GLOBAL OPTIMIZATION AND META-HEURISTICS

Manuel Laguna
University of Colorado at Boulder, USA

Keywords: optimization, tabu search, scatter search, genetic algorithms, meta-heuristics

Contents

Summary

This article describes the origin and significant developments associated with the field of meta-heuristics as they relate to global optimization. Meta-heuristics provide a means for approximately solving complex optimization problems. These methods are designed to search for global optima; however, they cannot guarantee that the best solution found after termination criteria are satisfied is indeed a global optimal solution to the problem. Experimental testing of meta-heuristic implementations show that the search strategies embedded in such procedures are capable of finding solutions of high quality to hard problems in industry, business, and science.

1. Introduction

The theory of optimization refers to the quantitative study of optima and the methods for finding them. The technical verb “optimize” means to achieve the optimum, and optimization is the act of optimizing. To achieve the optimum entails in some cases obtaining the most of some measure of success (e.g., revenue), or in some other cases obtaining the least of another measure (e.g., cost). Choosing a quantitative measure of effectiveness and then optimizing it has become the typical way in which many important decisions are made. Decisions involving how to design, build, or operate a physical or economics system are reached in three steps (see *Fundamentals of Operations Research*).

1. Identify the decision variables in the system and determine, accurately and qualitatively, how they interact.
2. Identify a measure of system effectiveness that can be expressed in terms of the system variables. This measure is often referred to as the objective function.
3. Choose those values of the system variables that yield optimum effectiveness.

In classical optimization methods, such as linear programming, these three steps result in a model formulation of the type:

Maximize or minimize $f(x)$
Subject to $g(x) \leq b$

In this formulation, $f(x)$ is the quantitative measure of effectiveness (or objective function) and x are the decision variables. The set of constraints is also formulated in terms of the decision variables and represented as bounds on the function $g(x)$. In the case of linear programming both $f(x)$ and $g(x)$ are linear functions. Linear programming is considered a general-purpose tool because the only requirement is to represent the optimization model as a linear objective function subject to a set of linear constraints. The state-of-the-art linear programming solvers are quite powerful and can successfully solve models with thousands and even millions of variables employing reasonable amounts of computer effort (see *Linear Programming*). Evidently, however, not all business, industrial, and scientific problems can be expressed by means of a linear objective and linear equalities or inequalities. Many complex systems may not even have a convenient mathematical representation, linear or nonlinear. Techniques such as linear programming and its cousins (nonlinear programming and integer programming) generally require a number of simplifying assumptions about the real system to be able to properly frame the problem (see *Nonlinear Programming, Combinatorial Optimization, and Integer Programming*).

Linear programming solvers are design to exploit the structure of a well-defined and carefully studied problem. The disadvantage to the user is that in order to formulate the problem as linear program, simplifying assumptions and abstractions may be necessary. This leads to the well-known dilemma of choosing between finding the optimal solution to a model that does not represent the real system accurately and developing a model that is a good abstraction of the real system, but for which only inferior suboptimal solutions can be obtained. When dealing with the optimization of complex systems, a course of action taking for many years has been to develop specialized heuristic procedures that, in general, do not require a mathematical formulation of the problem. These procedures were appealing from the standpoint of simplicity, but generally lacked the power to provide high-quality solutions to complex problems.

For example, consider the well-known traveling salesman problem. This is the problem of finding the shortest route that visits each of a given collection of locations precisely once and eventually returns to the starting point. One simple heuristic for this problem may be to arbitrarily select a starting location and then always choose from a candidate list the next location that is closest to the current location. Once a location is chosen, it is deleted from the candidate list of unvisited locations. While this heuristic may occasionally give acceptable results in problems that consist of only a handful of

locations, in general, its performance is predicted to be extremely poor. This procedure falls within the class of heuristics called myopic, because they make decisions based on limited—also called "local"—information without considering the consequences of implementing those decisions. In the case of the traveling salesman problem, choosing the nearest city at every point of the way may build a route that systematically moves the salesman farther from the starting point, making the return trip unnecessarily long.

In addition to heuristics designed to construct solutions, there are also procedures for improving solutions. The most common way of improving a solution is via the application of a local search. To continue with our traveling salesman illustration, suppose that a delivery truck driver is asked to visit five locations in the following order:

Warehouse $\rightarrow$ A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E $\rightarrow$ Warehouse

Also suppose that the route was constructed using the "nearest location" rule discussed above. A local search procedure would attempt to modify the current route by performing a move (or change). One possible move is to exchange the position of two locations and measure the impact on the objective function. We assume that the objective is to minimize the total length of the route, so typically a move that decreases the length of the route is considered "better," in the local sense, than one that increases it. If we limit our local search to moves that exchange locations that immediately follow each other in the current route, then we only have to test four moves: (A,B), (B,C), (C,D), and (D,E). We would pick the "best" one of those moves. However, if we would like to test all possible exchanges of two locations as part of the local search effort, the number of moves to be examined is ten. The amount of exploration, which is directly related to the amount of computational effort, is an important design issue in local search procedures. The effort to explore the neighborhood of a solution (that is the set of solutions reachable from the current solution by applying a move mechanism) can vary considerably. In a traveling salesman problem with n locations, there are $n - 1$ neighbors if the move is defined as exchanging the positions of two locations that immediately follow each other in the current solution. However, if the move is defined as the exchange of positions of any two locations, the size of the neighborhood (i.e., the cardinality of the set of solutions reachable with such a move) is $(n^2 - n)/2$. Regardless of the move mechanism, local search typically explores only a small fraction of the solution space. In the case of the traveling salesman problem, for example, the solution space consists of $n!$ solutions. So, local search procedures that explore in the order of n^2 or even n^3 solutions are only dealing with a fairly small fraction of the entire solution space as the dimension of the problem increases.

