European Journal of Operational Research 321 (2025) 345–362

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Invited review in celebration of the 50th anniversary of EURO

Fifty years of metaheuristics

Rafael Martí[a,*], Marc Sevaux[b], Kenneth Sörensen[c]

[a] Department of Statistics and Operations Research, Universitat de València, Spain
[b] Lab-STICC, UMR 6285, CNRS, Université Bretagne Sud, France
[c] Department of Engineering Management, University of Antwerp, Belgium

ARTICLE INFO

Keywords:
Heuristics
Combinatorial optimization
Critical review
Metaheuristics

ABSTRACT

In this paper, we review the milestones in the development of heuristic methods for optimization over the last 50 years. We propose a critical analysis of the main findings and contributions, mainly from a European perspective. Starting with the roots of the area that can be traced back to the classical philosophers, we follow the historical path of heuristics and metaheuristics in the field of operations research and list the main milestones, up to the latest proposals to hybridize metaheuristics with machine learning. We pay special attention to the theories that changed our way of thinking about problem solving, and to the role played by the European Journal of Operational Research in the development of these theories. Our approach emphasizes methodologies and their connections with related areas, which permits to identify potential lines of future research.

1. Introduction

Optimization is the process of selecting or approximating the best possible solution from a set of possible alternatives. In mathematical terms, it involves determining the best configuration of an input (values of *decision variables*) that corresponds to the best value (minimum or maximum) of an output (the *objective function*), subject to a set of *constraints*. This best possible solution is called the *optimal solution* of the optimization problem.

For many optimization problems, efficient algorithms have been developed that could also be proven to always find the optimal solution. Famous examples include Kruskal's (Kruskal, 1956) and Prim's (Prim, 1957) algorithms for the minimum spanning tree problem, Dijkstra's algorithm for the shortest path problem (Dijkstra, 1959), Ford–Fulkerson's method for the max flow problem (Ford & Fulkerson, 1956), and the Hungarian method for the assignment problem (Kuhn, 1955).

These problem-specific algorithms, however, were only able to solve a single problem. To solve more general classes of optimization problems, *mathematical programming models* have been instrumental. Called *prescriptive* models, they translate complex real problems in business and industry into standard formulations that express the problem as a set of decision variables, an objective function, and (often many) constraints. When the resulting mathematical programming model meets certain criteria (e.g., when the objective function and constraints are all linear functions of the decision variables), standardized optimization methods can be used in order to find the optimal solution of the problem they represent. This successful approach to represent a real problem as a mathematical optimization model, proposed by George Dantzig together with the Simplex method around the 1950s (Dantzig, 1951), has lead to a large number of optimization methods to solve them.

However, the approach to use standardized solution methods on mathematical programming models that guarantee to find the optimal solution, has important limitations. These limitations surface when solving large-scale problems and problems that have some complicating mathematical characteristics such as discrete variables (combinatorial optimization problems), non-linear constraints, or a complex objective function. In such situations, finding the optimal solution becomes intractable, not because it is more complex to find it, but because it takes too much time to be practically feasible. The study of the relationship between the size and nature of an optimization problem and the time required to solve it, called *complexity theory* has provided insights into the deeper reasons why this is the case and why it is unlikely that, for many practical combinatorial optimization problems, we will ever find a fast algorithm to solve them in a reasonable amount of time.

Real problems require solutions, however, optimal or not. Research on solving large-scale combinatorial optimization problems has therefore out of necessity focused on the only practical solution in such a situation: to relax the requirement of finding the optimal solution and resort to a solution method that does not guarantee that the optimal solution will be found. Such a solution method is called a *heuristic optimization algorithm*, or simply *heuristic*.

* Corresponding author.
E-mail addresses: rafael.marti@uv.es (R. Martí), marc.sevaux@univ-ubs.fr (M. Sevaux), kenneth.sorensen@uantwerpen.be (K. Sörensen).

https://doi.org/10.1016/j.ejor.2024.04.004
Received 28 November 2023; Accepted 2 April 2024
Available online 9 April 2024

Heuristics have been around long before we coined the term. In fact, one could argue that one of the primary functions of the human brain is to solve optimization problems using heuristics. Whether choosing the best berries from a bush, determining the best way home from work, or selecting a car to purchase, our mind invariably uses some form of heuristic to make the decision. However, the specifics of these mental shortcuts often remain elusive to us. While heuristics have been a part of human decision-making for ages, their formal study is a relatively new endeavor. Much like fish in water, we are so surrounded by optimization processes and heuristics that it has proven to be a challenge to recognize them as subjects worthy of study and formalization.

Modern heuristics endeavor to understand the process of solving problems, especially the mental operations typically useful in this process. A serious study of heuristic should take into account both the logical and the psychological background, and should not neglect what historical authors as Pappus, Descartes, Leibnitz, and Bolzano have to say about the subject (Hertwig & Pachur, 2015), but it should least neglect unbiased experience. Experience in solving problems and experience in watching other people solve problems has traditionally formed the basis on which a heuristic is built. Heuristics have become a very popular family of solution methods for optimization problems because they are capable of finding acceptable solutions in a reasonable amount of time.

Researchers found out very early that simple problem-specific rules of thumb, like the nearest neighbor heuristic for the traveling salesperson problem, the best-fit and first-fit algorithms for bin packing problems, the Clarke–Wright algorithm for the vehicle routing problem (Clarke & Wright, 1964), and many others, could often find satisfactory solutions in far less time than their exact counterparts.

Exact methods, however, still had an advantage in the form of a general methodology to model and solve optimization problems for which no problem-specific methods were available. The equivalent of the linear and integer programming paradigm, but with the resulting algorithm being a heuristic, remained elusive. While mathematical programming methods could always be truncated, i.e., stopped before they either found or proved the optimal solution, this approach did not necessarily yield a feasible solution. A gap therefore existed for general purpose methods or frameworks, that could provide support to develop *heuristic* optimization methods to solve problems for which no specific heuristic existed.

In the last decades, algorithmic advances as well as hardware and software improvements have provided an excellent environment on which to build such general-purpose frameworks. From the 80s on, several frameworks were proposed that could be used to develop effective heuristic algorithms for a wide range of different optimization problems. These frameworks were sometimes called *modern heuristics*, and are now generally known as *metaheuristics*.

The term metaheuristic was coined by Fred Glover in his seminal article "Future Paths for Integer Programming and Links to Artificial Intelligence" (Glover, 1986). A metaheuristic can be seen as a methodology that includes master strategies capable of guiding the search for the globally optimal solution. They are considered more complex and efficient than simple heuristic algorithms because they explore areas in the solution space that go beyond those explored by the simple heuristics, which tend to focus on finding a single locally optimal solution.

In this paper, we will adopt the definition of Sörensen and Glover (2013): "*A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms*". In this definition, a metaheuristic is itself not an algorithm (i.e., a precisely defined series of steps), but plays a looser role as a more or less consistent set of high-level ideas that can be used to develop a problem specific heuristic optimization algorithm. This means that the designer of a heuristic based on a metaheuristic framework still has a large amount of freedom in choosing the specific characteristics of their method, and that the degree of "engineering" required to instantiate a metaheuristic framework so it can solve a specific optimization problem, can be considerable.

Surveying the literature, one could be excused for thinking that there exist dozens, if not hundreds or even thousands of different metaheuristics. In the following sections, we classify those metaheuristics in order to create some clarity, according to the way in which the metaheuristic proposes to manipulate the solutions. *Constructive* heuristics, described in Section 2 build solutions from their constituting elements. *Local search* heuristics perform small changes to a single solution iteratively to improve it as much as possible. Section 3 covers these important methods that are at the core of many metaheuristics. Finally, *Population-based* heuristics, described in 4, combine solutions into new ones. We consider these categories to organize this paper, since they provide a natural way to approach heuristics both from a theoretical and a historical perspective. We do not try to cover all existing metaheuristic methodologies, but we limit ourselves to those that changed our way of thinking in problem solving, and in that sense may be considered the most relevant ones.

During the last few decades an increasing number of "novel" metaheuristics have been proposed based on a metaphor of some natural or man-made process that is often unrelated to optimization. The range of sources from which inspiration has been drawn is simply baffling (Sörensen, 2015). From African buffaloes to zombies, and from black holes to bats, frameworks have been proposed whose novelty seems to lie in the description of the source process that inspired them. The arguments against the proliferation of these "novel" techniques have been well documented (see e.g., Camacho-Villalón, Dorigo, and Stützle (2023) for a scathing argument against several "well-known" metaphor based metaheuristics). We do not go in depth on this issue, but just note that there is usually nothing "novel" about these proposals (but that the use of metaphor-specific terminology obfuscates this fact), and that even the metaphor itself more often than not does not make sense. In summary: metaphor-based metaheuristics are simply bad science, and it is good to see that a large majority of reputable OR journals do not publish them anymore, and that several journals even have an explicit policy against them (Aranha et al., 2021). In the remainder of this paper, we therefore ignore these "contributions".

After the revision in Sections 2, 3, and 4 of the three heuristic categories described above (constructive, local search, and population-based respectively), we report on the history of heuristics. Section 5 provides a critical review of the main historical developments and their impact on problem solving, and Section 6 focuses on the contributions of the European Journal of Operational Research to the metaheuristic field. Section 7 presents the ecosystem around the metaheuristic community. What is expected in the future with the next generation of metaheuristics, including for example meta-analysis or hybridizations with machine learning is introduced in Section 8. The paper concludes summarizing this critical review in Section 9.

2. Constructive metaheuristics

As the name suggests, heuristics based on a constructive metaheuristic *construct* solutions from their constituting elements. These *elements* depend on the model that is being solved. Examples include: the items in a knapsack problem, the arcs between nodes in a routing problem, the order of the tasks in a scheduling problem, etc. Generally, the constructive process starts from an empty solution, i.e., a solution in which the status of each solution element, either part of the solution or not, is undefined.

The construction process selects one element at a time from the list of elements whose status (included in the solution or not) is still undefined. This distinguishes them from local search heuristics, where the status of each element is known at each step of the search process. Generally one can order the potential elements in order of some measure of desirability. In a TSP, e.g., the potential arcs of the underlying graph can be sorted in order of increasing distance. Items in the knapsack problem can be ordered e.g., by increasing profit, decreasing weight, or a combination of both. The element list is not necessarily static and its ordering can change as the constructive process continues. For example, in a nearest neighbor heuristic for the TSP the arcs are sorted by increasing length, but arcs that do not depart from the current node do not appear on the element list.

By adding one element at a time, the solution iteratively becomes more complete. When the status of each potential element is determined, the process finishes and a complete solution has been generated. Due to its nature, the constructive process only has a complete solution at the end. Determining the value of the objective function is therefore only possible after the construction process finishes. The same holds true for determining the solution feasibility, i.e., whether all constraints are satisfied, although it is often possible to ensure that the construction process automatically leads to a feasible solution. E.g., a nearest-neighbor heuristic for the VRP can ensure that the solution remains feasible during the construction process by returning to the depot once the capacity of the vehicle has been reached.

When the construction process always selects the "best" element, i.e., the most desirable element on the element list, the heuristic is called *greedy*. Greedy heuristics are both very natural to a human problem-solver and very common in practice. For some optimization problems, like the minimum spanning tree problem, they result in optimal solutions. They do come with some drawbacks, however. First, always choosing the "most desirable" element is not guaranteed to result in a good solution, for which reasons, greedy heuristics have been called *myopic*. Second, greedy heuristics are typically deterministic and generate the same solution every time they are executed. They can therefore not be simply repeated in the hopes of finding better solutions.

For these reasons, several possible strategies have been proposed to improve the performance of a greedy heuristic. We will briefly discuss the most important of them here. For each of these strategies, many different variants have been proposed, but an exhaustive overview of those is well beyond the scope of this paper.

2.1. GRASP

GRASP (Feo & Resende, 1995; Resende & Ribeiro, 2016) adds randomness to the greedy selection process by not selecting the most desirable element at each step, but rather selecting one element randomly from a *restricted candidate list*. The size of the restricted candidate list determines the balance of greediness versus randomness of the heuristic (a larger restricted candidate list means the algorithms behaves in a more greedy way). Typically, the constructive phase of the search is followed by a local search phase in which the solution is improved. Many variations on the idea of GRASP have been proposed, such as *reactive* GRASP (Prais & Ribeiro, 2000) in which the size of the restricted candidate list is dynamically varied.

2.2. The pilot method

The pilot method (Duin & Voß, 1999) uses a *look-ahead strategy* in which the selection of the element from the list does not depend on that element's raw "value", but on its potential to find a good solution when selected next. Like most constructive metaheuristics, the pilot method assumes that a fast greedy heuristic is available. At each iteration, the method determines a *pilot* for each possible element that can be selected next, equal to the objective function value of the solution that would be obtained by the greedy heuristic under the assumption that the element is selected next. The element selected next is the element with the best pilot. Of course, several variations on this theme are possible.

2.3. Ant colony optimization

Ant colony optimization (Dorigo, Birattari, & Stutzle, 2006) uses a parallel set of independent construction processes (called "ants") and base the construction on a combination of randomness and information gathered by the "ants" on the desirability of each element. The constructive processes are then allowed to update the desirability of each element, based on the quality of the solutions that they have produced. A "pheromone evaporation" mechanism is also in place to increase the influence of more recently constructed solutions.

