Metaheuristics for operations and supply chain management: Fundamentals and applications

Celso C Ribeiro[a] **and Rafael A Melo**[b], [a]Institute of Computing, Universidade Federal Fluminense, Niterói, RJ, Brazil; [b]Institute of Computing, Universidade Federal da Bahia, Salvador, BA, Brazil

© 2025 Elsevier Inc. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Abstract

This chapter overviews approximate optimization methods, from greedy and semigreedy construction algorithms to local search and from local search to metaheuristics. Metaheuristics are general high-level procedures coordinating simple heuristics, methods, and rules to find high-quality approximate solutions to hard optimization problems. We review the basic principles and ideas in designing the most widely and successfully used metaheuristics. We conclude with a discussion about the application of metaheuristics in operations and supply chain management optimization.

Glossary

Constructive heuristic Heuristic that builds a feasible solution step-by-step from scratch.
Exact algorithm Algorithm that is always able to find the optimal solution for some problem in finite time.
Genetic algorithm Population-based method that uses selection, recombination, and mutation operators to generate new solutions, imitating the process of natural selection and evolution of species.
GRASP (Greedy randomized adaptive search procedure) Metaheuristic resulting from the hybridization of a greedy randomized algorithm with a local search embedded in a multistart framework.
Heuristic or approximate algorithm Algorithm to compute a feasible solution for some problem in finite time.
Local search heuristic Heuristic that iteratively improves a feasible solution by minor modifications until a solution that cannot be further improved is found.
Metaheuristic High-level procedure that coordinate, simple methods and rules to find high-quality solutions to optimization problems.
NP-hard Class of computationally challenging problems for which it is unknown whether they can be exactly solved in polynomial time.
Optimal solution Best feasible solution for some optimization problem.
Simheuristics Decision support systems that combine simulation and metaheuristics.
Simulated annealing Memoryless metaheuristic that mimics the physical annealing process.
Tabu search Dynamic neighborhood method that uses memory to drive the search by escaping from local optima previously visited.
VNS (Variable neighborhood search) Metaheuristic based on a systematic change of neighborhoods.

 https://doi.org/10.1016/B978-0-443-28993-4.00079-2

Introduction

The optimization of processes in operations and supply chain management plays an essential role in ensuring the survival of enterprises in nowadays highly competitive markets. Optimized operations allow cost reduction, profit maximization, and effectiveness enhancements. Many operations and supply chain management problems can be formulated in the context of combinatorial optimization, which consists of finding optimal solutions to problems defined over a discrete set of feasible solutions. Typical problems in this category are related to facility location and layout, production planning and scheduling, personnel scheduling, freight consolidation, and delivery of goods and services, among others. Some of these problems cannot be solved to optimality in reasonable computation times due to their intrinsic complexity or their size. Furthermore, reaching optimal solutions is meaningless in many practical situations since we often deal with rough simplifications of reality and the available data is not precise.

Metaheuristics such as simulated annealing, tabu search, greedy randomized adaptive search procedures (GRASPs), variable neighborhood search, genetic algorithms, and others, are among the most successful and widely used techniques to tackle optimization problems related to operations in manufacturing, services, supply chain management, and logistics. Moreover, as noticed by Lourenço and Ravetti (2018), heuristics and metaheuristics are two of the best optimization tools to be used in solving and providing business insights for supply chain management problems related to the management of all activities along a network of organizations to provide a good or service to final customers. Chen and Hall (2022) also discuss the role of metaheuristics in supply chain scheduling.

The remainder of this chapter is structured as follows: Section "Basic concepts in optimization: From construction to metaheuristics" provides the fundamental nomenclature, definitions, and building blocks of optimization with metaheuristics. Section "Metaheuristics" reviews the basic principles and ideas involved in the design of the main and most widely and successfully used metaheuristics. References to seminal works are provided throughout the text. Section "Applications to operations and supply chain management" reports applications of metaheuristics to operations and supply chain management. Section "Conclusions" concludes the article with some final discussions.

Basic concepts in optimization: From construction to metaheuristics

Definition 1: A *combinatorial optimization problem* can be formulated as the constrained minimization (or maximization) of some function $f(S) : S \in F$. Here, E is a discrete ground set formed by the elements that form each problem solutions S, $F \subseteq 2^E$ is the set of feasible solutions, and $f : 2^E \to \mathbb{R}$ is the objective function.

Definition 2: $S^* \in F$ is an *optimal solution* (or, simply, an *optimum*) to the above problem if $f(S^*) \leq f(S)$ (in this case, S^* is a minimum) or if $f(S^*) \geq f(S)$ (in this case, S^* is a maximum), for all $S \in F$. An optimal solution is also known as a *global optimal solution*.

Definition 3: A method (or algorithm) for solving an optimization problem is *exact* if it is guaranteed to produce, in finite time, a global optimum for this problem and a proof of its optimality, in case one exists, or otherwise to show that no feasible solution exists.

Global optimal solutions are often referred to as *exact solutions*. Exact solution methods for combinatorial optimization problems follow different algorithmic paradigms.

Efficient (i.e., running in polynomial time) exact algorithms are unknown (and are unlikely to exist) for a broad class of optimization problems classified as NP-hard. These problems are often referred to as *intractable*. Although the size of the problems that can be solved exactly to optimality has been continuously increasing due to algorithmic and technological developments, there are problems (or problem instances) that are not amenable to being solved by exact methods. Garey and Johnson's (1979) is the most influential textbook on computational complexity, introducing the theory of NP-completeness and computer intractability.

