements for evolutionary based approaches for even greater acceptance within the VLSI community.

The next two chapters (28 and 29) discuss issues related to designing communication channels. In Chapter 28 Zimmermann et al. report results on the application of EAs constrained to multi-objective, large, real-world antenna placement problems for mobile radio networks. EAs are used by Back et al. in Chapter 29 to find a routing table that increases the performance of a communication network by reducing the probability of end-to-end blocking, and is applied to a non-hierarchical network.

Scheduling by EAs is discussed in Chapters 30 and 31. Ross and colleagues present a survey with critical analysis on the application of EC in timetable scheduling problems in Chapter 30. They claim that a wide ranging investigation is needed on this problem. Chapter 31 by Dorndorf et al. describes techniques to use GAs as meta heuristics to guide an optimal design schedule decomposition sequence for solving the minimum makespan problem for job shop scheduling, resulting in shorter makespans than in other local search algorithms. A scheme for bus driver scheduling along various routes is suggested by Yoshihara in Chapter 32.

Chapters 33-34 demonstrate the usefulness of EAs for data mining problems. Alex Freitas presents an excellent survey of different EAs for data mining problems in Chapter 33. He also discusses whether the tasks of data mining and knowledge discovery in data bases will influence the design of EAs. Interactive EC is used by Teraso and Irada in Chapter 34 to get effective features from the data and inductive learning is used to acquire simple decision rules from the subset of data for data mining problems using clinical data.

Chapter 35 by Bhanu and Fonder discusses an approach to image segmentation which is guided by GAs and learns the appropriate subset and spatial combination of a collection of discriminating functions associated with image features. In this context they also suggest techniques for physics-based segmentation evaluation, novel crossover operator and fitness function, as well as a system prototype, and demonstrate experimental results on real synthetic aperture radar imagery of varying complexity.

Cao and Dasgupta propose an immunogenetics approach to recognize spectra for chemical analysis in Chapter 36. Their experimental results show the effectiveness of the approach in finding products responsible for a composite spectrum in which there are multiple, physically mixed products.

Steffen Kremer describe techniques which applied GAs to two-dimensional protein folding in Chapter 37. The results and limitations of the applicability of GAs to the problem of three-dimensional fold prediction are also presented.

Hasegawa and Fukuda suggest a method to control the regrasping motion of a four-fingered robot hand using EP in Chapter 38.

In Chapter 39 Lanzi and Riolo review recent advances and trends in learning classifier systems (LCS). These include credit assigned to rules, alternative LCS architectures like rule syntax and semantics, and the increase in number and the range of LCS.

David Fogel describes a hybrid technique in Chapter 40 for exploiting the advantages of neural networks and EC for designing a program which plays checkers at an expert level.

Kolkata, July 2002 *Ashish Ghosh*
Osaka, July 2002 *Shigeyoshi Tsutsui*

Contents

Part II