There is a interesting connection here between ant colony optimization and tabu search. Note that several of the principal methods to initiate tabu search make use of constructive search. For example, the initial phase of the probabilistic tabu search (Glover, 1989) employs a candidate list and selects moves probabilistically based on their objective function evaluations, accounting for sequential and cumulative differences in the evaluations. Probabilities are increased for moves that appear longer or more frequently on the candidate list. After a first constructive pass, subsequent constructions introduce diversification criteria as part of the evaluation process.

2.4. (Adaptive) Large neighborhood search

Large neighborhood search (Shaw, 1998) alternates a constructive heuristic with a destructive heuristic. The latter partially destroys the solution, so that it can be rebuilt by the constructive heuristic, and is usually severely randomized so as to introduce sufficient diversification in the construction process.

Adaptive large neighborhood search (Ropke & Pisinger, 2006) uses a *set* of constructive (repair) operators and a *set* of destroy operators and adapts the probability of these heuristics being selected to their performance in previous iterations.

Constructive and destructive heuristics have received a lot of attention and their alternation has been applied within different metaheuristics, as in the iterated greedy approach of Ruiz and Stützle (2007). A more systematic alternation of constructive and destructive phases was initiated in the context of tabu search in references such as Glover and Laguna (1993) and has been a key accompaniment of many tabu search approaches. Strategic oscillation operates not only with constructive and destructive moves, but more generally is defined in terms of approaching, potentially crossing, and receding from a boundary determined by feasibility or structure, or alternatively by moving toward or away from a region where the search appears to gravitate.

3. Local search

Local search methods were very popular in non-linear (continuous) optimization in the eighties. We may find many papers applying multi-start methods to find a global solution by starting a local solver from multiple starting points in the solution space S. The most basic multi-start method generates uniformly distributed points in S, applying the local solver from each of them. In the case of optimizing differentiable functions, the solver is usually based on the gradient vector. In any case, local search basically produces a sequence of solutions converging to a local optimum. From a theoretical point of view, this process converges to a global solution with probability one as the number of initial points approaches infinity (Solis & Wets, 1981). In practical terms, these multi-start procedures are heuristics that produce a good local optimum (see for example the *multi-level single linkage* by Rinnooy Kan and Timmer (1987)).

When considering a combinatorial optimization problem in which the solution space is usually defined in terms of integer variables, we cannot directly apply the concept of gradient that requires a continuous space. But we can adapt it. As a matter of fact, the adaptation of the local search to the integer domain has probably been the most powerful heuristic tool to solve combinatorial optimization problems.

Local search for combinatorial optimization starts from an initial solution S, obtained either randomly or with the application of a constructive method (as shown in Section 2), and explores the solutions obtained when applying a small change, called *move*, to S. The neighborhood $N(S)$ contains all the solutions that are obtained applying a move to S. In a first-improving variant, local search resets S to be the first improving solution S' found in $N(S)$, i.e. $S \leftarrow S'$ and local is restarted at S. In a best-improving variant, local search resets S to be the best improving solution found in $N(S)$ and restarts. If no improving solution is found in $N(S)$, then we say that S is a locally optimal solution and the local search halts.

By applying local search from different starting solutions S, a variety of locally optimal solutions may be found. Embedding local search within a multi-start procedure where each local search starts from a different starting solution will produce a set of locally optimal solutions, the best of which could perhaps be a global optimum. Algorithm 1 shows a pseudo-code of a standard local search algorithm for a minimization problem.

Algorithm 1: Local search algorithm

Input : Current solution S
Result: Improved (local optimal) solution S'
Generate the neighborhood $N(S)$
Improve ← **true**
while *Improve* **do**
  Identify the best solution $S' \in N(S)$
  if $f(S') < f(S)$ **then**
    $S \leftarrow S'$
  else
    Improve ← **false**
  Generate the neighborhood $N(S)$

The effectiveness of local search depends on several factors, such as the neighborhood structure, the function to be minimized, and the starting solution. Move definition plays a central role since the neighborhood contains the solutions generated by applying the move. For example, in problems where solutions are represented as permutations, such as the well-known TSP, insertions are probably the most direct and efficient way to modify a solution. Note that other movements, such as swaps, can be obtained by composition of two or more insertions. It is clear that the cardinality of the neighborhood depends on the move definition, and it is difficult to determine before-hand which move would produce better local optimal solutions. As it is customary in heuristic optimization, to disclose the best strategy, we usually have to resort to experimentation.

It is difficult to identify the first paper proposing a local search method, but we can place the first ones in the late fifties. In particular, Croes (1958) proposed a method for solving the traveling salesman problem based on first computing a trial solution, and then iteratively improving it by performing what the author called *inversions*. In this seminal paper, there is no explicit reference to local search, move or neighborhood, to mention terms common nowadays. Using the standard terminology, we would say that the method first constructs a greedy solution and then applies a local search based on a move that inverts a partial sequence of the permutation that represents the current solution. Special attention is given to the incremental computation of the objective function, what now we call the *move value*. The paper ends with some final remarks comparing the performance of the method with the previous one by Dantzig, Fulkerson, and Johnson (1954) on a 20×20 matrix and with some tips for its implementation both by hand and in a mechanized way, which illustrates how much the field has improved considering the current perspective.

Local search was originally based on a deterministic conception in which the best way to optimize is to improve as much as possible a given solution in a short term horizon. It computes a greedy evaluation function reflecting the objective function improvement, to select the best solution in the neighborhood of the current solution. This simple mechanism, although relatively effective, turned out to be limited to obtain high-quality solutions for large instances of difficult problems. Researchers in many different fields of operations research, such as routing, scheduling or graph theory, proposed different search strategies in the seventies and eighties to overcome the limitations of local search. They have been coined as *Stochastic Local Search*, or more recently as *Metaheuristics*.

The contrast between the meta-heuristic conception and the local search conception is significant. For many years, the primary objective of a heuristic procedure (a conception that in some ways it is still prevalent today) was to envision an iterative rule that terminates as soon as no solutions immediately accessible could improve the last one found. 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. Metaheuristics in their modern forms are based on different interpretations of what constitutes "intelligent" search that goes beyond the application of stochastic elements to a local search.

Randomization usually plays an important role in stochastic local search methods, either to generate starting solutions or to improve them in a non-deterministic way. However, deterministic strategies are also the foundations of important methods that transformed our way of thinking on problem solving. We now briefly describe tabu search and variable neighborhood search to illustrate very effective strategies that lead to two of the most important local search based methodologies: the use of memory and the systematic change in the move definition.

3.1. Iterated local search

One popular metaheuristic based on local search is known as *iterated* local search (ILS). In its basic form, the idea of ILS is to iteratively restart the local search operator from a perturbed solution. The perturbation operator partially scrambles the last local optimum found, and then starts the local search operator from this scrambled solution. The goal of the perturbation operator is to change the current solution enough to end up in a different *basin of attraction*, i.e., a solution that will lead to a different local optimum. ILS can be considered as a random walk over local optima. An early example of ILS in EJOR can be found in Lourenço (1995).

3.2. Tabu search

Tabu search (Glover, 1986) begins in the same way as ordinary local or neighborhood search, proceeding iteratively from one solution S to another one until a termination criterion is met. We may contrast tabu search (TS) with a simple descent method that only permits moves to neighbor solutions that improve the current objective function value, and ends when no improving solutions can be found. TS permits moves that deteriorate the objective function value of the current solution, but the moves are chosen from a modified neighborhood, $N^*(S)$, which is the result of keeping track information during the search. In the TS strategies based on short term considerations, $N^*(S)$ is usually a subset of $N(S)$, and the tabu classification serves to exclude some of its elements.

TS usually applies attributive memory for guiding purposes (i.e., to compute $N^*(S)$). Instead of recording full solutions, attributive memory structures are based on recording attributes. This type of memory records information about solution properties (attributes) that change in moving from one solution to another. The most common attributive memory approaches are recency-based memory and frequency-based memory. Recency, as its name suggests, keeps track of solutions attributes that have changed during the recent past. Frequency typically consists of ratios about the number of iterations a certain attribute has changed or not (depending whether it is a transition or a residence frequency). The interplay (usually alternation) between recency and frequency memories lead to intensification and diversification strategies that permit to explore the search space in an efficient way.

3.3. Variable neighborhood search

The basic strategy in variable neighborhood search (Mladenović & Hansen, 1997) is to systematically change the neighborhood of the solution. This strategy is applied in the context of a standard descent phase to find a local optimum and coupled with a perturbation phase to get out of its basin of attraction.

Variable neighborhood search (VNS) is based on three well-known principles:

- A local minimum with respect to a neighborhood is not necessarily so for another one.
- A global minimum is a local minimum with respect to all possible neighborhoods.
- Local minima with respect to one or several neighborhoods are relatively close to each other for many problems.

Note that while the first two principles are theoretical, the third one is more an observation, that may or may not hold, and has been observed in different problems. The methodology exploits these three principles combining deterministic and stochastic changes of neighborhoods. The deterministic part is provided by a local search heuristic while the stochastic comes from a random perturbation called shaking.

4. Population-based metaheuristics

Constructive methods and local search algorithms, as described in the previous sections, may build and modify many solutions during the search process. But they have one big characteristic in common, they only work on one solution at a time.

Population-based metaheuristics are a class of optimization algorithms that draw inspiration from natural processes to solve complex problems. They are designed to explore large solution spaces efficiently by maintaining a population of candidate solutions and iteratively improving them over multiple generations. These algorithms emulate the Darwinian principles of evolution to search for high-quality solutions, and fall under the banner of Evolutionary Algorithms (EA). More specifically, the first known population-based methods are genetic algorithms from Holland (1975a) and made more usable by Goldberg (1989).

At the core of population-based metaheuristics is the concept of a population, which consists of multiple solutions (called individuals). The population evolves over time through a series of operations, such as selection, reproduction, and mutation, which mimic the natural processes of genetic variation and inheritance. A simple framework of EA is presented by Taillard (2023) and is reproduced here in Algorithm 2. Each of the operations has a different contribution in the search for diversification or intensification.

The main idea behind population-based metaheuristics is to encourage exploration and exploitation of the search space. Exploration involves searching a wide range of solutions to avoid getting trapped in local optima, while exploitation focuses on intensively searching the vicinity of promising solutions to refine and improve them. As in every metaheuristic, the balance between exploration and exploitation is crucial to achieve a good trade-off between convergence to high-quality solutions and maintaining diversity within the population.

Population-based metaheuristics often incorporate mechanisms for evaluating and comparing the objective function value (called fitness) of candidate solutions. Fitness evaluation determines the quality of a solution and is used to guide the search process. Based on the fitness values, individuals are selected for reproduction, and their genetic material (the core components of the solution) is combined through recombination or crossover to create new offspring. Mutation operators introduce small random changes to the offspring, introducing exploration and diversification into the population.

Algorithm 2: Framework of evolutionary algorithms

```
Input : Parameters μ and λ, selection for reproduction,
        crossover, mutation and selection for survival operators
Result: Population of solutions P
Generate a population P of μ solutions
repeat
    Select individuals from P with the selection for
      reproduction operator
    Combine the selected individuals with the crossover
      operator and apply the mutation operator to get λ new
      solutions
    Among the μ + λ solutions, select μ individuals with the
      selection for survival operator; these individuals constitute
      the population P for the next generation
until a stopping criterion is satisfied
```

The population evolves iteratively, with each iteration representing a generation. The selection process, based on fitness, determines which individuals survive to the next generation, while the reproduction and mutation operators create new individuals. Over time, the population tends to converge towards better solutions as the search progresses.

Genetic algorithms, considered to be the original EAs were deceptive because of premature convergence (all individuals are clones) and more important they may miss the optimal solution. Other population methods have been established to overcome there difficulties. Some popular population-based metaheuristics include memetic algorithms (MA), biased random key genetic algorithms (BRKGA), scatter search (SS), and path relinking (PR). Each of these algorithms has its unique characteristics, but they all share the fundamental principles of maintaining a population of candidate solutions and iteratively improving them through inspired search and interaction mechanisms.

An almost endless list of bio/nature-inspired metaheuristics has emerged in recent years. Most of these methods are not bringing any insight from the research point of view. This article is not the place to debate again the pros and cons of such methods. The interested reader should consult for example these two Aranha et al. (2021), Sörensen (2015) to have a better understanding of the situation.

4.1. Memetic algorithms

The general idea behind memetic algorithms is to exploit all possible knowledge of the problem being solved inside the solution process. This is also where the name "memetic" takes its roots. The knowledge can take different forms, but always in the goal of favoring the balance between exploration and exploitation. These mechanisms are designed to overcome the difficulties encountered by traditional genetic algorithms.

Moscato (1989) designed the memetic algorithm as a population method where a local search operator is applied to each offspring generated. It would be oversimplifying to state that memetic algorithms are simply adding a local search operator to a population method. There is a lot more behind MAs and this is testified by the success of the many applications that have been published since the 90's. In the case of MAs, the local search operator is clearly a mean for intensification (getting closer to the optimal solution) whereas the crossover operator, initially designed in GAs for intensification, could also play a role as a diversification operator.

It is obvious that a memetic algorithm is at least as good as its genetic algorithm counterpart and is also as good as the local search operator. The success of the MA design for a specific problem comes from the right combination of its components.

