

IEEE Press Book

**Modern Heuristic Optimization Techniques:
Theory and Applications to Power Systems**

Edited by

K. Y. Lee

and

M.A. El-Sharkawi

Disclaimer

The Editors are not endorsing evolution as a scientific fact, in that species evolve from one kind to another. The term “evolutionary” in the evolutionary computation (EC) simply means that the characteristics of an individual changes within the population of the same species, as observed in the nature.

Introduction

Several heuristic tools have evolved in the last decades that facilitate solving optimization problems that were previously difficult or impossible to solve. These tools include evolutionary computation, simulated annealing, tabu search, particle swarm, etc. Reports of applications of each of these tools have been widely published. Recently, these new heuristic tools have been combined among themselves and with knowledge elements, as well as with more traditional approaches such as statistical analysis, to solve extremely challenging problems. Developing solutions with these tools offers two major advantages: 1) development time is much shorter than when using more traditional approaches, and, 2) the systems are very robust, being relatively insensitive to noisy and/or missing data.

The purpose of this book is to provide with basic knowledge of evolutionary computation, and other heuristic optimization techniques, and how they are combined with knowledge elements in computational intelligence systems. Applications to power problems are stressed, and example applications are presented. The book is composed of two parts: The first part gives an overview of modern heuristic optimization techniques, including fundamentals of evolutionary computation, genetic algorithms, evolutionary programming and strategies, particle swarm optimization, ant colony search algorithm, simulated annealing, tabu search, hybrid systems of evolutionary computation, and hybrid optimization of local heuristics and dynamical trajectory.

The second part of the book gives an overview of power system applications and deals with specific applications of the heuristic approaches to power system problems, such as security assessment, generation and maintenance scheduling, economic dispatch, transmission network expansion planning, generation expansion and reactive power planning, distribution system optimization, and power plant and power system control, FACTS, and hybrid systems of heuristic methods.

Evolutionary Computation:

Natural evolution is a hypothetical population-based optimization process. Simulating this process on a computer results in stochastic optimization techniques that can often out-perform classical methods of optimization when applied to difficult real-world problems. This tutorial will provide a background in the inspiration, history, and application of evolutionary computation and other heuristic optimization methods to system identification, automatic control, gaming, and other combinatorial problems.

The objectives are to provide an overview of how evolutionary computation and other heuristic optimization techniques may be applied to problems within your domain of expertise, to provide a good understanding of the design issues involved in tailoring heuristic algorithms to real-world problems, to compare and judge the efficacy of modern heuristic optimization techniques with other more classic methods of optimization, and to program fundamental evolutionary algorithms and other heuristic optimization routines.

Genetic Algorithms:

Genetic Algorithm (GA) is a search algorithm based on the conjecture of natural selection and genetics. The features of genetic algorithm are different from other search techniques in several aspects. First, the algorithm is a multi-path that searches many peaks in parallel, and hence reducing the possibility of local minimum trapping. Secondly, GA works with a coding of parameters instead of the parameters themselves. The coding of parameter will help the genetic operator to evolve the current state into the next state with minimum computations. Thirdly, GA evaluates the fitness of each string to guide its search instead of the optimization function. The genetic algorithm only needs to evaluate objective function (fitness) to guide its search. There is no requirement for derivatives or other auxiliary knowledge. Hence, there is no need for computation of derivatives or other auxiliary functions. Finally, GA explores the search space where the probability of finding improved performance is high.

Evolution Strategies and Evolutionary Programming:

Evolution Strategies (ES) employ real-coded variables and, in its original form, it relied on Mutation as the search operator, and a Population size of one. Since then it has evolved to share many features with GA. The major similarity between these two types of algorithms is that they both maintain populations of potential solutions and use a selection mechanism for choosing the best individuals from the population. The main differences are: ES operate directly on floating point vectors while classical GAs operate on binary strings; GAs rely mainly on recombination to explore the search space, while ES uses mutation as the dominant operator; and ES is an abstraction of evolution at individual behavior level, stressing the behavioral link between an individual and its offspring, while GAs maintain the genetic link.