Heuristics designed for constructing solutions are typically combined with local search procedures to create what is called a "hill climbing" method. These methods start from a solution and apply a local search in an attempt to find an improved solution. If an improved solution is found, the search moves to it and the local search is applied again. The method stops when the local search is not capable of finding a solution that improves upon the current solution (i.e., when the "best" possible move cannot improve upon the objective function value of the current solution). The hill climbing terminology refers to the trajectory of the objective function values in a maximization problem. The steps in a hill-climbing procedure can be summarized as follows:

1. Generate an initial solution *s*.
2. Apply a local search to find the best neighbor solution *s*′.
3. If s′ is better than *s* then make the current solution *s* equal to *s*′ and go to step 2, else terminate.

The main shortcoming of a hill climbing method is its inability to escape local optimality. Figure 1 shows a nonlinear function with a single variable for which a hill climbing method would be able to find the local maximum only if the initial solution happens to fall within the range (*b, c*). However, since the range (*a, b*) is considerably wider, a hill climbing method that starts from a randomly generated initial point would likely be "trapped" in the inferior local optimal point *x*′ instead of finding the global maximum point x^*.

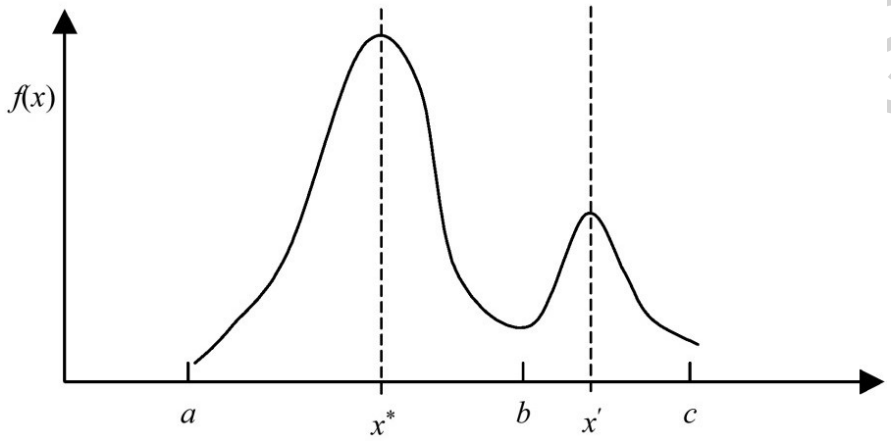

Figure 1. Bimodal function

Meta-heuristics provided a way of considerably improving the performance of simple heuristic procedures, such as those based on hill climbing. The search strategies proposed by meta-heuristic methodologies result in iterative procedures with the ability to escape local optimal points. Meta-heuristics have been developed to solve complex optimization problems in many areas, with combinatorial optimization being one of the most fruitful. Generally, the best procedures achieve their efficiencies by relying on context information. The solution method can be viewed as the result of adapting meta-heuristic strategies to specific optimization problems.

The term "meta-heuristic" (also written "metaheuristic") was coined by Fred Glover in 1986 and has come to be widely applied in the literature, both in the titles of comparative studies and in the titles of volumes of collected research papers. A meta-heuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality. The heuristics guided by such a meta-strategy may be high level procedures or may embody nothing more than a description of available moves for transforming one solution into another, together with an associated evaluation rule.

The contrast between the meta-heuristic orientation and the "local optimality" orientation is significant. For many years, the primary conception of a heuristic

procedure—a conception still prevalent today—was to envision either a clever rule of thumb or an iterative rule that terminates as soon as no solutions immediately accessible could improve the last one found. Such iterative heuristics are often referred to as descent methods, ascent methods, or local search methods. (A sign of the times is that "local search" now sometimes refers to search that is not limited to being local in character.) Consequently, the emergence of methods that departed from this classical design—and that did so by means of an organized master design—constituted an important advance. Widespread awareness of this advance only began to dawn during the last decade, though its seeds go back much farther.

The evolution of meta-heuristics during the past ten years has taken an explosive upturn. Meta-heuristics in their modern forms are based on a variety of interpretations of what constitutes "intelligent" search. These interpretations lead to design choices that in turn can be used for classification purposes. However, a rigorous classification of different meta-heuristics is a difficult and risky enterprise, because the leading advocates of alternative methods often differ among themselves about the essential nature of the methods they espouse. This may be illustrated by considering the classification of meta-heuristics in terms of their features with respect to three basic design choices:

1. The use of adaptive memory,
2. The kind of neighborhood exploration used, and
3. The number of current solutions carried from one iteration to the next.