4.2. Biased random-key genetic algorithms

Genetic algorithms with random keys (Gonçalves & Resende, 2011), or random-key genetic algorithms (RKGA), were introduced in the nineties for solving sequencing problems. In this method, chromosomes are represented as vectors of randomly-generated real numbers in the interval [0, 1]. RKGA rely on a decoder, which is basically a deterministic algorithm that transforms a chromosome into a solution of the combinatorial optimization problem at hand. In this way, the classical evolution of a genetic algorithm, takes place here in the vectors with the keys that code the solutions.

The search is initiated when each component of the solution vector, or random key, is randomly generated in the real interval [0, 1]. Then, after the fitness of each individual is computed by the decoder, the standard GAs operators are applied to the random keys. In short, the simulated evolution of the algorithm takes place on the keys instead of on the solution themselves.

As described in Gonçalves and Resende (2011), a biased random-key genetic algorithm (BRKGA) differs from an RKGA in the parents selection for combination. Specifically, in the original RKGA both parents are selected completely at random, but in BRKGA one of them is selected at random from a restricted set only containing the best individuals in the population (and the other one from the rest of the population). This biases the search towards better regions of the solution space. An application to job-shop scheduling can be found in Gonçalves, de Magalhães Mendes, and Resende (2005).

4.3. Scatter search

Scatter search (Glover, 1998b) is a population-based metaheuristic used to solve complex combinatorial and continuous optimization problems. Compared to classical population methods (like genetic algorithms), it maintains diversity in the population through a small set of candidate solutions called the reference set, in which solutions are all combined together in an extensive way. It standard implementation (Martí, Laguna, & Glover, 2006) follows the "five-method template" according to:

- A Diversification Generation Method to generate a collection of diverse trial solutions.
- An Improvement Method to transform a trial solution into one or more enhanced trial solutions. This is typically a local search method.
- A Reference Set Update Method to build and maintain a reference set consisting of the best solutions found. Solutions gain membership to the reference set according to their quality or their diversity (in a broad meaning of best).
- A Subset Generation Method to operate on the reference set, to produce a subset of its solutions as a basis for creating combined solutions.
- A Solution Combination Method to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solution vectors.

The strength of this method relies on the reference set which is composed of local optima but all diverse. The tuning of the parameters can be tricky and the user should pay attention to the exhaustive combination part which can be time consuming. Many examples of scatter search applications can be found in Martí, Corberán, and Peiró (2015).

4.4. Path relinking

In all population metaheuristics, the path relinking (PR) method may have a particular role to play. Path relinking (Glover, Laguna, & Martí, 2000) starts with two high-quality solutions, usually one from the current search (the starting solution) and another from a reference set (the target solution). A path is constructed between these two solutions by transforming the current solution into the target solution, applying modifications on the composition of the solution itself. Path relinking can be thought of as a constrained neighborhood search, where the search is limited to explore the solutions in the neighborhood with characteristics of the guiding solution. The selected neighborhood will determine the set of solutions visited by path relinking. If the two solutions are *distant* enough, the path (and all intermediate solutions built during the construction of the path) allows the exploration of different parts of the solution space bringing an extraordinary diversity in the search. To intensify the search, the intermediate solutions along the path are improved by using a local search operator. Again, an efficient path relinking implementation will find a good balance between intensification and diversification. The importance of the local search operator is then crucial.

The current solution as well as the solutions of the reference set can be obtained by any method, including other metaheuristics. Laguna and Martí (1999) proposed to apply PR to the best solutions obtained with GRASP. This hybridization, simply known as GRASP with PR, is very popular and has lead to many successful implementations. The construction of the path is usually done in a deterministic manner, although in the last few years randomized designs have been explored as well (see for example greedy randomized path relinking). It may exist a multitude of variants to change one solution into another, letting the path relinking bring a large algorithmic flexibility based on problem characteristics. For these reasons, the path relinking can be seen as a deterministic generalization of many population metaheuristics. Resende and Ribeiro (2016) give many examples of path relinking applications.

Path relinking and scatter search are joined in a common perspective in the "template" paper of Glover (1997), which introduces multiple strategies including diversification methods that have largely been overlooked.

4.5. Combination of methods

The balance between *intensification* and *diversification* is a very sensitive issue in the conception of efficient population-based metaheuristics. The observation of the evolution from the simplest evolutionary algorithm to the more complex path relinking shows that none a single method can outperform any other and the combination of good practices in one method can bring enhancement into another one. There is no limit in the combination of these methods. But one should keep in mind that the user of these metaheuristics need to be in control and too many parameters may lead to unpredictable behaviors.

Overall, population-based metaheuristics provide a flexible framework for solving optimization problems that are difficult or intractable using traditional methods. They excel in addressing complex, multimodal, and combinatorial problems where the search space is vast, and the objective function might be non-linear, discontinuous, or noisy.

5. Historical review

It is very difficult to find the origins of an area of knowledge since in one way or another it probably has always been with us. This is especially true in the case of heuristics, which are intimately connected with the way in which human beings think. As a matter of fact, the roots of the term heuristic are established in the ancient Greek words *eurika* and *heuriskein*, which mean find or discover, and are connected with the famous quote exclaimed by Archimedes, *heúreka*, when he discovered how to measure the volume of an object. There is

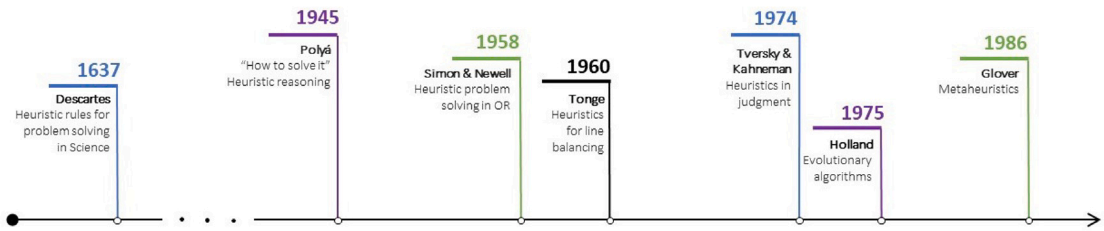

Fig. 1. Timeline of the origins of heuristics.

an agreement however that the causal explanations introduced by the French philosopher René Descartes in the 17th century established a milestone in Science, since it permitted the rational formulation and solving of problems. The Cartesian criterion of rationality is based on two assumptions, representation (algebraic symbols) and organization (mathematical systems), and Descartes formulated simple rules to guide the solver toward relevant aspects of the problem. Operations research practitioners can easily identify these assumptions as key components of thinking about an optimization problem to devise a heuristic for it.

In its origins, heuristic was the name of a field of study, that was not very clearly defined, belonging to logic and philosophy and meant to study the methods and rules of discovery and invention. Following the deductive principles of Descartes, we can find other mathematicians, such as Bernard Bolzano (1781–1848), applying general procedures mostly described in philosophical terms to solve problems. Mathematics and philosophy were very connected, if not the same, in the early times of Science. The term *heuristic reasoning* was introduced much earlier than the so-called *heuristic algorithm*, which is very popular in optimization at present. Heuristic reasoning refers to thinking strategies that allow us to make judgments or even find solutions not in a rigorous way, but quickly find the most plausible provisional solution. In an effort to trace back to the origins of the heuristic optimization, and draw a timeline of the principal moments of heuristics to our days, we traverse different areas of knowledge, from philosophy, crossing mathematics, psychology and operations research, to end in artificial intelligence of today. Fig. 1 depicts the timeline with the milestones in the origin of heuristics cited in this section.

It seems that the famous book by the Hungarian mathematician G. Polya, *How to Solve It*, published in 1945 by Princeton University Press (Polya, 1945), is the first document where we may find the use of heuristics to solve mathematical problems from a modern scientific perspective. The way in which Polya presented and categorized heuristics laid the foundations of our current approach to heuristic algorithms. On one hand, the author recommends to follow the logical principles, such as induction and analogy, and previous experience in the mental process to solve a problem. On the other hand, he states that we cannot take heuristic reasoning for proof, thus implicitly pointing out the current distinction between exact versus heuristic optimization methods. In Polya's words, "What is bad is to mix up heuristic reasoning with rigorous proof. What is worse is to sell heuristic reasoning for rigorous proof".

The connection between heuristics and operations research (OR) can be found in the article by Simon and Newell (1958) entitled *Heuristic Problem Solving: The Next Advance in Operations Research* in which the authors connected the use of heuristics with ill-structured problems, in which classic OR methods cannot be directly applied since they do not have a well-structured mathematical formulation. A search of the OR journals published in the 60s, such as *Operations Research* or *Management Science*, returns several papers with the first uses of heuristics for specific optimization problems, such as Tonge (1960) for line balancing or Karg and Thompson (1964) for the traveling salesman problem.

When examining the European context, particularly the first publications of OR journals, we can find the Operational Research Quarterly (published from 1950), and the European Journal of Operational Research (EJOR), in which heuristics have been a subject of interest since its inaugural issue. Specifically, in the first volume of the journal, published in 1977, we can find a heuristic for queueing systems, in which approximate formulae were proposed for the average times experienced by customers. Cosmetatos (1977) elaborated on Cobham's model with the assumption of exponential service times relaxed in a multi-server priority system. Comparisons between approximate and simulated results indicated a satisfactory performance of the heuristic.

Since this seminal paper, EJOR consistently published papers devoted to heuristic optimization. In the second volume, we can find heuristics for scheduling problems (Mack & Smith, 1978) where complicated sets of constraints render standard mathematical approaches intractable. This paper illustrates how list processing and problem oriented data structures were utilized to facilitate development of an effective scheduling heuristic for itinerant teachers. The techniques employed would prove to be extremely useful in solving many other scheduling problems. The role of EJOR in the development of metaheuristics has been equally important, and we discuss this role in some detail in the next section.

As described by Hjeij and Vilks (2023) in their brief history of heuristics recently published, we have to take "problem solving" in a broad sense that includes decision-making and judgements when researching what has been termed as heuristics. In line with that, we must highlight the work by Daniel Kahneman and Amos Tversky in cognitive and social psychology. In 2016, the NewYorker magazine reviewed a book on the life of these two economists (also labeled as mathematical psychologists) who changed *how people think about how people think*. Their paper "Judgement under uncertainty: heuristics and biases", published in 1974, had a huge impact on behavioral economics, crossed over from specialized literature to the general audience (Tversky & Kahneman, 1974). The authors identified three types of heuristics to process information when making judgements: availability, representativeness, and adjustment. A better understanding of these heuristics and of the biases to which they lead could improve judgments and decisions in situations of uncertainty.

Sörensen, Sevaux, and Glover (2018) in their *History of Metaheuristics* distinguished among six periods of time, starting with the pre-theoretical period (until c. 1940) in which heuristics were applied but not theoretically studied in the specific context of mathematics. This initial period was followed by the early period, starting with the publication of Polya's book commented above and ending around 1980. After this period in which many heuristics were proposed for classic optimization problems, follows the method-centric period (1980–2000) that witnessed the proposal and development of several metaheuristics such as simulated annealing, tabu search, or the popular genetic algorithms (Holland, 1975b). The term metaheuristic was introduced by Fred Glover in 1986, in the same paper that tabu search (Glover, 1986). Metaheuristics refer to problem-independent frameworks, as opposed to the previous heuristics specifically designed to solve a particular problem. Although this vision of frameworks as opposed to methods was conceived later (in the framework-centric-period after 2000), we may say that metaheuristics substantially changed the optimization area and permitted us to solve very complex problems with high-quality solutions, optimal in most cases.

In an attempt to establish the first papers proposing each metaheuristic methodology, we move back to 1966 when (Fogel, Owens, & Walsh, 1966) proposed the first evolutionary algorithms. It is worth mentioning that early approaches were not conceived for optimization, but were meant to study the mechanics of the systems they modeled. In particular, Fogel worked on the simulated evolution of finite-state machines to forecast nonstationary time series. In line with that, Holland (1975a) proposed the first genetic algorithms, although we have to wait until 1989 when (Goldberg, 1989) clearly applied them to solve combinatorial optimization problems. Holland's theory of adaptive systems was originally meant to understand complex forms of adaptation in natural systems to design adaptive artifacts.

Table 1 shows a timeline with the main methodologies where we can see in the third row corresponding to 1977 scatter search (SS). This metaheuristic also belongs to the family of evolutionary methods but from its origins has been successfully applied to hard optimization problems. SS was first introduced by Glover (1977) as a heuristic for integer programming and it was based on strategies presented at a management science and engineering management conference held in Austin, Texas in September of 1967.

Kirkpatrick, Gelatt, and Vecchi (1983) proposed simulated annealing, the first local search based metaheuristic. This method established a connection between statistical mechanics modeling the thermal equilibrium at a finite temperature and combinatorial optimization. Simulated annealing was very popular at that time and the years after. Its popularity seems to have dwindled with the advent of local search metaheuristics that rely less on randomness (e.g., tabu search and variable neighborhood search). On the other hand, this drop in popularity ostensibly only occurred in the metaheuristics community itself. In the wider world, thousands of papers that apply simulated annealing are still published every year.