Evolutionary Programming (EP) is a stochastic optimization strategy similar to GA, which places emphasis on the behavioral linkage between parents and their offspring, rather than seeking to emulate specific genetic operators as observed in nature. EP is similar to Evolutionary Strategies, although the tow approaches developed independently. Like both ES and GAs, EP is a useful method of optimization when other techniques such as gradient descent or direct analytical discovery are not possible. Combinatorial and real-valued function optimization in which the optimization surface or fitness landscape is “rugged”, possessing many locally optimal solutions, are well suited for Evolutionary Programming.

Particle Swarm Optimization:

Particle swarm optimization (PSO) is an exciting new methodology in evolutionary computation that is somewhat similar to a genetic algorithm in that the system is initialized with a population of random solutions. Unlike other algorithms, however, each potential solution (called a particle) is also assigned a randomized velocity and then flown through the problem hyperspace. Particle swarm optimization has been found to be extremely effective in solving a wide range of engineering problems. It is very simple to implement (the algorithm comprises two lines of computer code) and solves problems very quickly.

Ant Colony Search Algorithm:

The ant colony search algorithms mimic the behavior of real ants. As is well known, real ants are capable of finding shortest path from food sources to the nest without using visual cues. They are also capable of adapting to changes in the environment, for example, finding a new shortest path once the old one is no longer feasible due to a new obstacle. The studies by ethnologists reveal that such capabilities ants have are essentially due to what is called "pheromone trails" which ants use to communicate information among individuals regarding path and to decide where to go. Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction rich in pheromone rather than a poorer one.

Tabu Search:

Tabu Search (TS) is basically a gradient-descent search with memory. The memory preserves a number of previously visited states along with a number of states that might be considered unwanted. This information is stored in a Tabu List. The definition of a state, the area around it and the length of the Tabu list are critical design parameters. In addition to these Tabu parameters, two extra parameters are often used: Aspiration and Diversification. Aspiration is used when all the neighboring states of the current state are also included in the Tabu list. In that case, the Tabu obstacle is overridden by selecting a new state. Diversification adds randomness to this otherwise deterministic search. If the Tabu search is not converging, the search is reset randomly.

Simulated Annealing:

In statistical mechanics, a physical process called annealing is often performed in order to relax the system to a state with minimum free energy. In the annealing process, a solid in a heat bath is heated up by increasing the temperature of the bath until the solid is melted into liquid, then the temperature is lowered slowly. In the liquid phase all particles of the solid arrange themselves randomly. In the ground state the particles are arranged in a highly structured lattice and the energy of the system is minimum. The ground state of the solid is obtained only if the maximum temperature is sufficiently high and the cooling is done sufficiently slowly. Based on the annealing process in the statistical mechanics, the Simulated Annealing (SA) was introduced for solving complicated combinatorial optimization.

The name ‘simulated annealing’ originates from the analogy with the physical process of solids, and the analogy between physical system and simulated annealing is that the cost function and the solution (configuration) in the optimization process correspond to the energy function and the state of statistical physics, respectively. In a large combinatorial optimization problem, an appropriate perturbation mechanism, cost function, solution space, and cooling schedule are required in order to find an optimal solution with simulated annealing. SA is effective in network reconfiguration problems for large-scale distribution systems, and its search capability becomes more significant as the system size increases. Moreover, the cost function with a smoothing strategy enables the simulated annealing to escape more easily from local minima and to reach rapidly to the vicinity of an optimal solution.