Definition 4: A method for solving an optimization problem is *approximate* or *heuristic* if it aims at providing feasible solutions that are not necessarily optimal.

Approximate methods usually have lower running times than exact methods. They can handle larger problem instances than exact methods, at the expense of solution quality.

Definition 5: *Constructive heuristics* are those that build a feasible solution step by step from scratch.

In their simplest form, constructive heuristics are often based on low-complexity greedy algorithms.

Definition 6: In a *greedy algorithm*, a feasible solution is built step-by-step from scratch. At each step, the element from E incorporated to the solution S under construction is that with the smallest (resp. greatest) incremental cost in the case of a minimization (resp. maximization) problem until a feasible solution in F is obtained.

The solution obtained by a greedy algorithm is not necessarily optimal. Greedy algorithms are often used to build initial solutions that will be explored by local search or metaheuristics. Chapter 7 of Lawler (1976) and Chapter 16 of Cormen et al. (2022) cover greedy algorithms.

Definition 7: *Greedy randomized algorithms* (or *semigreedy algorithms*) follow the same principle of greedy algorithms but use randomization at the selection step.

Therefore, greedy randomized algorithms can produce different solutions in different runs of a multistart procedure. Semigreedy algorithms were introduced by Hart and Shogan (1987) and independently developed by Feo and Resende (1989).

Definition 8: A *local search* heuristic starts from any feasible solution and improves it by successive minor modifications that visit other (feasible or infeasible) solutions until a solution that cannot be further improved is encountered.

Local improvements are evaluated concerning neighbor solutions that can be obtained by slight modifications applied to the current solution.

Definition 9: A *neighborhood* $N(S)$ of a feasible solution $S \in F$ can be defined as any subset of F. More formally, a neighborhood is a mapping associating each feasible solution $S \in F$ with a subset $N(S) = \{S_1, \ldots, S_p\} \subseteq F$ of feasible solutions.

Each solution $S' \in N(S)$ can be reached from S by an operator called a *move*. Typically, two neighboring solutions S and $S' \in N(S)$ differ only by a few elements of the ground set, and a move from S to S' consists simply of modifying one or more elements in S. Usually, $S \in N(S')$ if and only if $S' \in N(S)$.

Definition 10: A solution S is a local optimum concerning a neighborhood N if there is no strictly better solution in $N(S)$ than S.

Local search is a common component of most metaheuristics. Yagiura and Ibaraki (2002) traced the history of local search since the work of Croes (1958). Hoos and Stützle (2005) developed a thorough study of the foundations and applications of stochastic local search, that is, methods based on local search that use randomization to generate or select candidate solutions. In their seminal work, Lin and Kernighan (1973) developed a local search heuristic based on 2-opt and 3-opt exchanges for approximately solving the symmetric traveling salesman problem, one of the best approaches for the problem. Michiels et al. (2007) discussed theoretical aspects of local search.

Although local search methods often provide high-quality solutions whose values are close to those of the optimal solutions, they can become prematurely trapped in low-quality, locally optimal solutions.

Definition 11: *Metaheuristics* are general high-level procedures that coordinate simple methods and rules to find high-quality solutions to optimization problems. They are based on distinct paradigms and offer different mechanisms to escape from locally optimal solutions (as opposed to greedy algorithms or local search methods).

Metaheuristics are among the most effective strategies for solving challenging combinatorial optimization problems in practice. They often produce much better solutions than those obtained by the simple heuristics and rules they involve. Section "Metaheuristics" recalls the principles and templates of some of the most frequently used metaheuristics, which have been instrumental and contributed to most developments and applications in the field.

Metaheuristics

In this section, we review the basic principles and ideas involved in the design of some of the main metaheuristics. We describe five widely and successfully used metaheuristics: simulated annealing, tabu search, GRASP, VNS, and genetic algorithms. Sections "Simulated annealing" to "Variable neighborhood search" provide examples of trajectory-based metaheuristics in which the initial solution evolves by neighborhood moves (together with perturbations and restarts) until the method stops. Section "Genetic algorithms" presents an introduction to genetic algorithms, which are a typical example of a population-based metaheuristic: a population of solutions evolves along the iterations (named generations in this case) by different types of operations (typically, crossover, mutation, and selection), and it is not possible to trace the individual trajectory of each specific solution.

The interested reader in the field of metaheuristics and the many existing approaches, some of them summarized in the following sections, may check the broad and ever-evolving literature on the subject, in particular, the handbooks edited by Reeves (1993b), Glover and Kochenberger (2003), Burke and Kendall (2005, 2014), and Gendreau and Potvin (2010, 2019). Sörensen (2015), de Armas et al. (2022), Camacho-Villalón et al. (2022), and Camacho-Villalón et al. (2023) offer a critical view of the explosion of metaheuristic methods based on metaphors of some natural, bestial, or artificial processes.

Simulated annealing

The principle of simulated annealing lies in the analogy between the physical annealing process and the combinatorial optimization problem to be solved. In physics, annealing is a thermal process for obtaining the low-energy states of a solid in a heat bath. Solutions to the combinatorial problem are equivalent to states of the physical process, while the cost of each solution corresponds to the state's energy. Neighbor solutions correspond to states generated by a perturbation method. At each step of the simulated annealing, a new solution S' in the neighborhood of the current solution S is randomly generated. The new solution replaces the current solution if it improves the latter or with a decreasing probability in case it does not. The best solution found is returned by the algorithm. It is a memoryless procedure that does not require the implementation of a neighborhood search procedure.