In the same paper that coined the term *metaheuristic*, Glover (1986) proposed the tabu search methodology, combining principles of local search with memory structures taken from learning principles. This framework has had a tremendous impact in the field, and has dramatically changed our way of thinking about heuristic problem solving. As described above, the method was initially applied as an intelligent local search, but rapidly included other search elements such as constructive and combination methods.

In 1989 a very effective and easy-to-implement metaheuristic was proposed with the acronym GRASP (greedy randomized adaptive search procedure). Feo and Resende (1989) proposed to couple a greedy randomized construction with a local search method and applied this method to many combinatorial optimization problems. The publication was followed by a tutorial in the ORSA/TIMS meeting in Nashville in 1991 and published as Feo and Resende (1995).

A further step in local search based methods was taken by VNS. Mladenović and Hansen (1997) proposed the variable neighborhood search methodology in which several neighborhoods are combined in an efficient way. We observe nowadays an increasing application of this metaheuristic — even though this is not always recognized as such — and it is probably fair to say that many commercial and open source solvers, general-purpose or problem-specific, also employ some variant of this metaheuristic.

An analogy with the foraging behavior of ants in a colony suggested the definition of a new computational paradigm called Ant System. Dorigo, Maniezzo, and Colorni (1996) proposed this new approach to stochastic combinatorial optimization. The main characteristics of this model are positive feedback, distributed computation, and the use of a constructive greedy heuristic. In this metaheuristic, the inspiration with a natural system plays an important role. The simulation of social models is extended within the particle swarm paradigm, proposed by Kennedy and Eberhart (1995) to optimize non-linear functions. In this way, they opened a line of research based on the social metaphor that resulted in a wide range of models, that in our opinion create confusion in the field and are limited in terms of their contribution to heuristic optimization (Sörensen, 2015). As mentioned in the introduction, we do not cover in our review these nature or social inspired methods, since they are mainly devoted to explain how to adapt the concepts from the analogy to the optimization problem, more than studying the search elements and strategies that are effective for a given problem.

In the early 90s, Moscato (1993) proposed memetic algorithms as a family of metaheuristics blending several elements from evolutionary algorithms and simulated annealing: individual improvement, population cooperation, and competition, as they are present in many social/cultural systems. A memetic algorithm can be considered to be a search strategy in which a population of optimizing agents cooperate and compete. They can be considered as a bridge between metaphor methods and search-based metaheuristics.

Our last methodologies in Table 1 are hyperheuristics, proposed in 2001, adaptive large neighborhood search (ALNS), proposed in 2006, and Biased Random Key Genetic Algorithms, proposed in 2011. The concept of a hyperheuristic was introduced by Cowling et al. (2001) as an approach that operates at a higher lever of abstraction than current metaheuristics. Hyperheuristics manage the choice of which lower level heuristic method should be applied at any given time, depending upon the characteristics of the region of the solution space currently under exploration. On the other hand, ALNS is composed of a number of competing subheuristics that are used with a frequency corresponding to their historic performance. Both methodologies basically propose frameworks to manage different heuristics that are selected depending on the instance solved, which is an interesting approach that can be described as intelligent problem solving. We can consider that they operate in a higher decision level than the standard metaheuristics, but due to their complexity, and associated running times (initially longer than other simpler methods), it is still not clear nowadays if they will constitute a standard approach in the future.

From a historical perspective, we must note that while mathematicians and psychologists elaborated on the notion of heuristic reasoning during the 50s and 60s, computer scientist were implementing the first algorithms based on the seminal ideas of Alan Turing in the 40s. Operations research embraces mathematics, economics, and computer science, and it naturally merged *heuristic reasoning* and *optimization algorithms* into *heuristic algorithms*. We devote this paper to the last 50 years of metaheuristic algorithms for optimization in this exciting journey originating in the center of Europe a few centuries ago that has turned into one the most crucial technologies in the development of our modern societies.

Even though the field of metaheuristics has evolved into a thriving research area within the broader area of operations research, with dozens of journals devoting a considerable part of their pages to studies involving (meta)heuristics, it has so far resisted most attempts at formalization and theorization, unlike e.g., the field of mathematical programming. Early attempts to underpin the field with a coherent set of theorems and conjectures have so far proven futile, or — to put it mildly — not very practically useful. Some randomized algorithms, like simulated annealing, e.g., have been proven to converge under some mild assumptions, but convergence at infinity in itself is a rather trivial conclusion in this context. Note that a "random walk" through the solution space will eventually converge to the global optimum given infinite time, as complete enumeration would do, which is not particularly relevant for a time-pressed practitioner. Out of necessity, the field of metaheuristics has therefore adopted a more *empirical* approach to research. Typically, research in metaheuristics is deamed of a high-quality if it can demonstrate that it yields a "competitive" heuristic, i.e., a heuristic that achieves results that are at least comparable to the state-of-the-art. Not all authors agree that this approach necessarily yields the most insights, and several have proposed better ways to turn the field of metaheuristics into a more scientifically underpinned field. We will discuss some of these ideas later.

Table 1
Timeline of the main metaheuristic proposals.

Year	Methodology	Citation
1966	Evolutionary Algorithms	Fogel et al. (1966)
1975	Genetic Algorithms	Holland (1975a)
1977	Scatter Search	Glover (1977)
1983	Simulated Annealing	Kirkpatrick et al. (1983)
1986	Tabu Search	Glover (1986)
1989	Genetic Algorithms (for opt.)	Goldberg (1989)
1989	GRASP	Feo and Resende (1995)
1993	Memetic Algorithms	Moscato (1993)
1995	Particle Swarm	Kennedy and Eberhart (1995)
1995	Iterated Local Search	Lourenço (1995)
1996	Ant Colony	Dorigo et al. (1996)
1997	Variable Neighbor. Search	Mladenović and Hansen (1997)
1998	Large Neighborhood Search	Shaw (1998)
2001	Hyper-heuristics	Cowling, Kendall, and Soubeiga (2001)
2006	Adaptive Large Neighborhood Search	Ropke and Pisinger (2006)
2011	Biased Random Key Genetic Algorithms	Gonçalves and Resende (2011)

6. The role of EJOR in the metaheuristics literature

EJOR can undoubtedly be described as a *flagship* journal of the operations research community. As a general OR journal, EJOR obviously has its biases (e.g., it has probably published more research on multi-criteria decision making than the average OR journal). Nevertheless, EJOR can be considered a cross-section of the operations research literature and therefore a good barometer for the current trends in the field, as well as many of its sub-domains, including the domain of (meta)heuristics.

In this section, we intend to survey the field of metaheuristics as viewed from the perspective of EJOR. Our methodology is simple: we go through all issues of EJOR since its inception in 1977 and include those articles that discuss metaheuristics. We focus mostly on *meta*heuristics and skip papers that simply develop a heuristic for a specific problem. In the beginning, we can include almost every paper published. From the 80s on we have to be more selective, as research on metaheuristics truly takes off. From around 2005, the number of papers on metaheuristics explodes, and we have to skip most of them.

Even though EJOR has definitely played a role in the development of the field of metaheuristics, and has published many papers that develop and use heuristic methods to solve various optimization problems, the European Journal on Operational Research remains mostly absent from the discussion on and the development of the field of metaheuristics in the early days. None of the important early techniques such as tabu search (TS), genetic algorithms (GA), and simulated annealing (SA) were initially introduced in EJOR. This suggests that the journal, although comprehensive in scope, has not been a primary vehicle for groundbreaking innovations in the specialized domain of metaheuristics.

Evidently, EJOR's importance in the field of metaheuristics changes rather dramatically from the early 2000s. From then on, and especially after the EURO Winter Institute on Metaheuristics and the special issue of EJOR that arises from it, both the number of articles and their importance increases. The "new wave" of metaheuristics that arises in these years (guided local search, variable neighborhood search, etc.) find a home in EJOR. Importantly, unlike other top journals in the field EJOR never succumbs to the tsunami of metaphor-based methods that have flooded the research field (a few exceptions notwithstanding).

We have mentioned before that EJOR has published papers on heuristics since its first issue in 1977. One of the first contributions that does not simply present a heuristic method in the journal is a tutorial article (Silver, Victor, Vidal, & de Werra, 1980) on heuristics. In this article, the authors provide an introduction to the various heuristic methods that exist around that time. Borrowing from Nicholson (1971), the authors define a heuristic method as a procedure "...for solving problems by an intuitive approach in which the structure of the problem can be interpreted and exploited intelligently to obtain a reasonable solution". Operationalizing the term "reasonable", the authors then develop the following desirable characteristics of a heuristic: (1) the computational effort to obtain the solution should be realistic; (2) the solution should be close to the optimum on average; (3) the probability of obtaining a poor solution should be small; (4) the heuristic should be as simple as possible. Clearly, these principles still hold for all heuristics developed today.

Even though the term *metaheuristic* does not appear in Silver et al. (1980), the authors touch upon the topic when classifying the various existing heuristics. The authors, e.g., make the distinction between *constructive methods* and *local improvement methods*, two categories that reappear in Sörensen and Glover (2013)'s overview. Some of the principles mentioned in Polya (1945) also appear here (heuristics based on induction, decomposition, feature extraction, . . .).

About a year later, Müller-Merbach (1981) classifies heuristics according to six different characteristics, including the "organization of the iteration tree", or the "determination of the sets of potential candidate solutions" and the "selection of the candidates". Again, the paper does not mention the term *metaheuristic*, but comes very close to developing a general classification of these methods. After all, a metaheuristic can be seen as a formalization of one or more specific characteristics of a set of heuristics. In some sense, metaheuristics can be claimed to reverse the process: rather than organizing existing heuristic methods according to a set of characteristics, metaheuristics determine a set of characteristics that define the heuristics developed according to its framework.

In the 1980s we notice a steady increase in the number of articles that describe heuristic algorithms and also explicitly (i.e., in the title) recognize this. Papers on heuristic algorithms are still the exception in EJOR, but their importance is clearly increasing. *Meta*heuristics, however, are still not on the radar. Of course, the term would only be coined in 1986 (Glover, 1986).

Probably the first paper published in EJOR that studies a *meta*heuristic is Burkard and Rendl (1984). In this paper, the authors apply the procedure of Kirkpatrick et al. (1983), that had appeared a year earlier in the journal Science, to the quadratic assignment problem (QAP), and find that it produces excellent results. The "thermodynamically motivated simulation procedure for combinatorial optimization problems" mentioned in the title, is — of course — *simulated annealing*, one of the earliest metaheuristics.

Near the end of the 1980s, the field of OR is ostensibly in crisis. Ackoff (1979) publishes a paper in which the relevance of the entire field of operations research is called into question. More specifically, there was a growing concern that OR had become too theoretical and mathematical, with a focus on complex models and methods that were difficult to apply in real-world situations. This led to a perception that OR was losing its relevance to business and industry, where practical, implementable solutions were needed. This perceived crisis had its repercussions, e.g., in OR education a sense grew that it was not adequately preparing students for the practical challenges they would face in industry. The curriculum was often criticized for being too theoretical and not sufficiently focused on practical problem-solving skills. As OR struggled to define its unique value proposition in a rapidly changing world, there was an identity crisis within the community, and many people questioned whether OR should continue to emphasize its mathematical and theoretical aspects or pivot more towards practical applications and consultancy.

In EJOR, Hansen (1989) discusses the OR crisis, essentially defending many novel innovations of the field and refuting the claims that OR is in a true crisis. The author does mention that innovations in integer programming have allowed researchers and practitioners to express all optimization problems in mathematical notation, but agrees this does not mean that they can also solve them. General-purpose tools that could truly help to solve all optimization problems, no matter how large or complex, would only appear with metaheuristics.

It is therefore probably not a coincidence that the increase in the number of (meta)heuristic optimization techniques coincides with the OR crisis, as many researchers start to see the benefits of solving optimization problems without clinging to the guarantee of optimality. The end of the 1980s is also when the first metaheuristics start to be mentioned in EJOR papers.

Glover and Greenberg (1989) publish a survey on different types of heuristic search methods, exploring several frameworks that would later be known as metaheuristics, like simulated annealing, and neural networks. Two metaheuristics make their debut appearance in this article: genetic algorithms, and tabu search. The authors mention a fifth metaheuristic, called Target Analysis. This metaheuristic is described as "[. . .] an integration of artificial intelligence with operations research that gives a new strategy for solving combinatorial optimization problems. One may use any conventional strategy, such as implicit enumeration, and subordinate its control parameters to a learning model patterned after classification problems". Target Analysis seems to be an early forerunner of the methods integrating machine learning models and heuristics, that are currently being studied.

In the same year, the first application of tabu search in EJOR appears in Widmer and Hertz (1989). In this paper the authors develop a heuristic method for the flow shop sequencing problem, relying on a heuristic developed according to the framework of *taboo* search. The alternative spelling does not have a long lifetime.

To end the 1980s, a survey on heuristics (but not on metaheuristics) appears in one of the final issues of the decade. Zanakis, Evans, and Vazacopoulos (1989) categorize a large number of heuristics found in the literature in a number of categories: construction, improvement, mathematical programming, decomposition, partitioning, solution space restriction, relaxation. All these ideas will later be generalized and developed into actual metaheuristic frameworks.

In 1990, Eglese (1990) publishes a paper on simulated annealing in which the design choices are mentioned (such as the cooling scheme), as well as some of the theoretical results that have been obtained (such as convergence at infinity, but also exponential running times to converge).