List of Contributors

Koay Chin Aik, National University of Singapore;
Eduardo Nobuhiro Asada, Universidade Estadual de Campinas
Hsiao-Dong Chiang, Cornel university;
Alexandre P. Alves da Silva, Federal University of Rio de Janeiro;
ZhaoYang Dong, University of Queensland
Mohamed A. El-Sharkawi, University of Washington
D. M. Falcao, Federal University of Rio de Janeiro
David B. Fogel, Natural Selection, Inc.
Warren L.J. Fox, University of Washington
Yoshikazu Fukuyama, Fuji Electric Co. R&D
Hamid Ghezelayagh, LSI Design & Integration Corp.;
Youngjae Jeon, Korea Electric Power Research Institute,
Ioannis N. Kassabalisdis, University of Washington
Mingoo Kim, University of Washington
Loi Lei Lai, City University of London
Germano Lambert-Torres, Federal Engineering School at Itajuba
Jaewook Lee, Cornel university
Kwang Y. Lee, Pennsylvania State University;
Chen-Ching Liu, Iowa State University
Y. H. Lu, University of Technology;
Vladimiro Miranda, University of Porto
R. J. Marks, Baylor University
Alcir J. Monticelli, Universidade Estaduial de Campinas;
Koichi Nara, Ibaraki University;
N. Ngatchou, University of Washington
Jong-Bae Park, Konkuk University
Ruben Romero, Universidade Estadual Paulista Julio de Mesquita Filho;
N. Sinha, City University of London
Yong Hua Song, University of Liverpool ; Y. H. Lu, University of Technology;
Dipti Srinivasan, National university of Singapore
J. G. Vlachogiannis, Industrial and Energy Informatics Laboratory;
Kit Po Wong, Hong Kong Polytechnique University
Ying Xiao, Brunel University;
Anahita Zarei, University of Washington
I. K. Yu, Changwon National University

Table of Contents

PART 1: THEORY OF MODERN HEURISTIC OPTIMIZATION

- Chapter 1. Introduction to Evolutionary Computation
By David B. Fogel
- Chapter 2. Fundamentals of Genetic Algorithm
By Alexandre P. Alves da Silva and D. M. Falcao
- Chapter 3. Fundamentals of Evolution Strategies and Evolutionary Programming
By Vladimiro Miranda
- Chapter 4. Fundamentals of Particle Swarm Optimization Techniques
By Yoshikazu Fukuyama
- Chapter 5. Fundamentals of Ant Colony Search Algorithms
By Yong Hua Song, Y. H. Lu, Kwang Y. Lee, and I. K. Yu
- Chapter 6. Fundamentals of Tabu Search
By Alcir J. Monticelli, Ruben Romero, and Eduardo Nobuhiro Asada
- Chapter 7. Fundamentals of Simulated Annealing
By Alcir J. Monticelli, Ruben Romero, and Eduardo Nobuhiro Asada
- Chapter 8 Fuzzy Systems
By Germano Lambert-Torres
- Chapter 9. Differential Evolution, an Alternative Approach to Evolutionary Algorithm
By Kit Po Wong and ZhaoYang Dong
- Chapter 10 Pareto Multi-Objective Optimization
By Patrick N. Ngatchou, Anahita Zarei, Warren L.J. Fox, and M. A. El-Sharkawi

PART 2: SELECTED APPLICATIONS OF MODERN HEURISTIC OPTIMIZATION IN POWER SYSTEMS

- Chapter 11. Overview of Applications in Power Systems
By Alexandre P. Alves da Silva, D. M. Falcao, and Kwang Y. Lee
- Chapter 12 Trust-Tech Paradigm for Computing High-quality Optimal Solutions: Method and Theory
By Hsiao-Dong Chiang and Jaewook Lee

Chapter 13 Application of Evolutionary Technique to Power System Vulnerability Assessment

By Mingoo Kim, M. A. El-Sharkawi, R. J. Marks, and Ioannis N. Kassabalisdis

Chapter 14. Applications to System Planning

By Eduardo Nobuhiro Asada, Kwang Y. Lee, Vladimiro Miranda, Alcir J. Monticelli, Koichi Nara, Jong-Bae Park, Ruben Romero, and Yong Hua Song

Chapter 15. Applications to Power System Scheduling

By Koay Chin Aik, Loi Lei Lai, Kwang Y. Lee, Y. H. Lu, Jong-Bae Park, Yong Hua Song, Dipti Srinivasan, J. G. Vlachogiannis, and I. K. Yu

Chapter 16. Power System Controls

By Yoshikazu Fukuyama, Hamid Ghezelayagh, Kwang Y. Lee, Chen-Ching Liu, Ying Xiao, and Yong Hua Song

Chapter 17. Genetic Algorithms for Solving Optimal Power Flow Problems

By Loi Lei Lai and N. Sinha

Chapter 18. An Interactive Compromise Programming Based Multi-objective Approach to FACTS Control

By Ying Xiao, Yong Hua Song, and Chen-Ching Liu

Chapter 19. Hybrid Systems

By Vladimiro Miranda