The main advantage of simulated annealing is its simplicity of implementation because there is no local search to be performed. However, its convergence speed and solution quality depend on several parameter values and implementation choices that are not easy to tune. Furthermore, the algorithm is memoryless and does not use cost information about the entire neighborhood.

The work on optimization by simulated annealing was pioneered by Kirkpatrick et al. (1983), with accounts of later developments and applications in textbooks (Aarts and Korst, 1989; van Laarhoven and Aarts, 1987). Convergence and implementation strategies are also discussed in Aarts and Korst (2002) and Henderson et al. (2003).

Tabu search

Tabu search is a dynamic neighborhood method that uses memory to drive the search by escaping from local optima and avoiding cycling. Contrarily to memoryless heuristics such as simulated annealing, and to methods that use rigid memory structures such as branch-and-bound, tabu search makes use of flexible and adaptive memory designs.

For any solution S, in the case of tabu search, the neighborhood $N(S)$ is not a static set and can be modified according to the history of the search. At each local search iteration, tabu search looks for the neighbor solution that best improves the objective function (or, if none exists, for the one that least deteriorates the current solution). However, some neighbors in $N(S)$ are forbidden and discarded. The set of forbidden (or *tabu*) neighbors is stored in a short-term memory, formed by the most recently visited solutions, to prevent the search from returning to previously visited solutions. Contrary to simple local search procedures, tabu search chooses the move that least deteriorates the objective function value whenever there are no improving moves.

The basic tabu search scheme can be extended by incorporating more sophisticated strategies: aspiration criteria to override the tabu status of unvisited solutions, medium- and long-term memories to implement intensification and diversification procedures, candidate list strategies to speed up the search, and hashing tables to accelerate and filter the search. These comprehensive strategies lead to much more powerful and effective implementations in terms of the quality of the solutions they obtain.

Tabu search is undoubtedly among the most effective approaches for solving hard combinatorial optimization problems. However, implementations of tabu search often involve setting many parameters, which have to be appropriately tuned for achieving good performance in practice.

The seminal papers of Glover (1989, 1990) established the fundamentals, extensions, and uses of tabu search. They provided solid foundations and originated most of the developments from where the field of metaheuristics flourished, as reported in the retrospective by Glover (2022). The reader is referred to the textbook of Glover and Laguna (1997) for a thorough study of tabu search and related ideas. Glover et al. (1993) gave an updated presentation of the fundamental ideas of tabu search with some hints for an efficient implementation. A variety of applications of tabu search with basic modeling suggestions were described by Hertz et al. (1997).

Greedy randomized adaptive search procedures

A GRASP is a metaheuristic resulting from the hybridization of a greedy randomized algorithm with a local search method embedded in a multistart framework.

An especially appealing characteristic of GRASP is its ease of implementation. Few parameters must be set and tuned; therefore, development can focus on implementing efficient data structures to ensure fast computations in all iterations. Basic implementations of GRASP rely almost exclusively on two parameters. The first is the stopping criterion, often defined by the maximum number of iterations (or by the maximum number of iterations without improvement in the best solution found). The second is a parameter used to limit the size of the restricted candidate list within the greedy randomized algorithm used in the construction phase. GRASP is also amenable to efficient and straightforward parallel implementations, see, for example, Ribeiro and Rosseti (2002, 2007). Despite its simplicity and ease of implementation, GRASP is a very effective metaheuristic and produces the best known solutions for many problems.

GRASP was proposed by Feo and Resende (1989); see also Feo and Resende (1995). Reviews and tutorials appeared in Ribeiro (2002) and Resende and Ribeiro (2003b, 2005, 2010, 2014). Festa and Resende (2002, 2009) presented reviews of applications of GRASP. The book by Resende and Ribeiro (2016b) consolidated the fundamentals, advances, and extensions of optimization with GRASP.

GRASP, as initially proposed, is a memoryless procedure in which each iteration does not use information gathered in previous iterations. In other words, all its iterations are independent. Path relinking enhances the basic GRASP significantly, adding a long-term memory mechanism to GRASP heuristics. GRASP with path relinking implements long-term memory using an elite set of diverse and high-quality solutions previously found during the search. In its most basic implementation, path relinking is applied at each iteration between the solution found at the end of the local search phase and any randomly selected solution from the elite set. The solution resulting from path relinking is a candidate for inclusion in the elite set. The hybridization of GRASP with path relinking can significantly improve the solution time and quality.

Path relinking was initially proposed by Glover (1996) as an intensification strategy exploring trajectories connecting elite solutions obtained by tabu search or scatter search. The use of path relinking embedded in a GRASP procedure as an intensification strategy applied to each locally optimal solution was first proposed in Laguna and Martí (1999). Several strategies (backward, forward, backward-and-forward, mixed, truncated, randomized, evolutionary) have been considered and combined with extensions and improvements of path relinking in successful implementations hybridized with tabu search, GRASP, and genetic algorithms;

see, e.g., Canuto et al. (2001), Reeves and Yamada (1998), Festa et al. (2002), Ribeiro and Rosseti (2002), Ribeiro et al. (2002), Resende and Ribeiro (2003a, b), Souza et al. (2003), Ribeiro et al. (2007), Martins et al. (2004), Resende and Werneck (2004), Resende and Ribeiro (2005), and Ribeiro and Vianna (2009). Resende and Ribeiro (2005) reviewed these strategies, showing that they involve trade-offs between computation time and solution quality. Chapters 8 and 9 of Resende and Ribeiro (2016b) reviewed path relinking strategies and hybridizations with GRASP in detail.