In this period, neural networks are also considered a metaheuristics, even though it is clear at this point that this approach is more suitable for pattern recognition tasks. In EJOR, Másson and Wang (1990) discuss different types of artificial neural networks, including some applications.

Papers on algorithms that are explicitly based on a metaheuristic are still few and far between in EJOR in this period, even though their numbers are increasing. Two metaheuristics are at the forefront: tabu search, and simulated annealing.

Taillard (1990) describes several algorithms for the flow shop sequencing problem, including the then new algorithm (which he also calls *taboo* search). Interestingly, the author also shows that the algorithm can be easily and successfully parallelized. Hertz (1991) introduce a tabu search heuristic for large scale timetabling problems (switching back to the now more common spelling of the framework).

Simulated annealing is also rather popular. Jørgensen, Thomsen, and Vidal (1992) presents a simulated annealing algorithm for the (real-life) afforestation problem. Kouvelis, Chiang, and Fitzsimmons (1992) introduce simulated annealing for machine layout problems in the presence of zoning constraints, and Sofianopoulou (1992) present SA for the process allocation problem (allocating communication processes to a network of processors), and Jeffcoat and Bulfin (1993) present a heuristic based on SA for resource constrained scheduling.

We also start to see papers in this period in which authors do not simply present a metaheuristic, but (also) study its properties. Adenso-Díaz (1992) investigate a feature of tabu search, i.e., the size of the neighborhoods. He introduces a *restrictive* neighborhood based on the observation that (1) in early stages of the search, jobs are exchanged that are far away from each other in the solution, whereas (2) in later stages of the search only jobs that are close to each other (or even adjacent) are swapped. The number of solutions checked (i.e., the size of the neighborhood) is decreased while the search progresses. Dowsland (1993) presents some experiments on SA for packing problems, while Laursen (1993) experiments with SA on the QAP.

Probably the first evolutionary algorithm to be published in EJOR is due to Tam (1992), who develop a genetic algorithm for facility layout design.

A bit later, Skorin-Kapov and Skorin-Kapov (1994) presents a tabu search heuristic for the location of interacting facilities, whereas Chen and Srivastava (1994) discuss an application of simulated annealing for forming machine cells in group technology.

Maniezzo, Dorigo, and Colorni (1995) make a comparison of eight "evolutionary" heuristic algorithms (these algorithms include tabu search, simulated annealing and multi-start local search, frameworks which we would no longer call "evolutionary" now) applied to the Quadratic Assignment Problem (QAP) using a developed software system called Algodesk. The focus of the study is not to determine the best result achievable (since that data is already available) but to assess the efficiency of the algorithms in producing good solutions within a 1-hour timeframe on identical IBM-PC machines. Key findings are (1) Multigreedy approaches (we would call those *multi-start local search* now), particularly those using local search operators, are very effective for solving QAP. (2) Single-solution approaches are generally more effective than population-based approaches, especially when the latter are run on single-processor hardware and not coupled with local search operators. (3) The study suggests that current communication operators (i.e., the "crossover operators") in population-based heuristics may not be worth their computational cost. Finally, (4) Boltzmann machines (a variant of simulated annealing) are found to be inefficient for searches of limited duration.

Laguna, Kelly, González-Velarde, and Glover (1995) introduce a tabu search heuristic for the multilevel generalized assignment problem. This paper also sees the first appearance of ejection chains in EJOR. Ejection chains are sequences of moves where each move "ejects" or displaces an element from its current position and possibly replaces it with another. This creates a chain of changes across the solution space.

The 1995 special issue on the 10th EURO Summer Institute, focused on Combinatorial Optimization edited by C. Roucairol, H. Thiriez, J. Krarup, G. Plateau, P. Tolla features a single paper that mentions a metaheuristic in the title, Rego and Roucairol (1995) entitled "Using Tabu search for solving a dynamic multi-terminal truck dispatching problem". Other papers also use metaheuristics, but do not mention them in the title.

In this special issue, an article (Bjorndal et al., 1995) is devoted to the participants offering their views on combinatorial optimization. Their views on the role of metaheuristics are interesting:

> The observation that many heuristics have a similar structure has led to the recent development of general meta-heuristics, such as Simulated Annealing, Tabu Search, and Genetic Algorithms. These meta-heuristics give algorithms with an essentially user-definable

complexity, since the user has great flexibility in deciding whether to trade solution quality for speed. Since these techniques are based on local search, they frequently do not require much problem-specific knowledge in order to generate good solutions.

We believe that this is an exciting time to be in the field of combinatorial optimization, since it seems likely that new and more powerful meta-heuristics will emerge in the coming years. In particular, we expect that meta-heuristics will become more widely available to practitioners through commercial software packages and libraries, in the same way as has happened for the meta-heuristic technique of constraint-based reasoning.

Meta-heuristics also present a formidable theoretical challenge to the mathematical community, since despite very promising results from experimentation and practice, there are few papers addressing the scientific reasons why these techniques should indeed be effective. It seems that the use of probabilistic techniques might provide interesting results here, and we believe that significant strides will be made in this area in the next few years.

A few new metaheuristics or variants of existing metaheuristics have been proposed in EJOR, some of which did not gain traction, such as Righini (1995).

From 1996, we find combinations of metaheuristics, such as Bölte and Thonemann (1996), who use genetic programming to automatically set the parameter values of a simulated annealing algorithm. This type of parameter tuning algorithms can be seen as a precursor of the state-of-the-art methods like irace. Adenso-Díaz (1992) use simulated annealing to allow tabu search to be started from a better initial solution in a SA/TS mixture algorithm for the scheduling tardiness problem.

Pirlot (1996) presents a tutorial on the three most widely used metaheuristics. Simulated annealing, tabu search, and genetic algorithms. Each method is described in detail and illustrated with an example application. Additionally, the paper offers a preliminary assessment and comparison of these methods from a practical standpoint. This paper contains some very interesting conclusions and insights, that still drive many of the overarching research questions of metaheuristics today, almost two decades later.

- Comparing heuristics is difficult. Comparing heuristics involves evaluating multiple factors such as ease of implementation, robustness, flexibility, computational burden, and solution quality. Moreover, defining "solution quality" is challenging (e.g., do we mean "average quality" or "worst-case quality"?). The author also highlights that heuristics can be sensitive to initial conditions and parameter settings, and their performance can be highly variable. Due to the non-standardized nature of heuristic algorithms, direct comparisons can be misleading. One should test multiple implementations and remain cautious in drawing conclusions. An enlightening quote, attributed to Johnson, Aragon, McGeoch, and Schevon (1989), is that "Although experiments are capable of demonstrating that the approach performs well, it is impossible for them to prove that it performs poorly. Defenders of SA can always say that we made the wrong implementation choices".
- Metaheuristics (the author calls them "general heuristics") like Tabu Search (TS), Simulated Annealing (SA), and Genetic Algorithms (GAs) remain popular due to their analogy with natural processes and theoretical convergence. However, the author suggests caution in relying too heavily on these features.
- There is a lot of experimentation in the field, with researchers combining classical heuristics or developing new heuristic search ideas. However, the author emphasizes the need for serious experimentation.
- TS is considered a foundational toolbox for heuristic search, incorporating features like flexible memory structures. Short-term memory effects are implemented via tabu lists, while long-term memory guides the search through intensification and diversification phases.
- Hybridization: The author notices that SA algorithms can be enhanced by incorporating TS methods, such as altering the cooling schedule or using "strategic oscillation" principles. The Genetic Algorithm can be blended with TS, employing crossover operators during collective search phases. Traditional techniques like Linear Programming can also be combined with general heuristics.
- At this time, the author was surprised that few new general heuristic methods were emerging, but some notable ones in this period included the "Great Deluge","Record-to-Record", and "Ant Algorithm". Another approach called "Noising" adds random noise to the objective function, proving to be efficient and robust in early applications. The author foresees the field evolving by building toolboxes and libraries of well-tried heuristic search methods, and suggests that the design of a heuristic can be seen as an art due to the almost unlimited combinations possible.

Around this period, i.e., the late 1990s, the number of papers on metaheuristics (still including neural networks) in EJOR keeps increasing. There is even a special issue on Neural Networks and Operations Research/Management Science (Sharda & Wang, 1996). The special issue on the Thirteenth EURO Summer Institute: Stochastic Optimization (Pflug & Ruszczyński, 1997), on the other hand, does not contain a single paper on metaheuristics, demonstrating that the worlds of (stochastic) mathematical programming and (meta)heuristics are still very much separated.

An early integration of AI techniques and metaheuristics can be found in Grolimund and Ganascia (1997). The paper discusses using an AI-based Case-Based Reasoning approach to automate the configuration of meta-heuristics like tabu search without user interaction. This method, which is domain-independent and uses a first-order representation language for problem modeling, aims to enhance operator selection in tabu search and is validated through experiments on facility location benchmark problems.

Near the end of the 1990s, we notice a series of new metaheuristics or combinations (hybrids) of existing metaheuristics. First, the combination of a constructive procedure to generate initial solutions and a local search heuristic to improve them was proposed under the name "Jump Search" (Tsubakitani & Evans, 1998). Obviously, the name did not catch on.

A more successful attempt to coin a new metaheuristic based on a combination of a constructive procedure and local search was GRASP. GRASP had the advantage over Jump Search because it specified a novel way to perform a constructive procedure, combining randomness and greediness. Arguably, this is the true innovation of GRASP, and the local search procedure is optional. The first GRASP in EJOR was Mavridou, Pardalos, Pitsoulis, and Resende (1998), followed in the same year by Ríos-Mercado and Bard (1998).

A combination of evolutionary techniques and simulated annealing (dubbed "Darwin and Boltzmann mixed strategy") is found in Tian, Ma, and Zhang (1998).

A full special issue (containing 27 contributions) on tabu search with a foreword by Glover (1998a), appears at the end of the 1990s. This issue focuses on various applications and enhancements of tabu search, a metaheuristic optimization technique. It includes articles on topics such as flow shop scheduling, early/tardy scheduling problems, resource-constrained scheduling, production line optimization, telecommunication network optimization, and audit scheduling. There are also papers on more specialized applications like nurse scheduling, forest harvest scheduling, feeder bus network design, and facility layout problems. Additionally, the issue explores the use of tabu search in global optimization for artificial neural networks and various problem-solving approaches in mixed integer programming, among other topics.

The first article in EJOR on guided local search is due to Voudouris and Tsang (1999). Guided local search is a new metaheuristic where the core objective is to optimize the navigation through the extensive and complicated landscapes of NP-hard optimization challenges. This optimization is facilitated by dynamically adjusting penalty terms to the base objective function. These modifications serve to fine-tune local search algorithms, allowing for a more targeted approach. Additionally, this strategy enables the prioritization of search efforts towards specific sectors of the search space, which are deemed more likely to offer high-value solutions. This is not the first appearance of this metaheuristic in the literature (it has been published in several proceedings), but it seems that it is the second time for it to appear in a journal (the first one was in Operations Research letters (Tsang & Voudouris, 1997)). The authors claim: "In this paper, we present the technique to the wider Operations Research (OR) audience by explaining its application to the TSP, a widely known problem in the OR community".

From the turn of the millennium, the number of papers on metaheuristics published in EJOR increases rapidly, and surveying them all becomes an impossible task. Some authors also try to create some order and some structure in the different methods being proposed, like Hertz and Kobler (2000) for evolutionary algorithms.

Traditionally, most metaheuristics have been applied to combinatorial optimization problems. From the 2000s, we see that adaptations of these frameworks are increasingly being introduced for other types of problems. In that vain, Chelouah and Siarry (2000) present an application of tabu search to continuous optimization. Jones, Mirrazavi, and Tamiz (2002), on the other hand, presents an early metaheuristic for *multi-objective optimization* In the same issue, Genetic Local Search for multi-objective optimization is proposed by Jaszkiewicz (2002). Conclusions: multi-objective optimization (also using metaheuristics) becomes quite a thing in the early 2000s.

From 2000 on, the role of EJOR in the metaheuristics literature increases in importance. Hansen and Mladenović (2001) cite some earlier papers on VNS and variants, but seems to be the first general introduction to this novel metaheuristic. Similarly, memetic algorithms are also quickly picked up by EJOR (França, Mendes, & Moscato, 2001).

The article by Taillard, Gambardella, Gendreau, and Potvin (2001) presents an integrated perspective on various memory-based metaheuristic techniques like taboo search, scatter search, genetic algorithms, and ant colonies. It observes that these different methods are converging in their implementation, leading to the proposal of a unified framework termed "Adaptive Memory Programming" (AMP). This paper reviews several recent methods applied to problems like quadratic assignment, vehicle routing, and graph coloring, reinterpreting them through the AMP lens. AMP is noted for its significant potential for parallelization and its capability to handle real-world and dynamic applications.

The first paper on ant colony optimization in EJOR, applies this framework to a bi-objective problem (T'kindt, Monmarché, Tercinet, & Laugt, 2002) in a special issue on graphs and scheduling dedicated to ECCO XIII conference, that contains many papers on heuristics and metaheuristics.

