

Preface

Evolutionary computation (EC) is one of the most important emerging technologies of recent times. Over the last years, there has been exponential growth of research activity in this field. Despite the fact that the concept itself has not been precisely defined, EC has become the standard term that encompasses several stochastic, population-based, and system-inspired approaches.

EC methods use as inspiration our scientific understanding of biological, natural, or social systems, which at some level of abstraction can be represented as optimization processes. They intend to serve as general-purpose easy-to-use optimization techniques capable of reaching globally optimal or at least nearly optimal solutions. In their operation, searcher agents emulate a group of biological or social entities which interact to each other based on specialized operators that model a determined biological or social behavior. These operators are applied to a population (or several subpopulations) of candidate solutions (individuals) that are evaluated with respect to their fitness. Thus, in the evolutionary process, individual positions are successively approximated to the optimal solution of the system to be solved.

Due to their robustness, EC techniques are well-suited options for industrial and real-world tasks. They do not need gradient information, and they can operate on each kind of parameter space (continuous, discrete, combinatorial, or even mixed variants). Essentially, the credibility of evolutionary algorithms relies on their ability to solve difficult, real-world problems with the minimal amount of human effort.

There exist some common features clearly appear in most of the EC approaches, such as the use of diversification to force the exploration of regions of the search space, rarely visited until now, and the use of intensification or exploitation, to investigate thoroughly some promising regions. Another common feature is the use of memory to archive the best solutions encountered.

Numerous books have been published tacking into account any of the most widely known methods, namely simulated annealing, tabu search, evolutionary

algorithms, ant colony algorithms, particle swarm optimization, or differential evolution, but attempts to consider the discussion of alternative approaches are scarce.

The excessive publication of developments based on the simple modification of popular EC methods presents an important disadvantage, in that it distracts attention away from other innovative ideas in the field of EC. There exist several alternative EC methods which consider very interesting concepts; however, they seem to have been completely overlooked in favor of the idea of modifying, hybridizing, or restructuring traditional EC approaches.

The goal of this book is to present advances that discuss alternative EC developments or highlight non-conventional operators which prove to be effective in adapting a determined EC method to a specific problem. This book has been structured so that each chapter can be read independently from the others. Chapter 1 describes evolutionary computation (EC). This chapter concentrates on elementary concepts of evolutionary algorithms. Readers that are familiar with EC may wish to skip this chapter.

In Chap. 2, a swarm algorithm, namely the Social Spider Optimization (SSO), is presented for solving optimization tasks. The SSO algorithm is based on the simulation of the cooperative behavior of social spiders. In SSO, individuals emulate a group of spiders which interact to each other based on the biological laws of the cooperative colony. Different to the most EC algorithms, SSO considers two different search agents (spiders): males and females. Depending on the gender, each individual is conducted by a set of different evolutionary operators which mimic the different cooperative behaviors assumed in the colony. To illustrate the proficiency and robustness of the SSO, it is compared to other well-known evolutionary methods.

Chapter 3 presents a nature-inspired algorithm called the States of Matter Search (SMS). The SMS algorithm is based on the modeling of the states of matter phenomenon. In SMS, individuals emulate molecules which interact to each other by using evolutionary operations based on the physical principles of the thermal energy motion mechanism. The algorithm is devised considering each state of matter one different exploration–exploitation ratio. In SMS, the evolutionary process is divided into three phases which emulate the three states of matter: gas, liquid, and solid. In each state, the evolving elements exhibit different movement capacities. Beginning from the gas state (pure exploration), the algorithm modifies the intensities of exploration and exploitation until the solid state (pure exploitation) is reached. As a result, the approach can substantially improve the balance between exploration–exploitation, yet preserving the good search capabilities of an EC method. To illustrate the proficiency and robustness of the proposed algorithm, it was compared with other well-known evolutionary methods including recent variants that incorporate diversity preservation schemas.

In Chap. 4, an EC algorithm inspired by the collective animal behavior (CAB) is presented. In this algorithm, the searcher agents represent a group of animals that interact to each other based on simple behavioral rules which are modeled as mathematical operators. Such operations are applied to each agent considering that

the complete group has a memory storing their own best positions seen so far, by using a simple competition principle. The approach has been compared to other well-known optimization methods. The results confirm a high performance of the proposed method for solving various benchmark functions.