Variable neighborhood search

Variable neighborhood search (VNS) is based on a systematic change in the neighborhood. It makes use of a finite set of k_{max} preselected neighborhood structures identified as $N_1, \ldots, N_{k_{max}}$ that may be induced by one or more metric functions introduced in the solution space. Let $N_k(S)$ denote the set of solutions in the k-th neighborhood of S.

At each VNS iteration, a solution is randomly sampled in the k-th neighborhood of the current solution S and a local search is applied to the resulting solution. If the local minimum obtained does not improve the current solution, then the algorithm moves to the $(k + 1)$-th neighborhood.

VNS is also a memoryless heuristic. It is easy to implement and relies on a few parameters: the stopping criterion and the number k_{max} of neighborhoods (and their definitions). It was originally proposed by Mladenovic and Hansen (1997), see Hansen and Mladenović (1999, 2002, 2003) for reviews.

Genetic algorithms

A genetic algorithm is a population-based method that uses selection, recombination, and mutation operators to generate new solutions in the search space, imitating the process of natural selection and evolution of species. Solutions are evaluated in terms of their fitness, which most often corresponds to the value of the objective function itself.

Contrary to the other metaheuristics reviewed in the previous sections, genetic algorithms explore a population of solutions that evolve along some generations rather than a single trajectory emanating from a unique initial solution. The method starts from a population formed by a fixed number of feasible solutions that can be randomly generated or constructed by a more elaborate approach such as a greedy randomized algorithm. At each iteration (which is called a generation in this context), pairs of randomly selected solutions (parent solutions) are combined, and new solutions (offspring) are generated (crossover operator). Some randomly selected solutions go through small random modifications in their structure (mutation operator) to ensure diversification. The best-fitted solutions of the population are selected and passed to the new generation (selection operator), and a new iteration resumes.

Hybrid genetic algorithms, also called memetic algorithms, use specific knowledge available to solve the problem. Such strategies incorporate greedy randomized algorithms to create the initial population and the application of local search to some elements of the population of each generation. Genetic algorithms hybridized with optimization strategies usually perform much better than basic genetic algorithms.

Genetic algorithms were first presented in the book of Holland (1975); see also, for example, Goldberg (1989), Holland (1992), Reeves and Rowe (2002), and Michalewicz (1996), among others. They are very appealing and easy to implement. However, many implementation choices exist for each of the above operators. Exemplary implementations require appropriate solution encodings as finite-length data strings and well-tuned strategies for the initial population generation, mate selection, crossover, mutation, and selection; see, for example, Reeves (1993a, 2003).

A significant difficulty with implementing genetic algorithms resides in the crossover operator. It occurs because, in most cases, the solutions obtained by crossover turn out to be infeasible and, therefore, unusable by the algorithm. Although some strategies have been proposed in the literature to handle infeasibilities (such as the use of penalties or repair operators), they are problem-specific and of complex generalized implementation in practice, with a limited number of successful cases.

A *biased random-key genetic algorithm* (BRKGA) encodes solutions as a vector of real numbers called *random keys*. A deterministic algorithm (called a *decoder*) takes a vector of random keys as the input and associates with it a feasible solution of the combinatorial optimization problem at hand, for which an objective value (or fitness function) can be computed. The crossover is said to be biased because one parent is always an elite (high-quality) solution, while the other is not. The elite parent still has a higher probability of passing its elements to the new generation. Biased random-key genetic algorithms can be implemented straightforwardly. The crossover is performed in the space of random keys, and the corresponding solutions are built by the decoder a posteriori. Therefore, the resulting solutions are always feasible. BRKGAs are particularly effective for scheduling problems or other problems whose solutions may be represented by permutation vectors; see Goncalves and Resende (2011), Resende and Ribeiro (2016a), Noronha and Ribeiro (2024), and Londe et al. (2025) for reviews and Buriol et al. (2005), Noronha et al. (2011), Brandão et al. (2015, 2017), Pinto et al. (2020) for some applications.

Applications to operations and supply chain management

Metaheuristics are essential to solve complex problems in operations and supply chain management. This section provides a few examples of successful metaheuristics applications in different sectors.

Manufacturing and service operations

In manufacturing, metaheuristics are widely applied to a variety of problems in several areas, including lot sizing (Yang et al., 2024), job shop scheduling (Ahmadian et al., 2021), flow shop scheduling (Zobolas et al., 2009), parallel machine scheduling (Moser et al., 2022), and harvest planning (Gómez-Lagos et al., 2021; Legues et al., 2007).

Examples of service operations that can benefit from metaheuristics encompass airport gate assignment (Karsu et al., 2021), aircraft scheduling (Atkin et al., 2007), waste collection (Jorge et al., 2022), rebalancing of resources in bike-sharing systems (Lu et al., 2020), technician routing and scheduling (Mathlouthi et al., 2021), maintenance scheduling (Froger et al., 2018), and railway planning (Canca et al., 2017).

Supply chain management and logistics operations

Some applications of metaheuristics in the context of supply chain management and operations include pharmaceutical supply chains (Goodarzian et al., 2021), integrated manufacturing and port operations (Cheimanoff et al., 2023), and sustainable closed-loop supply chain network design (Devika et al., 2014).