Volume 151, issue 2 of EJOR, published in December 2003, deserves a special mention in this section. This issue was published after the 18th EURO Summer/Winter Institute (ESWI XVIII) that took place during the spring 2000 in Switzerland (and which was attended by two of the authors of this paper). The topic of ESWI XVIII was "Meta-heuristics in Combinatorial Optimization". Issue 151 was the first special issue of EJOR devoted entirely to metaheuristics and contained an article describing some guidelines for the design of metaheuristics by the organizers of the winter institute (Hertz & Widmer, 2003). The Winter Institute also inspired the founding of EU/ME — the European Chapter on Metaheuristics (now called the EURO Working Group on Metaheuristics), the official working group on the topic supported by EURO.

Whether a coincidence or not, the number of papers on metaheuristics in EJOR increases dramatically following the 2003 special issue dedicated to the EURO Winter Institute on Combinatorial Optimization. Regular issues now typically contain one or more papers on metaheuristics, and it becomes impossible to survey a reasonable fraction of them. Several special issues and feature clusters on metaheuristics appear in the 2000s and 2010s, such as the special issue on applications of metaheuristics (Dullaert, Sevaux, Sörensen, & Springael, 2007), the special issue on Scatter Search (volume 169, issue 2 Martí, 2006), the special issue on the application of metaheuristics to continuous problems (volume 185, issue 3 Michalewicz & Siarry, 2008), and on multi-objective problems (volume 169, issue 3 Jaszkiewicz & Tuyttens, 2006).

The mid-2000s also sees the rise of *matheuristics* (i.e., combinations of (meta)heuristics and exact methods), even though that term is only coined later, e.g., by Nwana, Darby-Dowman, and Mitra (2005). Jourdan, Basseur, and Talbi (2009) try to create some order in this evolving field by presenting a taxonomy of matheuristics.

Griffis, Bell, and Closs (2012) presents a comprehensive overview on the use of metaheuristics in logistics and supply chain management.

We mention three more papers that underline the changed role of EJOR from a modest "follower" in the domain of metaheuristics to a flagship journal that is involved in the debate on the evolution of the field. In a paper entitled "metaheuristics in the large", Swan et al. (2022) emphasize the need for a formal framework to classify and design metaheuristics, akin to other machine learning algorithms. The authors highlight the necessity of a robust scientific and computational infrastructure to support the development, analysis, and comparison of new metaheuristic approaches, preventing fragmentation and reproducibility issues. Additionally, they advocate for standardized, explicit, machine-readable descriptions of metaheuristics to advance scientific progress and ensure rigor in communication and reproducibility of research results.

Secondly, Turkeš, Sörensen, and Hvattum (2021) introduce meta-analysis into the metaheuristics literature (see Section 8.2).

Finally, in a recent paper (Karimi-Mamaghan, Mohammadi, Meyer, Karimi-Mamaghan, & Talbi, 2022) present the state of the art on the integration of metaheuristics and machine learning, a topic that will definitely see more traction in the future (see Section 8.5).

7. Conferences and scientific associations

An important milestone in the metaheuristic field took place in 1995. In this year, a group of researchers from the University of Colorado, lead by Fred Glover, launched two important initiatives. On one hand, they celebrated the first conference specifically devoted to metaheuristics under the name Metaheuristic International Conference (MIC) in Breckenridge. On the other hand, the first issue of the Journal of Heuristics was released that year, devoted to metaheuristic methodologies, heuristic algorithms and their applications, with an editorial board in which most of the groups working on heuristics were somehow represented. Since then, the MIC has been celebrated every other year, totaling 15 conferences so far (the latest 15th edition was organized by one of the authors of this chapter in Lorient, France on June 4–7, 2024), and the Journal of Heuristics has published more than 100 issues.

As described in previous sections, many OR journals, including EJOR, have been giving more space to heuristics, and what started as a second option for researchers when the mathematical model failed, has nowadays become the first alternative tested given an optimization problem. Therefore, we may find heuristics in most operations research journals, as well as many computer science journals. In any of the 86 journals listed in the ISI Web of Knowledge under the category *Operations Research and Management Science*, we may find research papers devoted to describing a heuristic for an optimization problem.

We now mention the conferences mainly devoted to heuristic methodologies in line of the MIC. Genetic algorithms have been probably the most applied metaheuristics to solve optimization problems and its main conference GECCO is also one of the most popular conferences in the field. The Genetic and Evolutionary Computation Conferences

(GECCO) present the latest results in the growing field of genetic and evolutionary computation. It is the largest peer-reviewed conference in the field of Evolutionary Computation, and it is the main conference of the Special Interest Group on Genetic and Evolutionary Computation (SIGEVO) of the Association for Computing Machinery. Starting in 1999 in cooperation with the European Network of Excellence in Evolutionary Computing (EvoNet), their main topics include: genetic algorithms, genetic programming, evolution strategies, evolutionary programming, and their real-world applications.

In many cases, scientific conferences and journals are promoted by scientific societies. This is also true in optimization and operations research. EURO, the Association of European Operational Research Societies promotes operations research within Europe. Its American counterpart, INFORMS, is an international association for professionals in operations research, analytics, and management science. Similarly, IFORS is the International Federations of OR Societies, and ALIO focuses on their Latin American counterparts. These scientific societies organize large conferences in OR, usually with the same name as the association, and in all of them heuristics play an important role. General conferences in operations research, optimization or even artificial intelligence, more and more devote sessions, streams or mini-conferences to metaheuristics. For example, the annual IEEE Congress on Evolutionary Computation is one of the leading events in the field of evolutionary computation, and includes a stream on heuristics, metaheuristics and hyper-heuristics.

Special mention deserves the EURO Working Group on Metaheuristics called EU/ME — *the metaheuristics community*. It is a working group officially sanctioned and financially supported by EURO with the main purpose of providing a platform for communication among researchers, practitioners, and software developers in the field of metaheuristic optimization. EU/ME is the largest working group on metaheuristics worldwide, uniting over 1400 members from over 80 countries.

The European Conference on Evolutionary Computation in Combinatorial Optimization — EvoCop has also a long tradition supporting and promoting metaheuristics. It is organized by SPECIES, the Society for the Promotion of Evolutionary Computation in Europe and its Surroundings. Starting in 2004, EvoCop is a multidisciplinary conference that brings together researchers working on applications and theory of evolutionary computation methods and other metaheuristics for solving difficult combinatorial optimization problems appearing in various industrial, economic, and scientific domains. This conference is usually held together with EuroGP (devoted to Genetic Programming), EvoMUSART (evolutionary and biologically inspired music, sound, art and design), and EvoApplications (on the Applications of Evolutionary Computation), in a joint event collectively known as EvoStar (Evo*).

Several conferences devoted to particular aspects of metaheuristics have been also established during the last 20 years. The International Conference on Parallel Problem Solving From Nature (PPSN) brings together researchers and practitioners in the field of Natural Computing, the study of computing approaches which are gleaned from natural models. In their 18 editions since 1990, PPSN has evolved widening its scope, and it accepts nowadays any contribution in metaheuristics. The Learning and Intelligent Optimization Conference LION and the Hybrid Metaheuristics (HM) conference series are acting similarly.

On the other hand, there are some relatively small conferences devoted to specific methodologies that stick to their original design and limit their scope. This is the case of ANTS, an event dealing with swarm intelligence, behavioral models of social insects or other animal societies that can stimulate new algorithmic approaches. This is also the case of ICVNS specifically devoted to the variable neighborhood search metaheuristic and regularly co-organized with the EURO Working Group EU/ME.

8. What is next?

Previous works and analysis have shown that there is still a lot to do for the future of metaheuristics. This section presents some interesting directions and is, of course, far from being exhaustive.

8.1. The science of metaheuristics

The development of metaheuristics, despite its progress and practical successes, can still be considered more of a craft than a fully established scientific discipline. While metaheuristics have demonstrated their effectiveness in solving complex optimization problems, there are several reasons why they are perceived as lacking formal guidelines and a solid scientific foundation.

Lack of theoretical foundations. Metaheuristics are often developed based on intuition, heuristics, and trial-and-error approaches rather than being rooted in rigorous mathematical theories. Some metaheuristics draw inspiration from human intelligence, natural phenomena, or biological processes, yet the translation of these concepts into effective algorithms is often done in an informal and ad hoc manner.

Limited analytical understanding. Due to the complex nature of optimization problems and the complexity of the solution space of many (combinatorial) optimization problems, it is challenging to provide analytical proofs or guarantees of performance for metaheuristic algorithms. Unlike classical optimization methods that have well-defined convergence properties and theoretical analysis, metaheuristics rely more on empirical validation and experimental results. Moreover, existing mathematical analyses and/or theoretical proofs fall short of providing useful guidelines for the development of heuristics and metaheuristics. A proof of convergence *given an infinite amount of time*, e.g., is not particularly useful in practice.

Problem-specific tuning. Metaheuristics often require careful parameter tuning to achieve optimal performance on specific problem instances. The selection of appropriate parameter values typically relies on the experience and expertise of the practitioner, making it more of an art than a science. These parameters control the exploration–exploitation trade-off, and finding the right balance can be a challenging task. Of course, specialized tools exist to perform parameter tuning, but their benefits are still open for debate and their use is not very widespread.

Lack of standardized benchmarking protocols. Unlike in many scientific disciplines, there is no widely accepted set of benchmark problems for evaluating and comparing the performance of metaheuristic algorithms. While some problem domains have established benchmark suites, the coverage is often limited, and the results may heavily depend on the specific problem instances used. This lack of standardized benchmarking makes it difficult to objectively assess and compare different metaheuristic approaches. Moreover, the absence of benchmarking *protocols* allows researchers to cherry-pick the results that puts their own algorithm in the best possible light. It is still unclear, despite several research efforts, what it means to "outperform" a competing algorithm.

Absence of general guidelines. Metaheuristics are typically problem-agnostic, meaning they can be applied to a wide range of optimization problems. However, there are few general guidelines or rules-of-thumb that can guide practitioners in selecting the most suitable metaheuristic for a given problem. The choice of metaheuristic often relies on intuition, prior experience, or even personal preference, rather than being based on solid scientific guidelines.

Despite these limitations, it is worth noting that progress is being made in formalizing and advancing the field of metaheuristics. Researchers are working on developing theoretical foundations, creating standardized benchmark suites, and proposing methodologies for algorithm comparison and evaluation. As more studies and insights emerge, the craft of metaheuristics is gradually evolving into a more rigorous scientific discipline. However, it will still take time and further research to establish well-defined guidelines and a solid theoretical framework for metaheuristics.

8.2. Meta-analysis

From the domain of medicine comes a promising methodology called *meta-analysis*, a technique that is often used to establish the efficacy of a treatment. It involves systematically combining the results of multiple independent studies on a specific research question or topic. The primary goal of meta-analysis is to provide a more comprehensive and precise estimate of the effect or outcome being investigated than what individual studies alone can provide.

In a meta-analysis, researchers identify relevant studies from the existing literature through comprehensive literature searches. They then extract relevant data from each study and analyze them collectively. By pooling the data from multiple studies, meta-analysis increases the statistical power and reduces the impact of random variation and bias that may exist in individual studies.

Through statistical analysis, meta-analysis quantifies the overall effect size or magnitude of the relationship between variables or the effectiveness of a particular intervention or treatment. It can also explore factors that may influence the results, such as study characteristics or participant characteristics, through subgroup analyses or meta-regression.

Meta-analysis in medicine plays a crucial role in evidence-based practice and decision-making. It helps to synthesize existing research findings, resolve inconsistencies or controversies among individual studies, identify sources of heterogeneity, and provide more reliable estimates of treatment effects or associations between variables. By combining data from multiple studies, meta-analysis provides a broader and more robust perspective on the research question at hand, enhancing the overall understanding of a particular topic or intervention.

In a recent paper, Turkeš et al. (2021, 2020) present the first meta-analysis in the field of metaheuristics. The authors use this technique to gain insights into the importance of the adaptive layer in adaptive large neighborhood search (ALNS). ALNS is a widely used metaheuristic for solving various problems, but it remains unclear whether the adaptiveness of the algorithm actually contributes to its performance.

To conduct the meta-analysis, the authors identified a total of 134 relevant studies, out of which 63 met the eligibility criteria. They obtained results for 25 different implementations of ALNS by requesting data from the authors of the eligible studies. The collected data was then analyzed using a random-effects model.

The findings of the meta-analysis reveal that, on average, the addition of an adaptive layer in an ALNS algorithm improves the objective function value by a mere 0.14%. While the adaptive layer can provide added value in specific situations, it also introduces considerable complexity. The authors therefore conclude that its recommendation is limited to certain contexts.

Overall, this study emphasizes the importance of evaluating the contribution of metaheuristic components and the significance of knowledge gained through meta-analysis over solely relying on competitive testing. Nevertheless, few authors seem to be aware that such techniques exist, and/or are reluctant to apply them to understand the contribution of some metaheuristic components.

8.3. Instance space analysis

An early work from Rice (1976) reported the algorithm selection problem where a framework was given to select the most appropriate solving method for a specific problem. The Instance Space Analysis (ISA) (Smith-Miles & Muñoz, 2023), extending this initial work, is a novel approach that serves two main purposes: first, aiding the impartial evaluation of algorithms, and second, assessing the diversity of test instances used for evaluation of the different methods.

