

Preface

This is a textbook on evolutionary algorithms (EAs). In preparing the proposal and the manuscript, the following questions were always kept in our minds.

- Is this book convenient for teaching, studying, and self-study, i.e., have the contents been arranged in a pedagogically sound way?
- Does this book introduce the state of the art of EAs?
- Does this book cover the contents as comprehensively as possible?
- Does this book contain specific programming codes so that the reader can use them directly?

For the first two questions, we would like to say yes. We want this textbook to be intuitive because intuitive ideas, and not strict proofs, lead to the kind of innovation we seek. We also want to build a bridge connecting the basics and the cutting edge so that our students can reach the peak quickly yet with a solid grasp of the material. Our answer to the remaining two questions is no. This textbook is neither an encyclopedia nor a cookbook. We selected only interesting and important topics and left the reader to implement the codes for the algorithms after we have deliberately removed all understanding obstacles. The reason for the latter consideration lies in the belief contained in the following saying: “Tell me and I’ll forget; show me and I may remember; let me try, and I can understand.”

EAs have attracted considerable interest in recent years. To understand how “hot” the topic is, consider the number of papers indexed by the Science Citation Index (SCI) every year. If the reader has the access to the SCI, he/she is encouraged to do a search to verify the hotness of this topic.

The reason the topic is so hot is mainly because of its effects on various problems, i.e., it really works. We will introduce various application examples to show how simple ideas can be expanded to solve complex problems.

The procedure of designing or analyzing an algorithm for solving optimization or learning problems is full of challenges, i.e., problems are difficult. This is irrelevant. We will accompany and assist the reader when necessary. We will demonstrate the basic ideas behind the scary equations and try our best to make things easy to understand. Incidentally, this cumbersome procedure is also interesting. Before and

during the procedure of climbing Mount Fuji, we thought it would be torture. While at the summit, we felt that all the sufferings were interesting.

The prerequisites of this book lie in two aspects. In the mathematics part, a basic understanding of linear algebra, multivariable calculus, probability, and statistics is necessary.¹ The other demand is derived from a proficiency in programming. We assume a firm grasp of at least one programming language, i.e., you can implement an algorithm. Since these requirements are in the common curriculum of junior undergraduate programs, senior undergraduate or graduate students should be able to read this textbook without knowledge obstacles.

The pedagogical approaches used in this text can be summarized as follows:

- Building the mansion from the bricks. We will always focus on the key elements of an algorithm because later they might become your tools.
- From specific to general. We will always discuss specific simple examples before formal expressions.
- From idea to implementation. We will always introduce the initial notions before discussing the concrete contents.
- Explaining the critical part but leaving questions unanswered deliberately. We will leave some obstacles deliberately in the context to activate your thoughts.

R.W. Hamming² gave his motto as the following statement: “The purpose of computing is insight, not numbers” in his famous book *Numerical Methods for Scientists and Engineers*. Here we would like to use it again to represent our own attitude toward EAs. As algorithm designers, we care more about the solution landscape of the problem and the corresponding search ability of the algorithms,³ although we do seek the optimal solution to the problem. From this perspective, there will be less “how-to” in this textbook for specific instructions. On the contrary, there will be plenty of “why-to”s to explain the insights and the rationale of a given algorithm so that one can generate one’s own cookbook according to these “why-to”s. We believe that’s what the reader wants!

Other teaching and self-study considerations include plenty of footnotes for mentioning easily ignored yet important points, “Suggestions for Further Reading” sections in most chapters for further study, exercises with various degrees of difficulty to deepen one’s understanding and even promote one’s own “small-scale” research.

The pedagogical relationship among the ten chapters of this textbook is illustrated by Fig. 0.1, where the dotted circles separate the ten chapters into three categories: the basics, optimization problems, and other branches. The solid lines are the suggested pedagogical path for teaching or self-study. Chapters 1 to 3 are all necessary for subsequent chapters. After that, readers can skip around depending on personal interests. Dashed lines represent the improvements, e.g., an understanding

¹ Knowledge of operations research, e.g., linear programming, nonlinear programming, and combinatorial optimization, might contribute to one’s understanding but not be necessary.

² Father of the Hamming distance, which will be used often in this textbook.

³ These terms will be explained explicitly and used throughout the book. Here they can be taken as the procedure to find the optimal solution.

of Chap. 4 might contribute to, but not be a prerequisite of, understanding Chap. 6 and *vice versa*. We hope Fig. 0.1 helps the reader form a mental map for different branches and applications of EAs.

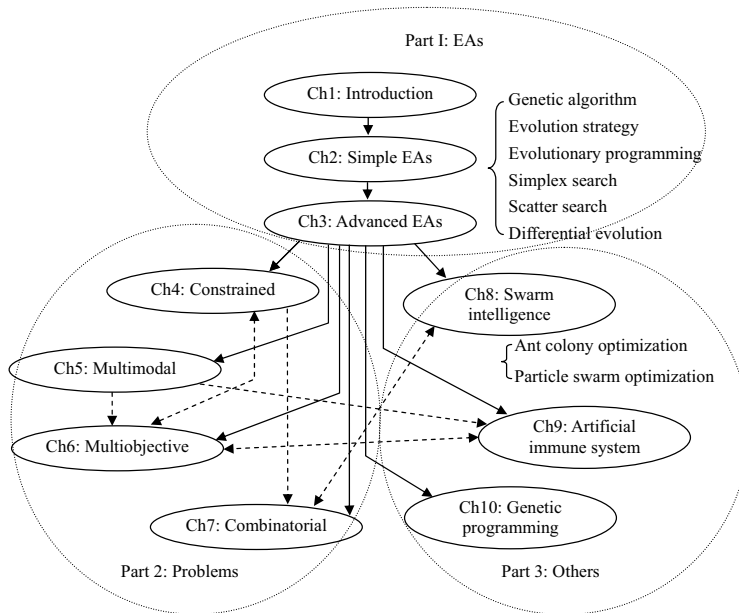


Fig. 0.1 The pedagogical relationship of the ten chapters in this textbook

This book is the result of the development of the course Evolutionary Computation and Its Applications given by the first author at Tsinghua University since 2002 and the numerous intensive courses taught by the second author around the world. Too many names require mention in the acknowledgments. But we wish to express our appreciation for our students first, especially those in the seminar Evolutionary Algorithms 2010. Without your critical comments during the seminar, we would have no appreciation of the reader's position, which is perhaps the most distinctive feature of this textbook. Many of old friends gave encourage and comments on the book, including Runwei Cheng, Baoding Liu, Kap Hwan Kim, etc. The fast response and the professional skill of Claire Protherough at Springer and Nadja Kroke at latex impressed us deeply. The final, but most important, acknowledgement belongs to Lin Wang and Eiko Gen because you are the backbones of our lives.

Beijing China,
Kitakyushu Japan,
May 2010

Xinjie Yu
Mitsuo Gen



<http://www.springer.com/978-1-84996-128-8>

Introduction to Evolutionary Algorithms

Yu, X.; Gen, M.

2010, XVI, 422 p. 168 illus., Hardcover

ISBN: 978-1-84996-128-8