Applications of metaheuristics to logistics exist in multidepot vehicle routing (Vidal et al., 2012), vehicle routing with drones (Sacramento et al., 2019), vehicle routing in city logistics (Rincon-Garcia et al., 2020), order picking in smart warehouses (Zhao et al., 2024), container retrieval (da Silva Firmino et al., 2019). In reverse logistics, we can find examples such as green vehicle routing (Santos et al., 2023) and integrated collection-disassembly-and-routing operations (Lei et al., 2024).

Healthcare operations

Some examples of applications to healthcare operations are home-healthcare routing and scheduling (Hiermann et al., 2015), nurse rostering (Ceschia et al., 2023), operating room planning and scheduling (Aringhieri et al., 2015), and biological samples transportation (Benini et al., 2022).

Humanitarian operations

Metaheuristic applications to humanitarian operations include disaster relief (Yi and Kumar, 2007), aid distribution (Ferrer et al., 2016), path planning in search and rescue missions using unmanned aerial vehicles (Morin et al., 2023), and postdisaster management assessment (Adsanver et al., 2025).

Conclusions

An essential aspect of a supply chain is integrating and coordinating all activities since decisions in one component directly affect the whole supply chain. Companies must refrain from suboptimization by managing the entire supply chain as a single entity. Powerful and robust techniques are necessary to respond to this challenge; see Lourenço (2001).

As discussed in Section "Basic concepts in optimization: From construction to metaheuristics," approximate methods and metaheuristics have many advantages in the solution of such problems concerning exact optimization methods in terms of flexibility and computer resources such as computation times and memory requirements, in particular when the problem size and the complexity of the models increase.

The modular nature of metaheuristics, as examined in Section "Metaheuristics," allows short development times, particularly important in implementing complex decision support systems. Furthermore, metaheuristics are simple, easy to implement, robust, and very effective in solving very hard and complex problems.

Decision support systems that combine simulation and metaheuristics, also known in the literature as *simheuristics* (Juan et al., 2015), offer another excellent opportunity for metaheuristics. Examples can be found in Bock (2010), Juan et al. (2018), and Ozgormus and Smith (2020). Their use as a surrogate for exact optimization methods leads to a more flexible framework where the user can analyze more and better scenarios within the same time environment.

Takeaways for operations managers

Operations managers can learn from this chapter that metaheuristics offer a good alternative to obtaining good-quality solutions for challenging problems in operations and supply chain management. Their modular nature allows short development times, which are particularly important in implementing complex decision support systems. Furthermore, they are simple, easy to implement, robust, and very effective in solving very hard and complex problems. In addition, they have the advantage of not relying on licensed commercial software. Metaheuristics are, in general, problem-tailored. There are several metaheuristic frameworks from which to choose. Such a choice depends on the problem characteristics. It can be guided by research results and by the practitioner's experience with the different frameworks.

Takeaways for researchers

Researchers can find in this chapter the main concepts of some selected well-established and scientifically sound metaheuristics to tackle their research problem. Obtaining state-of-the-art results often require efficient implementations of the involved metaheuristic operators. The researchers interested in more theoretical work can find many possibilities related to the structural and algorithmic aspects of the used operators. Modern challenges related to large amounts of data offer a fertile ground of research opportunities involving data-driven metaheuristics that make use of simulation or machine learning methods.

Acknowledgments

The work of Celso C. Ribeiro was supported by research grants CNPq 309869/2020-0 and FAPERJ E-26/200.926/2021. The work of Rafael A. Melo was partially supported by research grants CNPq 314718/2023-0 and FAPESB INCITE PIE0002/2022.