ISA, by using a vectorized representation of features, visualizes the entire space of possible test instances. ISA shows how algorithm performance is influenced by the characteristics of each instance. Rather than simply presenting algorithm performance based on average results over a selected set of test problems (the conventional method), ISA provides a more detailed comprehension of the unique strengths and weaknesses of algorithms across various sections of the instance space. These nuances might remain concealed when viewed solely through an average lens.

Additionally, ISA aids in the objective assessment of biases within the chosen test instances and offers guidance on the sufficiency of benchmark test suites.

8.4. Heuristic solvers

Exact solvers that attempt to find the optimal solution and that are generally based on linear and integer programming paradigms have a long history. Commercial solvers like CPLEX, Gurobi, etc. are household names, as are open source tools like GLPK and COIN-OR. All of these solvers rely on traditional simplex and other linear programming methods combined with advanced branch-and-cut or other methods for Integer Programming problems.

Recent years have seen the advent of a limited number of solvers that rely on heuristics, such as Andrade, Toso, Goncalves, and Resende (2021), Oliveira, Carravilla, Oliveira, and Resende (2022), Toso and Resende (2015). Given the flexibility of metaheuristic frameworks and paradigms like local search, combined with their success in solving combinatorial optimization problems both in research and in practice, this is perhaps a bit surprising.

A possible explanation for the lack of general-purpose heuristic-based solvers could be the absence of a commonly accepted formal paradigm to express different types of optimization problems. The paradigm of linear and integer programming requires the user to shoehorn every optimization problem into a strictly defined formal model structure, in which all decisions are defined in terms of either continuous or integer variables, and all objectives and constraints are either linear or linearized functions of these variables (with a few exceptions, like quadratic relationships, which can usually be handled directly).

Not all optimization problems, however, are naturally expressed in this way. A notable category of optimization problems for which the paradigm of linear and integer programming results in especially convoluted models, is the category of problems in which decisions are made on the *order* of a set of items. This category includes virtually all routing problems and all machine scheduling problems. Such problems are typically expressed in MIP models using binary variables that express whether an item i is immediately followed by another item j. Consider, e.g., the Miller–Tucker–Zemlin formulation for the TSP.

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j\neq i, j=1}^{n} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1, \qquad \forall j \in \{1,\dots,n\}, \tag{2}$$

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1, \qquad \forall i \in \{1,\dots,n\}, \tag{3}$$

$$u_i - u_j + n \cdot x_{ij} \leq n-1, \qquad \forall i,j \in \{1,\dots,n\}, i\neq j, \tag{4}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i,j \in \{1,\dots,n\}, i\neq j, \tag{5}$$

$$u_i \geq 1, \qquad \forall i \in \{1,\dots,n\}. \tag{6}$$

This model uses variables $x_{ij} \in \{0,1\}$ to express the fact that city j is the immediate successor of city i in the solution. This requires constraints (2) and (3) simply to ensure that each city is visited exactly once. Additionally, it requires an entire set of auxiliary variables u_i and an entire set of additional *subtour elimination constraints* (Eq. (4)) just to ensure that all cities are visited in a single tour, and not in a set of disjoint subtours.

The formulation of the TSP in the Hexaly Modeler language requires just three lines, each of which are easy to understand:

```
x <- list(n);
constraint count(x) == n;
minimize sum(1...n, i => c[x[i - 1]][x[i]])
 + c[x[n- 1]][x[0]];
```

The first line introduces `x` as a variable of type `list`. A list variable of size n is defined as a vector of variable size that contains each integer value between 0 and $n-1$ *at most* once. The only constraint necessary (line 2) restricts the size of this list to exactly n which, combined with the definition of the list variable, ensures that each city appears only once in the list. The last line minimizes the sum of inter-city distances.

The introduction of list variables (permutations) makes Hexaly models more expressive, and allows for more natural formulations of routing and scheduling problems. It also makes the model easier to read, as it does not require subtour elimination constraints and/or auxiliary variables. Similarly, other types of constraints that are difficult to express in the MIP paradigm (e.g., logical constraints), and that typically require a *big-M* constraint formulation, can be expressed and handled directly in Hexaly.

Hexaly is not unique, since other heuristic-based solvers exist (e.g., the open source TimeFold, formerly called OptaPlanner). There are also many software libraries available, of rather variable quality.

Moreover, some level of convergence is noticeable between exact and heuristic solvers: exact solvers like Gurobi and CPLEX extensively use heuristics in various parts of the solution process, while heuristic solvers like Hexaly also have exact techniques in their repertoire (e.g., to calculate bounds on the solution quality or determine that a solution is optimal).

8.5. Machine learning and metaheuristics

Machine learning has attracted significant attention for its potential in solving optimization problems due to its ability to learn patterns from data and find optimal solutions in complex scenarios. There are two ways of seeing the interaction of metaheuristics and machine learning.

First of all, many problems faced by designers of machine learning algorithms are essentially optimization problems for which heuristic optimization techniques can be (and have been) developed. A recent book shows this importance (Eddaly et al., 2023). Given the hype and interest of machine learning as a field of study, it is clear that this field could become one of the prime application areas for metaheuristics.

Another interest has taken roots in the seminal paper from Bengio, Lodi, and Prouvost (2021). It is however more concerned with the use of machine learning to support the development of metaheuristics, rather than the reverse. As mentioned, the development of heuristics remains a craft that requires experience by what we could call a "heuristic engineer". In recent years, several researchers have leveraged the potential of machine learning algorithms to essentially learn how to develop, guide or perfectly tune a metaheuristic.

Recent work by Lucas, Billot, Sevaux, and Sörensen (2020), as expected by Bengio, has shown that combination machine learning methods and metaheuristics could give a potential advantage in the search of solutions. But, as also expected, those methods are time consuming and in the hypothesis of a race for the best solutions, the metaheuristics might largely win.

8.6. Quantum metaheuristics

Quantum metaheuristics utilize principles from quantum computing to improve traditional metaheuristics algorithms. These new methods are inspired by quantum phenomena such as superposition, entanglement, and interference to potentially improve the efficiency and effectiveness of optimization algorithms.

D-Wave systems, a pioneer in this field, has developed quantum annealing machines. This technique searches the global minimum of a given function by exploiting quantum tunneling. Quantum annealing or quantum-inspired evolutionary algorithms, aim to mimic the idea of tunneling to efficiently explore and traverse vast solution spaces normally unexplored with classical metaheuristics. In addition, quantum-inspired evolutionary algorithms mimic superposition and entanglement within classical computing frameworks. They adapt classical metaheuristics, such as genetic algorithms or particle swarm optimization, to introduce quantum-inspired operators that might offer advantages in exploration and exploitation. Other algorithms, like the quantum approximate optimization algorithm (QAOA), leverage quantum circuits to solve combinatorial optimization problems. They aim to find approximate solutions using variational approaches, potentially applicable to metaheuristic optimization.

Quantum-inspired metaheuristics represent a burgeoning field investigating the integration of quantum computing principles into classical optimization methods. Despite the early stages of quantum computing, these techniques hold great promises for efficiently addressing intricate optimization problems.

9. Conclusions

In this paper we have presented a historical review of metaheuristic frameworks to guide the design of heuristic optimization algorithms. Most of these frameworks resulted from the adaptation of heuristic reasoning, developed in the 40s in mathematical psychology, to operations research in the late 50s. The need of OR practitioners to solve hard optimization problems, for which classic methods based on mathematical models were not able to produce practical solutions, was the context in which heuristics emerged as the scientific solution.

Simple heuristics were introduced for combinatorial optimization problems in the 60s and 70s, and most of the metaheuristic frameworks were proposed in the 80s and 90s, in the period now known as the *method-centric*. We have seen how the advent of metaheuristics coincides with the "OR crisis" of the same period, in which there was a general sentiment that OR as a science was not achieving its goals and was more focused on building ever more intricate methods than on developing practical solution methods. Metaheuristics introduced the idea of following a set of rules (framework) to efficiently explore the solution space when creating a heuristic. Even though this lead to a significantly improved workflow and to better heuristics, this came with the downside that some researchers felt that describing a methodology inspired in natural or social behavior would constitute a contribution in itself. As a result, the mimicking of natural models has created confusion in the field with little contribution in terms of problem solving.

In our discussion, we also highlighted the role of EJOR, publishing solid heuristic papers during the last 40 years, but taking its time to enter in the field.

In our view, the field of heuristic optimization has reached a maturity that permits nowadays to solve very complex problems, with a growing number of researchers applying them, as shown in the numerous conferences and related events. On the other hand, there are some deficiencies that reveal areas of improvement. We observe a lot of fragmentation, and each group of research usually applies the same methods regardless the type of problem being solved. We do not know yet which method performs better in which type of problem.

The No Free Lunch (NFL) theorem (Wolpert & Macready, 1997) in the context of metaheuristics essentially states that there is no single metaheuristic that universally outperforms all others across all types of problems. This theorem highlights the importance of understanding problem-specific characteristics and selecting or designing appropriate metaheuristics based on these characteristics. For practitioners in the field of metaheuristics, this theorem emphasizes the need to experiment and adapt algorithms to suit specific problem instances, rather than relying on a one-size-fits-all approach. It encourages researchers and practitioners to develop a diverse set of algorithms and to understand the problem domain to choose or design the most effective metaheuristic for a given problem and for a given set of instances. We hope that this review helps researchers in understanding the metaheuristic field and in using these frameworks to develop powerful optimization algorithms.

CRediT authorship contribution statement

Rafael Martí: Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Marc Sevaux:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Kenneth Sörensen:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization.

Acknowledgments

Rafael Martí's research is partially supported by ERDF -A way of making Europe- with grant PID2021-125709OB-C21 (MCIN/AEI/10.13039/501100011033), and by Generalitat Valenciana, Spain (code CIAICO/2021/224). Marc Sevaux's research is partially supported by the French Agence Nationale de la Recherche (ANR), under grant ANR-22-CE22-0016-01 (project MAMUT).

References

Ackoff, R. L. (1979). The future of operational research is past. *Journal of the Operational Research Society*, *30*(2), 93–104. http://dx.doi.org/10.2307/3009290.

Adenso-Díaz, B. (1992). Restricted neighborhood in the tabu search for the flowshop problem. *European Journal of Operational Research*, *62*(1), 27–37. http://dx.doi.org/10.1016/0377-2217(92)90174-8.

Andrade, C. E., Toso, R. F., Goncalves, J. F., & Resende, M. G. (2021). The multi-parent biased random-key genetic algorithm with implicit path-relinking and its real-world applications. *European Journal of Operational Research*, *289*(1), 17–30.

Aranha, C., Villalón, C. L. C., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., et al. (2021). Metaphor-based metaheuristics, a call for action: the elephant in the room. *Swarm Intelligence*, *16*(1), 1–6. http://dx.doi.org/10.1007/s11721-021-00202-9.

Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, *290*(2), 405–421. http://dx.doi.org/10.1016/j.ejor.2020.07.063.

Bjorndal, M., Caprara, A., Cowling, P., Della Croce, F., Lourenço, H., Malucelli, F., et al. (1995). Some thoughts on combinatorial optimisation. *European Journal of Operational Research*, *83*(2), 253–270. http://dx.doi.org/10.1016/0377-2217(95)00005-B.

Bölte, A., & Thonemann, U. W. (1996). Optimizing simulated annealing schedules with genetic programming. *European Journal of Operational Research*, *92*(2), 402–416. http://dx.doi.org/10.1016/0377-2217(94)00350-5.

Burkard, R. E., & Rendl, F. (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, *17*(2), 169–174. http://dx.doi.org/10.1016/0377-2217(84)90231-5.

Camacho-Villalón, C. L., Dorigo, M., & 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*(6), 2945–2971. http://dx.doi.org/10.1111/itor.13176.

Chelouah, R., & Siarry, P. (2000). Tabu search applied to global optimization. *European Journal of Operational Research*, *123*(2), 256–270. http://dx.doi.org/10.1016/S0377-2217(99)00255-6.

Chen, W.-H., & Srivastava, B. (1994). Simulated annealing procedures for forming machine cells in group technology. *European Journal of Operational Research*, *75*(1), 100–111. http://dx.doi.org/10.1016/0377-2217(94)90188-0.

Clarke, G., & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*(4), 568–581. http://dx.doi.org/10.1287/opre.12.4.568.

Cosmetatos, G. P. (1977). Cobham's model on nonpreemptive multi-server queueing systems: A heuristic method for its generalization. *European Journal of Operational Research*, *1*(4), 262–264. http://dx.doi.org/10.1016/0377-2217(77)90096-0.

Cowling, P., Kendall, G., & Soubeiga, E. (2001). A hyperheuristic approach to scheduling a sales summit. In *Selected papers from the 3rd international conference on the practice and theory of automated timetabling (PATAT 2001). lecture notes in computer science, vol. 2079* (pp. 176–190). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/3-540-44629-x_11.

Croes, G. A. (1958). A method for solving traveling-salesman-problems. *Operations Research*, *6*(6), 791–812. http://dx.doi.org/10.1287/opre.6.6.791.

Dantzig, G. B. (1951). Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans (Ed.), *Activity analysis of production and allocation* (pp. 339–347). Wiley, New York.

