Michalewicz, Zbigniew
Genetic Algorithms + Data Structures = Evolution Programs
Reviewer: David Aldous (U.C. Berkeley)

The idea of using genetic algorithms for optimization problems is so intuitively appealing that one often sees it mentioned in popular science articles. This book is a self-contained account, presupposing only basic undergraduate mathematics. Being based on various papers of the author, it is "not really a textbook" in style (e.g. no exercises), but in fact would be quite usable as a course text. The second edition adds various topics, (e.g. varying population size implementations, constraint handling techniques for the knapsack problem) and corrects typos.

The flavor of the book can be illustrated by one of the topics, the non-linear transportation problem. Mathematically, there is a "feasible" set of matrices $\mathbf{v} = (v_{ij})$, defined as the set of matrices with non-negative entries and with certain prescribed row- and column-sums. There is also given a non-negative matrix (c_{ij}) and a non-negative function $f(c, x)$. The problem is to find a feasible matrix $\mathbf{v}$ which minimizes the objective function $F(\mathbf{v}) = \sum_i \sum_j f(c_{ij}, v_{ij})$. An algorithm GENETIC-2 is proposed and studied via simulation. Here is a brief description of the algorithm, with typical parameter values in parentheses. An initial population (size 40) of feasible matrices is chosen by a simple randomized procedure. To generate an individual $\mathbf{v}$ in the next generation, choose from the current generation two individuals $\mathbf{v}^1, \mathbf{v}^2$ at random with probabilities proportional to $F(\mathbf{v}^i)$. With a certain probability (0.75) set $\mathbf{v} = \mathbf{v}^1$. With a certain probability (0.05) perform a *crossover* by setting $\mathbf{v} = \gamma\mathbf{v}^1 + (1-\gamma)\mathbf{v}^2$ for a constant γ (0.35). With the remaining probability (0.2) construct $\mathbf{v}$ from $\mathbf{v}^1$ via a *mutation* procedure, as follows. Pick a random sets R, C of rows and columns. Set $\mathbf{v} = \mathbf{v}^1$ outside $R \times C$. On $R \times C$, generate the values of $\mathbf{v}$ by a simple randomized procedure designed to give a submatrix with row- and column-sums equal to those of the submatrix of $\mathbf{v}^1$. Two different procedures are used (chance 0.5 each): one is intended to produce submatrices nearer the center of the feasible region, and the other to produce submatrices nearer the boundary of the feasible region.

This is an example of what the author calls an *evolution program*, to distinguish it from a *genetic algorithm* in which feasible solutions are explicitly coded as binary strings. Other evolution programs are described for other problems: traveling salesman, drawing directed graphs, scheduling, path planning in a mobile robot environment, machine learning.

Readers' opinions of this book are likely to be determined by their prior tastes and experience. The book succeeds well as an introduction to the subject. Its strength is in discussing the numerous available variations of the details of setting up GA/EP algorithms, and I particularly liked the style of giving one-paragraph summaries of research papers which analyze such variations. Surely any non-expert who reads the book will learn some new tricks they wouldn't think of for themselves. As a non-expert, I cannot judge how well the selection of topics covers the range of problems for which GA/EP methods may be useful (ironically, one of the most popular application areas, stock market strategies, is never mentioned). Readers with theoretical tastes may be less satisfied. The lack of visible theoretical guidelines behind choice of specific mutation/crossover operators is a little disconcerting, the book's one foray into non-elementary mathematics (the fixed-point theorem for contraction mappings) reveals hopeless confusion, and the explanations of how and why these algorithms work hardly go beyond the common-sense heuristics of the "building-block hypothesis".