References

Aarts EHL and Korst J (1989) *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley.
Aarts E and Korst J (2002) Selected topics in simulated annealing. In: Ribeiro CC and Hansen P (eds.) *Essays and Surveys in Metaheuristics*, pp. 1–37. Kluwer.
Adsanver B, Coban E, and Balcik B (2025) A predictive multistage postdisaster damage assessment framework for drone routing. *International Transactions in Operational Research* 32: 626–668.
Ahmadian MM, Salehipour A, and Cheng TCE (2021) A meta-heuristic to solve the just-in-time job-shop scheduling problem. *European Journal of Operational Research* 288: 14–29.
Aringhieri R, Landa P, Soriano P, Tanfani E, and Testi A (2015) A two level metaheuristic for the operating room scheduling and assignment problem. *Computers & Operations Research* 54: 21–34.
Atkin JAD, Burke EK, Greenwood JS, and Reeson D (2007) Hybrid metaheuristics to aid runway scheduling at London Heathrow airport. *Transportation Science* 41: 90–106.
Benini M, Detti P, and de Lara GZM (2022) Mathematical programming formulations and metaheuristics for biological sample transportation problems in healthcare. *Computers & Operations Research* 146: 105921.
Bock S (2010) Real-time control of freight forwarder transportation networks by integrating multimodal transport chains. *European Journal of Operational Research* 200: 733–746.
Brandão JS, Noronha TF, Resende MGC, and Ribeiro CC (2015) A biased random-key genetic algorithm for single-round divisible load scheduling. *International Transactions in Operational Research* 22: 823–839.
Brandão JS, Noronha TF, Resende MGC, and Ribeiro CC (2017) A biased random-key genetic algorithm for scheduling heterogeneous multi-round systems. *International Transactions in Operational Research* 27: 1061–1077.
Buriol LS, Resende MGC, Ribeiro CC, and Thorup M (2005) A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. *Networks* 46: 36–56.
Burke EK and Kendall G (2005) *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer.
Burke EK and Kendall G (2014) *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, 2nd edn. Springer.
Camacho-Villalón CL, Dorigo M, and Stützle T (2022) An analysis of why cuckoo search does not bring any novel ideas to optimization. *Computers & Operations Research* 142: 105747.
Camacho-Villalón CL, Dorigo M, and Stützle T (2023) Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: Six misleading optimization techniques inspired by bestial metaphors. *International Transactions in Operational Research* 30: 2945–2971.
Canca D, De-Los-Santos A, Laporte G, and Mesa JA (2017) An adaptive neighborhood search metaheuristic for the integrated railway rapid transit network design and line planning problem. *Computers & Operations Research* 78: 1–14.
Canuto SA, Resende MGC, and Ribeiro CC (2001) Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks* 38: 50–58.
Ceschia S, Di Gaspero L, Mazzaracchio V, Policante G, and Schaerf A (2023) Solving a real-world nurse rostering problem by simulated annealing. *Operations Research for Health Care* 36: 100379.
Cheimanoff N, Féniès P, Kitri MN, and Tchernev N (2023) Exact and metaheuristic approaches to solve the integrated production scheduling, berth allocation and storage yard allocation problem. *Computers & Operations Research* 153: 106174.
Chen Z-L and Hall N (2022) *Supply Chain Scheduling*. Cham: Springer.
Cormen TH, Leiserson CE, Rivest RL, and Stein C (2022) *Introduction to Algorithms*, 4th edn. MIT Press.
Croes GA (1958) A method for solving traveling-salesman problems. *Operations Research* 6: 791–812.
da Silva Firmino A, de Abreu Silva RM, and Times VC (2019) A reactive GRASP metaheuristic for the container retrieval problem to reduce crane's working time. *Journal of Heuristics* 25: 141–173.
de Armas J, Lalla-Ruiz E, Tilahun SL, and Voß S (2022) Similarity in metaheuristics: A gentle step towards a comparison. *Natural Computing* 21: 265–287.
Devika K, Jafarian A, and Nourbakhsh V (2014) Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *European Journal of Operational Research* 235: 594–615.
Feo TA and Resende MGC (1989) A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8: 67–71.
Feo TA and Resende MGC (1995) Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6: 109–133.
Ferrer JM, Ortuño MT, and Tirado G (2016) A GRASP metaheuristic for humanitarian aid distribution. *Journal of Heuristics* 22: 55–87.
Festa P and Resende MGC (2002) GRASP: An annotated bibliography. In: Ribeiro CC and Hansen P (eds.) *Essays and Surveys in Metaheuristics*, pp. 325–367. Kluwer.
Festa P and Resende MGC (2009) An annotated bibliography of GRASP, Part II: Applications. *International Transactions in Operational Research* 16: 131–172.
Festa P, Pardalos PM, Resende MGC, and Ribeiro CC (2002) Randomized heuristics for the max-cut problem. *Optimization Methods and Software* 7: 1033–1058.
Froger A, Gendreau M, Mendoza JE, Pinson E, and Rousseau L-M (2018) Solving a wind turbine maintenance scheduling problem. *Journal of Scheduling* 21: 53–76.
Garey MR and Johnson DS (1979) *Computers and Intractability - A Guide to the Theory of NP-completeness*. New York: W.H. Freeman and Company.
Gendreau M and Potvin J-Y (2010) *Handbook of Metaheuristics*, 2nd edn. Springer.
Gendreau M and Potvin J-Y (2019) *Handbook of Metaheuristics*, 3rd edn. Springer.
Glover F (1989) Tabu search - Part I. *ORSA Journal on Computing* 1: 190–206.
Glover F (1990) Tabu search - Part II. *ORSA Journal on Computing* 2: 4–32.
Glover F (1996) Tabu search and adaptive memory programming - Advances, applications and challenges. In: Barr RS, Helgason RV, and Kennington JL (eds.) *Interfaces in Computer Science and Operations Research*, pp. 1–75. Kluwer.
Glover F (2022) Unforeseen consequences of "Tabu" choices - A retrospective. *INFORMS Journal on Computing* 34: 1306–1308.