These options can be embedded in a classification scheme of the form *x*/*y*/*z*, where the choices for *x* are A (if the meta-heuristic employs adaptive memory) and M (if the method is "memoryless"). The choices for *y* are N (for a method that employs some systematic neighborhood search either to select the next move or to improve a given solution) and S (for those methods relying on random sampling). Finally, *z* may be 1 (if the method moves from one current solution to the next after every iteration) or P (for a population-based approach with a population of size P). This simple three-dimensional scheme gives us a preliminary basis of classification, which discloses that agreement on the proper way to label various meta-heuristics is far from uniform. We show this by providing classifications for a few well-known meta-heuristics in Table 1.

Meta-heuristic	Classification 1	Classification 2
Genetic algorithms	M/S/P	M/N/P
Scatter search	M/N/P	A/N/P
Simulated annealing	M/S/1	M/N/1
Tabu search	A/N/1	A/N/P

Table 1. Meta-heuristic classification

Two different ways are given for classifying each of these procedures. The first classification most closely matches the "popular conception" and the second is favored by a significant (if minority) group of researchers. The differences in these classifications occur for different reasons, depending on the method. Some differences have been present from the time the methods were first proposed, while others represent recent changes that are being introduced by a subgroup of ardent proponents. For

example, the original form of simulated annealing has come to be modified by a group that believes stronger elements of neighborhood search should be incorporated. A similar change came about in genetic algorithms—a few years before it was introduced in simulated annealing—in the mid-1980s. Still, it should be pointed out that not all the advocates of simulated annealing and genetic algorithms view these changes as appropriate.

On the other hand, among those examples where different classifications were present from the start, the foundation papers for tabu search (TS) included population-based elements in the form of strategies for exploiting collections of elite solutions saved during the search. Yet a notable part of the literature has not embraced such population-based features of TS until recently. Similarly, scatter search was accompanied by adaptive memory elements as a result of being associated with early TS ideas, but this connection is likewise only beginning to be pursued.

A few proponents of simulated annealing and genetic algorithms have recently gone farther in modifying the original conceptions than is indicated in Table 1—proposing the inclusion of elements of adaptive memory as embodied in TS. Such proposals are often described by their originators as hybrid methods, due to their marriage of aspects from different frameworks.

-
-
-

Bibliography

Corne D., Dorigo M. and Glover F. (1999). *New Ideas in Optimization*. Maidenhead, UK: McGraw-Hill. [This book addresses novel meta-heuristic methods as well as the emerging ideas for future designs.]

Glover F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences* **8**, 156–166. [Seminal work on tabu search and scatter search.]

Glover F. and Laguna M.(1997) *Tabu Search.* Boston, MA: Kluwer Academic Publishers. [This book presents a comprehensive treatment of the meta-heuristic known as tabu search.]

Goldberg D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA: Addison-Wesley. [One of the first comprehensive books on genetic algorithms.]

Holland J.H. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press. [Seminal work on genetic algorithms.]

Kelly J. and Osman I. (1996). *Meta-heuristics: Theory and Applications.* Boston, MA.: Kluwer Academic Publishers. [It contains articles presented in the first conference on meta-heuristics held in Breckenridge, CO.]

Kirkpatrick S., Gellat C.D., and Vecchi M.P. (1983). Optimization by Simulated Annealing. *Science* **220**, 671–680. [Seminal work on simulated annealing.]

Michalewicz Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs, Third Edition*. Berlin: Springer-Verlag. [The associated computer code called GENOCOP has been extensively used as a benchmark for multimodal function optimization.]

Reeves C. (1993). *Modern Heuristic Techniques for Combinatorial Problems.* Oxford: Blackwell Scientific Publications. [This addresses in detail well-known meta-heuristics and also includes material on experimental testing.]

Biographical Sketch

Manuel Laguna is Associate Professor of Operations Management in the College of Business and Administration and Graduate School of Business Administration of the University of Colorado at Boulder. He received Masters and Doctoral degrees in Operations Research and Industrial Engineering from the University of Texas at Austin. He has done extensive research in the interface between computer science, artificial intelligence, and operations research to develop solution methods for problems in areas such as production planning and scheduling, routing and network design in telecommunications, combinatorial optimization on graphs, and optimization of simulations. Dr. Laguna has more than 40 publications, including articles in scientific journals such as *Operations Research*, *Management Science*, *European Journal of Operational Research*, *Computers and Operations Research*, *IIE Transactions*, and the *International Journal of Production Research.* He is the co-author of *Tabu Search*—the first book devoted to this innovative optimization technology. He is principal editor of the *Journal of Heuristics*, on the editorial board of *Combinatorial Optimization: Theory and Practice*, and has been guest editor for the *Annals of Operations Research* and the *European Journal of Operational Research*. Dr. Laguna is a member of the Institute for Operations Research and Management Science, the Institute of Industrial Engineering, and the International Honor Society Omega Rho.