Efficient Optimization Using Metaheuristics

T

Sergio Nesmachnow
Universidad de la República, Uruguay

INTRODUCTION

Nowadays, many real-world problems are intrinsically complex. Optimization is a process that deals with hard problems arising in many scientific, academic, and commercial applications. Many optimization problems are difficult to solve, because of having large solution spaces, handling complex functions and hard constraints, or managing large volumes of data. Traditional computing methods are not useful in practice for solving these hard problems, because they demand very large execution times when solving realistic instances.

Metaheuristics are soft computing methods that allow computing accurate solutions for hard-to-solve problems in reasonable execution times. Metaheuristics are conceived under simple paradigms to take advantage of tolerating imprecision, uncertainty, and approximation in the problem data. They can handle partial information and non-exact solutions in order to compute accurate solutions for optimization problems. These features make metaheuristics highly valuable techniques, as they allow meeting realistic resolution times in many application areas, from informatics to industrial and commercial.

Metaheuristics were proposed almost thirty years ago, and nowadays this research area is consolidated. Many metaheuristic proposals have been formulated, regarding both theoretical models and applications. This chapter provides an insight into the main concepts, advances, and results in the field of metaheuristics for efficiently solving optimization problems. A general view of most well-known metaheuristics is presented and the main applications of metaheuristics in nowadays real-world problems from several domains is described. Finally, the main current and future research lines in the field are also summarized.

DOI: 10.4018/978-1-5225-2255-3.ch669

BACKGROUND

Metaheuristics are high-level soft computing strategies that define algorithmic frameworks and techniques to find approximate solutions for optimization problems (Blum & Roli, 2003; Talbi, 2009). Many efficient and accurate metaheuristics have been proposed, which can be applied to solve a variety of optimization problems underlying many applications in science/technology, industry, and commerce.

Metaheuristics Concepts: Definition and Classification

Metaheuristics were originally proposed as high-level problem-independent strategies to coordinate several heuristic search methods, which can be instantiated to solve hard problems. This definition has broadened to include a wide range of search and learning processes (including shaking, construction/deconstruction, adaptation, swarming and collective behavior, hybridization, etc.) applied to improve the search.

The heuristic components in a metaheuristic are conceived to be applied in an intelligent way, providing accurate and balanced methods for diversification and intensification. These two concepts are very important to guarantee the efficacy of the search: *diversification* refers to achieving a good exploration pattern for the search space, providing a reasonable coverage and avoiding stagnation in local optima; *intensification* means exploiting or improving already found accurate solutions to increase their quality.

Regarding the number of solutions handled, metaheuristics are classified in two classes: *trajectory-based* and *population-based*. Trajectory-based metaheuristics work with a single solution, which is replaced by another (often the best) solution found in its neighborhood. The search is characterized by a trajectory in the space of solutions. Trajectory metaheuristics allow quickly exploiting solutions, thus they are referred as intensification-oriented methods. On the other hand, population-based metaheuristics work with a set of candidate solutions, which are modified and/or combined following some common guidelines. Some solutions in the population are replaced by newly generated solutions (often the best). These methods are characterized as diversification-oriented, because having multiple solutions allows significantly increasing the exploration capabilities.

Other classification criteria for metaheuristics are related to specific features of the search strategy, for example: memory vs. memory-less, or dynamic vs. static objective function.

Metaheuristic Techniques

Trajectory-based metaheuristics include many well-known techniques from the early years of this area:

- **Simulated Annealing (SA):** Probabilistically allows moving to a solution with worst objective function value, trying to escape from local optima (Kirkpatrick et al., 1983). SA is inspired on the annealing process of metals and it was the first metaheuristic proposed, although the term "metaheuristic" was not used in those years.
- **Tabu Search (TS):** Enhances a local search strategy by using memory to store information about visited solutions. To promote diversification, returning to recently visited solutions is not allowed. TS was introduced by Glover (1986), who first used the term "metaheuristic".
- **Greedy Randomized Adaptive Search Procedure (Feo and Resende, 1995):** A metaheuristic which greedily selects components to construct a solution, and then applies a local search to improve it.
- **Variable Neighborhood Search (Mladenovic & Hansen, 1997):** Local search techniques based on using different neighborhood structures during the search.
- **Iterated Local Search (Lourenço et al., 2002):** Uses a hill-climbing to find local optima and a stochastic perturbation/restart strategy to prevent the search getting stuck in local optima.

Population-based metaheuristics include techniques that use cooperation in the search:

- **Evolutionary Computation (EC):** Emulates the evolution of species in nature by applying stochastic operators (recombination and random changes) on a population. A selection-of-the-best strategy guides the search to high-quality solutions. EC includes Genetic Programming (Koza, 1992), Evolution Strategies, and Evolutionary Algorithms (EAs) (Goldberg, 1989).
- **Swarm Intelligence (SI):** Uses agents to perform explorations while they interact with neighbors and the environment, based on the collective behavior of self-organized systems. Agents have limited search capabilities, but an "intelligent" pattern emerges in large swarms. SI include Ant Colony Optimization (ACO), emulating the behavior of ants foraging for food (Dorigo, 1992), Particle Swarm Optimization (PSO), simulating flocks of birds (Kennedy & Eberhart, 1995), Artificial Immune System (DasGupta, 1998), and the recently proposed Bacterial Foraging, Fish/Glowworm Swarm, Firefly Algorithm, Cuckoo Search, and Bee Algorithms.
- **Evolutionary-Inspired Metaheuristics:** Apply an evolutionary search, with different features. Estimation of Distribution Algorithms learn by building/sampling probabilistic models of promising solu-

Related Content