Glover F and Kochenberger G (2003) *Handbook of Metaheuristics*. Kluwer.
Glover F and Laguna M (1997) *Tabu Search*. Kluwer.
Glover F, Taillard E, and de Werra D (1993) A user's guide to tabu search. In: Glover F, Laguna M, Taillard E, and de Werra D (eds.) *Tabu Search. Annals of Operations Research*, Vol. 41, pp. 3–28. Springer.
Goldberg DE (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
Gómez-Lagos JE, González-Araya MC, Soto-Silva WE, and Rivera-Moraga MM (2021) Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach. *European Journal of Operational Research* 290: 297–312.
Goncalves JF and Resende MGC (2011) Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics* ISSN: 1381-1231. 17: 487–525. https://doi.org/10.1007/s10732-010-9143-1.
Goodarzian F, Kumar V, and Ghasemi P (2021) A set of efficient heuristics and meta-heuristics to solve a multi-objective pharmaceutical supply chain network. *Computers & Industrial Engineering* 158: 107389.
Hansen P and Mladenović N (1999) An introduction to variable neighbourhood search. In: Voss S, Martello S, Osman IH, and Roucairol C (eds.) *Metaheuristics: Advances and Trends in Local Search Procedures for Optimization*, pp. 433–458. Kluwer.
Hansen P and Mladenović N (2002) Developments of variable neighborhood search. In: Ribeiro CC and Hansen P (eds.) *Essays and Surveys in Metaheuristics*, pp. 415–439. Kluwer.
Hansen P and Mladenović N (2003) Variable neighborhood search. In: Glover F and Kochenberger G (eds.) *Handbook of Metaheuristics*, pp. 145–184. Kluwer.
Hart JP and Shogan AW (1987) Semi-greedy heuristics: An empirical study. *Operations Research Letters* 6: 107–114.
Henderson D, Jacobson SH, and Johnson AW (2003) The theory and practice of simulated annealing. In: Glover F and Kochenberger G (eds.) *Handbook of Metaheuristics*, pp. 287–319. Kluwer.
Hertz A, Taillard E, and de Werra D (1997) Tabu search. In: Aarts E and Lenstra JK (eds.) *Local Search in Combinatorial Optimization*, pp. 121–143. Wiley.
Hiermann G, Prandtstetter M, Rendl A, Puchinger J, and Raidl GR (2015) Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research* 23: 89–113.
Holland JH (1975) *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
Holland JH (1992) Genetic algorithms. *Scientific American* 267: 44–50.
Hoos HH and Stützle T (2005) *Stochastic Local Search: Foundations and Applications*. Elsevier.
Jorge D, Antunes AP, Ramos TRP, and Barbosa-Póvoa AP (2022) A hybrid metaheuristic for smart waste collection problems with workload concerns. *Computers & Operations Research* 137: 105518.
Juan AA, Faulin J, Grasman SE, Rabe M, and Figueira G (2015) A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives* 2: 62–72.
Juan AA, Kelton WD, Currie CS, and Faulin J (2018) Simheuristics applications: Dealing with uncertainty in logistics, transportation, and other supply chain areas. In: *2018 Winter Simulation Conference (WSC)*, pp. 3048–3059.
Karsu Ö, Azizoğlu M, and Alanlı K (2021) Exact and heuristic solution approaches for the airport gate assignment problem. *Omega* 103: 102422.
Kirkpatrick S, Gelatt CD Jr, and Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598): 671–680.
Laguna M and Martí R (1999) GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing* 11: 44–52.
Lawler EL (1976) *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston.
Legues AD, Ferland JA, Ribeiro CC, Vega JR, and Weintraub A (2007) A tabu search approach for solving a difficult forest harvesting machine location problem. *European Journal of Operational Research* 179: 788–805.
Lei J, Che A, and Van Woensel T (2024) Collection-disassembly-delivery problem of disassembly centers in a reverse logistics network. *European Journal of Operational Research* 313: 478–493.
Lin S and Kernighan BW (1973) An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* 21: 498–516.
Londe MA, Pessoa LS, Andrade CE, and Resende MGC (2025) Biased random-key genetic algorithms: A review. *European Journal of Operational Research* 321: 1–22.
Lourenço HR (2001) *Supply chain management: An opportunity for metaheuristics*. UPF Economics and Business Paper No. 538, online reference available at. http://doi.org/10.2139/ssrn.273425. last access on March 6, 2024.
Lourenço HR and Ravetti M (2018) Supply chain management. In: Martí R, Pardalos PM, and Resende MGC (eds.) *Handbook of Heuristics*, pp. 1241–1258. Cham: Springer.
Lu Y, Benlic U, and Wu Q (2020) An effective memetic algorithm for the generalized bike-sharing rebalancing problem. *Engineering Applications of Artificial Intelligence* 95: 103890.
Martins SL, Ribeiro CC, and Rosseti I (2004) Applications and parallel implementations of metaheuristics in network design and routing. In: Manandhar S, Austin J, Desai U, Oyanagi Y, and Talukder AK (eds.) *Applied Computing. Lecture Notes in Computer Science*, Vol. 3285, pp. 205–213. Springer.
Mathlouthi I, Gendreau M, and Potvin J-Y (2021) A metaheuristic based on tabu search for solving a technician routing and scheduling problem. *Computers & Operations Research* 125: 105079.
Michalewicz Z (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer.
Michiels W, Aarts EHL, and Korst J (2007) *Theoretical Aspects of Local Search*. Springer.
Mladenovic N and Hansen P (1997) Variable neighborhood search. *Computers & Operations Research* 34: 1097–1100.
Morin M, Abi-Zeid I, and Quimper C-G (2023) Ant colony optimization for path planning in search and rescue operations. *European Journal of Operational Research* 305: 53–63.
Moser M, Musliu N, Schaerf A, and Winter F (2022) Exact and metaheuristic approaches for unrelated parallel machine scheduling. *Journal of Scheduling* 25: 507–534.
Noronha TF and Ribeiro CC (2024) Biased random-key genetic algorithms: A tutorial with applications. In: *ISMSI 2024: Proceedings of the 2024 8th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI)*, pp. 110–115. New York: ACM.
Noronha TF, Resende MGC, and Ribeiro CC (2011) A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization* 50: 503–518.
Ozgormus E and Smith AE (2020) A data-driven approach to grocery store block layout. *Computers & Industrial Engineering* 139: 105562.
Pinto BQ, Ribeiro CC, Rosseti I, and Noronha TF (2020) A biased random-key genetic algorithm for routing and wavelength assignment under a sliding scheduled traffic model. *Journal of Global Optimization* 77: 949–973.
Reeves CR (1993a) Genetic algorithms. In: Reeves CR (ed.) *Modern Heuristic Techniques for Combinatorial Problems*, pp. 151–196. Wiley.
Reeves CR (1993b) *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell.
Reeves C (2003) Genetic algorithms. In: Glover F and Kochenberger G (eds.) *Handbook of Metaheuristics*, pp. 65–82. Kluwer.
Reeves C and Rowe JE (2002) *Genetic Algorithms: Principles and Perspectives*. Springer.
Reeves CR and Yamada T (1998) Genetic algorithms, path relinking and the flowshop sequencing problem. *Evolutionary Computation* 6: 45–60.
Resende MGC and Ribeiro CC (2003a) A GRASP with path-relinking for private virtual circuit routing. *Networks* 41: 104–114.
Resende MGC and Ribeiro CC (2003b) Greedy randomized adaptive search procedures. In: Glover F and Kochenberger G (eds.) *Handbook of Metaheuristics*, pp. 219–249. Kluwer.
Resende MGC and Ribeiro CC (2005) GRASP with path-relinking: Recent advances and applications. In: Ibaraki T, Nonobe K, and Yagiura M (eds.) *Metaheuristics: Progress as Real Problem Solvers*, pp. 29–63. Springer.
Resende MGC and Ribeiro CC (2010) Greedy randomized adaptive search procedures: Advances and applications. In: Gendreau M and Potvin J-Y (eds.) *Handbook of Metaheuristics*, pp. 293–319. Springer.
Resende MGC and Ribeiro CC (2014) GRASP: Greedy randomized adaptive search procedures. In: Burke EK and Kendall G (eds.) *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pp. 287–312. Springer.

