Metaheuristics: Heuristic Techniques for Combinatorial Optimization Problems

Stephan Scheuerer
Fraunhofer Center for Applied Research on Technologies for the Logistics Service Industries ATL, Germany

INTRODUCTION

Decision support systems (DSSs) provide modern solution techniques that help the decision maker to find the best solution to a problem. These embedded solution techniques include and combine, but are not limited to, simulation, exact optimization methods, and heuristics. Especially in the field of heuristics, recent advances in metaheuristic methods have proved to be remarkably effective so that metaheuristics are nowadays the preferred way for solving many types of complex problems, particularly those of combinatorial nature. Some of these problems are, for example, the well-known "traveling salesman" problem, the generalized assignment problem, the set-covering problem, and vehicle and network routing applications. Most of all, metaheuristics allow us to solve real-world problems with a notably high level of complexity. This is where exact methods are often incapable of finding solutions whose qualities are close to that obtained by the leading metaheuristics. Metaheuristic applications with world-class performance can be found in all kinds of areas such as economics, engineering, and natural sciences.

This article will describe the basic notions of several metaheuristic techniques to show the user of a DSS how the principles of the embedded solution techniques could apply. For the sake of brevity, the following discussion will concentrate on a representative sample of the different heuristic paradigms, namely, tabu search (TS), simulated annealing (SA), greedy randomized adaptive search procedure (GRASP), ant colony optimization (ACO), evolutionary algorithms (EA), and scatter search (SS). For a more detailed insight into this emerging research area, the following small subset out of the overwhelming number of literature on this topic could provide a starting point for further reading: Gendreau and Potvin (2005), Glover and Kochenberger (2003), and Blum and Roli (2003).

The classical vehicle routing problem (VRP) will be used for illustration in the following sections. The VRP is a hard combinatorial problem with a high relevance for the logistics practice. Metaheuristics have proven to successfully tackle this complex problem. In the VRP, a set of customers has to be served by a number of vehicles located at a depot. Each customer has a nonnegative demand and a nonnegative service time. The VRP consists of designing a set of cost-minimal vehicle routes, each starting and ending at the depot. Each customer must be visited exactly once by a vehicle, with the total demand of any route not exceeding the vehicle capacity, and the total duration of any route not surpassing a preset limit. For a comprehensive overview on the VRP, refer to Toth and Vigo (2002).

BACKGROUND

The application of exact optimization methods to real-life problems assumes that it is possible to find an optimal solution in a reasonable computing time. This, however, is not the case for most real-life combinatorial optimization problems[1] like, for example, rich vehicle routing or scheduling problems. To overcome this shortcoming, heuristic methods were developed to help the decision maker find a satisfying solution, in the best case the optimum, in a realistic amount of computing time.

In the early years, specialized heuristics were developed for a range of specific optimization problems. In this context, the knowledge about the optimization problem itself plays a significant role in the design of the heuristic, especially for the solution quality, as most heuristic procedures in this category have a constructive nature. These constructive methods build up a complete solution from scratch by sequentially adding components to a solution fragment until the solution is complete. This sequential process often leads to good, rarely to optimal, solutions. To improve the solution, so-called improvement methods can be applied. A technique that is used very often in the class of improvement methods is local search algorithms that try to iteratively modify a given solution in order to generate improvements. Modifications are done on a local basis, for example, by shifting or exchanging a single component in a complete solution. These kinds of local operations (often called moves) could lead to a new best solution.[2] All solutions that can be reached by a given kind of modification are considered to build up the set of neighboring solutions, called the neighborhood, to the given solution. A descent local search algorithm performs these modifications as long as there is an improvement in the objective function value. The search stops whenever there is no further improvement. The solution is then considered to be a local optimum with respect to the neighborhood considered. An example for a constructive method in the context of vehicle routing is the well-known savings algorithm of Clarke and Wright (1964). Starting from single round trips from the depot to each customer, the routes are merged according to a savings criterion that expresses the distance saved through this step as long as feasibility of the solution is maintained. To improve the solution even more (sometimes still during the construction phase), for example, k-opt improvement routines[3] can be applied. This method iteratively modifies up to k edges of a single vehicle tour as long as there is an improvement in tour cost.

Using local search methods often helps to make significant improvements to an initially constructed solution. However, they have the disadvantage of possibly being trapped in a local optimum that is not the global optimum. To solve this problem, metaheuristic techniques can be applied. The way of achieving this aim depends on the philosophy of the metaheuristic. The term metaheuristic was originally coined by Glover (1986) and describes a more general, not problem-specific solution scheme. Metaheuristics can be considered[4] strategies that guide a search process to efficiently explore the search space for finding a (close to) optimal solution.

One possibility[5] for classifying metaheuristics is dividing them into the following two categories: single-solution metaheuristics (representatives are, e.g., TS, SA, GRASP), which basically work on a single solution at a time, and population-based metaheuristics (e.g., EA, SS, ACO), which consider a set of solutions rather than a single solution. Single-solution metaheuristics may also be called trajectory methods because the search is characterized by a trajectory in the search space (Blum & Roli, 2003).

In metaheuristic search, the goal is an efficient and effective exploration of the search space. In order to achieve this aim, the right balance between intensification and diversification is crucial. Intensification refers to a more thorough examination of attractive-looking regions whilst diversification encourages the search for unvisited regions of the search space and should thus lead to new and even more attractive regions. A well-balanced use of these two elements can be seen as the key to the design of efficient metaheuristics.

OVERVIEW OF METAHEURISTIC PRINCIPLES

In the following, some metaheuristics will be briefly described. The reader is referred to the literature at the end of each section for further readings.

Tabu Search (TS)

TS was first introduced by Glover (1986) and Hansen (1986). A simple TS can be seen as a local search strategy where always the best solution in the neighborhood of the current solution is selected as the new solution, even if it deteriorates solution cost. This strategy helps to overcome local optima. To prevent the search from cycling, recently visited solutions are considered forbidden or tabu for a certain number of iterations. A short-term memory, called a tabu list, is used to store these solutions and characteristic attributes of these solutions. Aspiration criteria define rules that allow us to overcome the tabu status under certain conditions. The algorithm stops when a termination condition, for example, a maximum number of iterations, is met.

For the VRP, a move can be a shift of a single customer into another vehicle's tour. The reverse move back into its original tour would be forbidden for a number of iterations (tabu tenure). The reverse move would, however, be allowed if it led to a new best solution (aspiration criteria). A whole number of successful TS implementations for the VRP can be reported.[6]

One crucial issue in the design of tabu search algorithms is the tabu tenure. Too small tabu tenures cannot effectively prevent the search from cycling. However,