In Chap. 5, a novel biologically inspired algorithm, namely Allostatic Optimization (AO), is presented. The AO algorithm uses as metaphor the allostasis mechanisms for designing an optimization methodology. AO provides a population-based search procedure under which all individuals, also seen as body conditions, are defined within the multidimensional space. They are generated or modified by using several evolutionary operators that emulate the different operations employed by the allostasis process, whereas the fitness function evaluates the capacity of each individual (body condition) to reach a steady health state (good solution). Different to several popular EC methods, AO implements evolutionary operators that avoid concentrating most of the particles in only one position favoring the exploration process and eliminating the flaws related to the premature convergence. The approach has been compared to other well-known evolutionary algorithms. The results confirm a high performance of the proposed method for solving various benchmark functions.

In Chap. 6, a swarm algorithm, called the Locust Search (LS), is presented for solving some optimization tasks. The LS algorithm is based on the behavioral modeling of swarms of locusts. In LS, individuals represent a group of locusts which interact to each other based on the biological laws of the cooperative swarm. The algorithm considers two different behaviors: solitary and social. Depending on the behavior, each individual is conducted by a set of evolutionary operators which mimics different cooperative conducts that are typically found in the swarm. Different to most of existent swarm algorithms, the behavioral model in the proposed approach explicitly avoids the concentration of individuals in the current best positions. Such fact allows not only to emulate in a better realistic way the cooperative behavior of the locust colony, but also to incorporate a computational mechanism to avoid critical flaws that are commonly present in the popular particle swarm optimization and differential evolution, such as the premature convergence and the incorrect exploration–exploitation balance. In order to illustrate the proficiency and robustness of the proposed approach, its performance is compared to other well-known evolutionary methods. The comparison examines several standard benchmark functions which are commonly considered in the literature.

Chapter 7 presents an EC method called the Adaptive Population with Reduced Evaluations (APREs) for solving optimization problems which are characterized by demanding an excessive number of function evaluations. APRE reduces the number of function evaluations through the use of two mechanisms: (1) adapting dynamically the size of the population and (2) incorporating a fitness estimation strategy that decides the amount of individuals to be evaluated with the original fitness function and the amount of individuals to be estimated by a very simple approximated model. APRE begins with an initial random population which will be used as a memory during the evolution process. After initialization, it is selected the elements to be evolved. Its number is automatically modified in each iteration. With

the selected elements, a set of new individuals is generated as a consequence of the execution of the seeking operation. Afterward, the memory is updated. For this process, the new individuals produced by the seeking operation compete against the memory elements to build the final memory configuration. Finally, a sample of the best elements contained in the final memory configuration is undergone to the refinement operation. This cycle is repeated until the maximum number the iterations have been reached. Different to other approaches that use an already existent EA as framework, the APRE method has been completely designed to substantially reduce the computational cost without degrading its good search capacities.

Different to global optimization, the main objective of multimodal optimization is to find multiple global and local optima for a problem in one single run. Finding multiple solutions to a multimodal optimization problem is especially useful in engineering, since the best solution may not always be the best realizable due to various practical constraints. In Chap. 8, the multimodal characteristics of the CAB algorithm are exposed in Chap. 4. The main objective is to analyze the particular CAB operators that permit its multimodal performance.

Finally, Chap. 9 presents a variant of the SSO algorithm (exposed in Chap. 2) for solving constrained optimization problems. The method, called SSO-C, implements additional mechanisms that the original one. For constraint handling, SSO-C incorporates the combination of two different paradigms in order to direct the search toward feasible regions of the search space. In particular, it has been added: (1) a penalty function which introduces a tendency term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained one and (2) a feasibility criterion to bias the generation of new individuals toward feasible regions increasing also their probability of getting better solutions.

As authors, we wish to thank many people who were somehow involved in the writing process of this book. We thank Dr. Raul Rojas and Dr. Gonzalo Pajares for supporting us to have it published. We express our gratitude to Prof. Janusz kacprzyk, who so warmly sustained this project. Acknowledgements also go to Dr. Thomas Ditzinger, who so kindly agreed to its appearance. We also acknowledge that this work was supported by CONACYT under the grant number CB 181053.

Guadalajara
Berlin
Zapopan
December 2015

Erik Cuevas
Margarita Arimatea Díaz Cortés
Diego Alberto Oliva Navarro

Advances of Evolutionary Computation: Methods and Operators

Cuevas, E.; Díaz Cortés, M.A.; Oliva Navarro, D.A.

2016, XIV, 202 p. 48 illus., 5 illus. in color., Hardcover

ISBN: 978-3-319-28502-3