Resende MGC and Ribeiro CC (2016a) Biased-random key genetic algorithms: An advanced tutorial. In: *Companion Volume - Proceedings of the 2016 Genetic and Evolutionary Computation Conference - GECCO'16*, pp. 483–514. ACM.
Resende MGC and Ribeiro CC (2016b) *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*. Springer.
Resende MGC and Werneck RF (2004) A hybrid heuristic for the p-median problem. *Journal of Heuristics* 10: 59–88.
Ribeiro CC (2002) GRASP: Une métaheuristique gloutone et probabiliste. In: Teghem J and Pirlot M (eds.) *Optimisation approchée en recherche opérationnelle*, pp. 153–176. Hermès.
Ribeiro CC and Rosseti I (2002) A parallel GRASP heuristic for the 2-path network design problem. In: Monien B and Feldmann R (eds.) *Euro-Par 2002 Parallel Processing*. *Lecture Notes in Computer Science*, Vol. 2400, pp. 922–926. Springer.
Ribeiro CC and Rosseti I (2007) Efficient parallel cooperative implementations of GRASP heuristics. *Parallel Computing* 33: 21–35.
Ribeiro CC and Vianna DS (2009) A hybrid genetic algorithm for the phylogeny problem using path-relinking as a progressive crossover strategy. *International Transactions in Operational Research* 16: 641–657.
Ribeiro CC, Uchoa E, and Werneck RF (2002) A hybrid GRASP with perturbations for the Steiner problem in graphs. *INFORMS Journal on Computing* 14: 228–246.
Ribeiro CC, Martins SL, and Rosseti I (2007) Metaheuristics for optimization problems in computer communications. *Computer Communications* 30: 656–669.
Rincon-Garcia N, Waterson B, Cherrett TJ, and Salazar-Arrieta F (2020) A metaheuristic for the time-dependent vehicle routing problem considering driving hours regulations—An application in city logistics. *Transportation Research Part A: Policy and Practice* 137: 429–446.
Sacramento D, Pisinger D, and Ropke S (2019) An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies* 102: 289–315.
Santos MJ, Jorge D, Ramos T, and Barbosa-Póvoa A (2023) Green reverse logistics: Exploring the vehicle routing problem with deliveries and pickups. *Omega* 118: 102864.
Sörensen K (2015) Metaheuristics - The metaphor exposed. *International Transactions in Operational Research* 22: 1–16.
Souza MC, Duhamel C, and Ribeiro CC (2003) A GRASP with path-relinking heuristic for the capacitated minimum spanning tree problem. In: Resende MGC and Souza J (eds.) *Metaheuristics: Computer Decision Making*, pp. 627–658. Kluwer.
van Laarhoven PJM and Aarts E (1987) *Simulated Annealing: Theory and Applications*. Kluwer.
Vidal T, Crainic TG, Gendreau M, Lahrichi N, and Rei W (2012) A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research* 60: 611–624.
Yagiura M and Ibaraki T (2002) Local search. In: Pardalos PM and Resende MGC (eds.) *Handbook of Applied Optimization*, pp. 104–123. Oxford University Press.
Yang J, Zhang M, Feng J, and He K (2024) A genetic algorithm to solve the production lot-sizing problem with capacity adjustment. *Computers & Operations Research* 172: 106806.
Yi W and Kumar A (2007) Ant colony optimization for disaster relief operations. *Transportation Research Part E: Logistics and Transportation Review* 43: 660–672.
Zhao Z, Cheng J, Liang J, Liu S, Zhou M, and Al-Turki Y (2024) Order picking optimization in smart warehouses with human-robot collaboration. *IEEE Internet of Things Journal* 11: 16314–16324.
Zobolas G, Tarantilis CD, and Ioannou G (2009) Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research* 36: 1249–1267.