Dantzig, G. B., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling salesman problem. *Operations Research*, *2*, 393–410. http://dx.doi.org/10.1287/opre.2.4.393.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. http://dx.doi.org/10.1007/BF01386390.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39. http://dx.doi.org/10.1109/MCI.2006.329691.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, *1*(26), 29–41. http://dx.doi.org/10.1109/3477.484436.

Dowsland, K. A. (1993). Some experiments with simulated annealing techniques for packing problems. *European Journal of Operational Research*, *68*(3), 389–399. http://dx.doi.org/10.1016/0377-2217(93)90195-S.

Duin, C., & Voß, S. (1999). The pilot method: A strategy for heuristic repetition with application to the steiner problem in graphs. *Networks: An International Journal*, *34*(3), 181–191, URL https://api.semanticscholar.org/CorpusID:6902893.

Dullaert, W., Sevaux, M., Sörensen, K., & Springael, J. (2007). Applications of metaheuristics. *European Journal of Operational Research*, *179*(3), 601–604. http://dx.doi.org/10.1016/j.ejor.2005.03.060.

Eddaly, et al. (2023). *Metaheuristics for machine learning*. Springer Nature Singapore, http://dx.doi.org/10.1007/978-981-19-3888-7,

Eglese, R. W. (1990). Simulated annealing: A tool for operational research. *European Journal of Operational Research*, *46*(3), 271–281. http://dx.doi.org/10.1016/0377-2217(90)90001-R.

Feo, T., & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, *8*, 67–71. http://dx.doi.org/10.1016/0167-6377(89)90002-3.

Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, *6*, 109–133. http://dx.doi.org/10.1007/BF01096763.

Fogel, L., Owens, A., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. New York: Wiley.

Ford, L., & Fulkerson, D. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, *8*, 399–404. http://dx.doi.org/10.4153/cjm-1956-045-5.

França, P. M., Mendes, A., & Moscato, P. (2001). A memetic algorithm for the total tardiness single machine scheduling problem. *European Journal of Operational Research*, *132*(1), 224–242. http://dx.doi.org/10.1016/S0377-2217(00)00140-5.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, *8*, 156–166. http://dx.doi.org/10.1111/j.1540-5915.1977.tb01074.x.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*, 533–549. http://dx.doi.org/10.1016/0305-0548(86)90048-1.

Glover, F. (1989). Tabu search—part i. *ORSA Journal on Computing*, *1*(3), 190–206.

Glover, F. (1997). A template for scatter search and path relinking. In *European conference on artificial evolution* (pp. 1–51). Springer.

Glover, F. (1998a). Tabu search — wellsprings and challenges. *European Journal of Operational Research*, *106*(2), 221–225. http://dx.doi.org/10.1016/S0377-2217(97)00259-2.

Glover, F. (1998b). A template for scatter search and path relinking. In *Lecture notes in computer science* (pp. 1–51). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/bfb0026589.

Glover, F., & Greenberg, H. J. (1989). New approaches for heuristic search: A bilateral linkage with artificial intelligence. *European Journal of Operational Research*, *39*(2), 119–130. http://dx.doi.org/10.1016/0377-2217(89)90185-9.

Glover, F., & Laguna, M. (1993). Tabu search. In *Modern heuristic techniques for combinatorial problems* (pp. 70–150).

Glover, F., Laguna, M., & Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, *29*(3), 653–684, URL https://tinyurl.com/glover-ea.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. New York: Addison-Wesley.

Gonçalves, J. F., de Magalhães Mendes, J. J., & Resende, M. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, *167*(1), 77–95.

Gonçalves, J. F., & Resende, M. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, (17), 487–525.

Griffis, S. E., Bell, J. E., & Closs, D. J. (2012). Metaheuristics in logistics and supply chain management. *Journal of Business Logistics*, *33*(2), 90–106. http://dx.doi.org/10.1111/j.0000-0000.2012.01042.x.

Grolimund, S., & Ganascia, J.-G. (1997). Driving tabu search with case-based reasoning. *European Journal of Operational Research*, *103*(2), 326–338. http://dx.doi.org/10.1016/S0377-2217(97)00123-9.

Hansen, P. (1989). A short discussion of the OR crisis. *European Journal of Operational Research*, *38*(3), 277–281. http://dx.doi.org/10.1016/0377-2217(89)90003-9.

Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, *130*(3), 449–467. http://dx.doi.org/10.1016/S0377-2217(00)00100-4.

Hertwig, R., & Pachur, T. (2015). History of heuristics. In J. D. Wright (Ed.), *International encyclopedia of the social & behavioral sciences* (pp. 829–835). Elsevier.

Hertz, A. (1991). Tabu search for large scale timetabling problems. *European Journal of Operational Research*, *54*(1), 39–47. http://dx.doi.org/10.1016/0377-2217(91)90321-L.

Hertz, A., & Kobler, D. (2000). A framework for the description of evolutionary algorithms. *European Journal of Operational Research*, *126*(1), 1–12. http://dx.doi.org/10.1016/S0377-2217(99)00435-X.

Hertz, A., & Widmer, M. (2003). Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, *151*(2), 247–252. http://dx.doi.org/10.1016/S0377-2217(02)00823-8.

Hjeij, M., & Vilks, A. (2023). A brief history of heuristics: how did research on heuristics evolve? *Humanities & Social sciences Communications*, *10*, 64. http://dx.doi.org/10.1057/s41599-023-01542-z.

Holland, J. H. (1975a). Adaptation in natural and artificial systems. Ann Arbor, MI, USA: University of Michigan Press.

Holland, J. H. (1975b). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press.

Jaszkiewicz, A. (2002). Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, *137*(1), 50–71. http://dx.doi.org/10.1016/S0377-2217(01)00104-7.

Jaszkiewicz, A., & Tuyttens, D. (2006). Metaheuristics in multiple objective optimization. *European Journal of Operational Research*, *169*(3), 873–874. http://dx.doi.org/10.1016/j.ejor.2004.10.007.

Jeffcoat, D. E., & Bulfin, R. L. (1993). Simulated annealing for resource-constrained scheduling. *European Journal of Operational Research*, *70*(1), 43–51. http://dx.doi.org/10.1016/0377-2217(93)90231-B.

Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1989). Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations Research*, *37*(6), 865–892, URL https://www.jstor.org/stable/171393.

Jones, D., Mirrazavi, S., & Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, *137*(1), 1–9. http://dx.doi.org/10.1016/S0377-2217(01)00123-0.

Jørgensen, R. M., Thomsen, H., & Vidal, R. V. (1992). The afforestation problem: A heuristic method based on simulated annealing. *European Journal of Operational Research*, *56*(2), 184–191. http://dx.doi.org/10.1016/0377-2217(92)90221-T.

Jourdan, L., Basseur, M., & Talbi, E.-G. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, *199*(3), 620–629. http://dx.doi.org/10.1016/j.ejor.2007.07.035.

Karg, R. L., & Thompson, G. L. (1964). A heuristic approach to solving travelling salesman problems. *Management Science*, *10*(2), 225–248. http://dx.doi.org/10.1287/MNSC.10.2.225.

Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E.-G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, *296*(2), 393–422. http://dx.doi.org/10.1016/j.ejor.2021.04.032.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *vol. 4*, In *Proceedings of the IEEE international conference on neural networks* (pp. 1942–1948). http://dx.doi.org/10.1109/ICNN.1995.488968.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680. http://dx.doi.org/10.1126/science.220.4598.671.

Kouvelis, P., Chiang, W.-C., & Fitzsimmons, J. (1992). Simulated annealing for machine layout problems in the presence of zoning constraints. *European Journal of Operational Research*, *57*(2), 203–223. http://dx.doi.org/10.1016/0377-2217(92)90043-9.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the Americal Mathematical Society*, *7*(1), 48–50. http://dx.doi.org/10.1090/S0002-9939-1956-0078686-7.

Kuhn, H. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97. http://dx.doi.org/10.1002/nav.3800020109.

Laguna, M., Kelly, J. P., González-Velarde, J., & Glover, F. (1995). Tabu search for the multilevel generalized assignment problem. *European Journal of Operational Research*, *82*(1), 176–189. http://dx.doi.org/10.1016/0377-2217(93)E0174-V.

Laguna, M., & Martí, R. (1999). GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, *11*(1), 44–52.

Laursen, P. S. (1993). Simulated annealing for the QAP — Optimal tradeoff between simulation time and solution quality. *European Journal of Operational Research*, *69*(2), 238–243. http://dx.doi.org/10.1016/0377-2217(93)90167-L.

Lourenço, H. R. (1995). Job-shop scheduling: Computational study of local search and large-step optimization methods. *European Journal of Operational Research*, *83*(2), 347–364.

Lucas, F., Billot, R., Sevaux, M., & Sörensen, K. (2020). Reducing space search in combinatorial optimization using machine learning tools. In *Learning and intelligent optimization* (pp. 143–150). Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-53552-0_15.

Mack, H. L., & Smith, L. D. (1978). Scheduling with list processing and problem oriented data structures: An itinerant teacher example. *European Journal of Operational Research*, *2*(3), 175–184. http://dx.doi.org/10.1016/0377-2217(78)90090-5.

Maniezzo, V., Dorigo, M., & Colorni, A. (1995). Algodesk: An experimental comparison of eight evolutionary heuristics applied to the quadratic assignment problem. *European Journal of Operational Research*, *81*(1), 188–204. http://dx.doi.org/10.1016/0377-2217(93)E0128-K.

Martí, R. (2006). Scatter search methods for optimization. *European Journal of Operational Research*, *169*, 351–697.

Martí, R., Corberán, A., & Peiró, J. (2015). Scatter search. In *Handbook of heuristics* (pp. 1–24). Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-07153-4_20-1.

Martí, R., Laguna, M., & Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research*, *169*, 359–372.

Másson, E., & Wang, Y.-J. (1990). Introduction to computation and learning in artificial neural networks. *European Journal of Operational Research*, *47*(1), 1–28. http://dx.doi.org/10.1016/0377-2217(90)90085-P.

Mavridou, T., Pardalos, P., Pitsoulis, L., & Resende, M. G. (1998). A GRASP for the biquadratic assignment problem. *European Journal of Operational Research*, *105*(3), 613–621. http://dx.doi.org/10.1016/S0377-2217(97)00083-0.

Michalewicz, Z., & Siarry, P. (2008). Feature cluster on adaptation of discrete metaheuristics to continuous optimization. *European Journal of Operational Research*, *185*(3), 1060–1061. http://dx.doi.org/10.1016/j.ejor.2006.09.009.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, (24), 1097–1100. http://dx.doi.org/10.1016/S0305-0548(97)00031-2.

Moscato, P. (1989). *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*: *Technical report C3P report 826*, California Institute of Technology, URL https://api.semanticscholar.org/CorpusID:1264156.

Moscato, P. (1993). An introduction to population approaches for optimization and hierarchical objective functions: The role of tabu search. *Annals of Operations Research*, *41*, 85–121. http://dx.doi.org/10.1007/BF02022564.

Müller-Merbach, H. (1981). Heuristics and their design: a survey. *European Journal of Operational Research*, *8*(1), 1–23. http://dx.doi.org/10.1016/0377-2217(81)90024-2.

Nicholson, T. (1971). *Optimization in Industry: Optimization Techniques*: *vol. 1*, Aldine.

Nwana, V., Darby-Dowman, K., & Mitra, G. (2005). A co-operative parallel heuristic for mixed zero–one linear programming: Combining simulated annealing with branch and bound. *European Journal of Operational Research*, *164*(1), 12–23. http://dx.doi.org/10.1016/j.ejor.2002.12.002.

Oliveira, B. B., Carravilla, M. A., Oliveira, J. F., & Resende, M. G. (2022). A C++ application programming interface for co-evolutionary biased random-key genetic algorithms for solution and scenario generation. *Optimization Methods & Software*, *37*, 1065–1086.

Pflug, G., & Ruszczyński, A. (1997). Thirteenth EURO summer institute: Stochastic optimization. *European Journal of Operational Research*, *101*(2), 229. http://dx.doi.org/10.1016/S0377-2217(96)00394-3.

Pirlot, M. (1996). General local search methods. *European Journal of Operational Research*, *92*(3), 493–511. http://dx.doi.org/10.1016/0377-2217(96)00007-0.

Polya, G. (1945). *How to solve it*. Princeton University Press.

Prais, M., & Ribeiro, C. C. (2000). Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS Journal on Computing*, *12*(3), 164–176. http://dx.doi.org/10.1287/ijoc.12.3.164.12639.

Prim, R. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, *36*(6), 1389–1401. http://dx.doi.org/10.1002/j.1538-7305.1957.tb01515.x.

Rego, C., & Roucairol, C. (1995). Using tabu search for solving a dynamic multi-terminal truck dispatching problem. *European Journal of Operational Research*, *83*(2), 411–429. http://dx.doi.org/10.1016/0377-2217(95)00016-J.

Resende, M. G. C., & Ribeiro, C. C. (2016). Path-relinking. In *Optimization by GRASP* (pp. 167–188). Springer New York, http://dx.doi.org/10.1007/978-1-4939-6530-4_8.

Rice, J. R. (1976). The algorithm selection problem. *15*, In *Advances in computers* (pp. 65–118). Elsevier, http://dx.doi.org/10.1016/s0065-2458(08)60520-3.

Righini, G. (1995). A double annealing algorithm for discrete location/allocation problems. *European Journal of Operational Research*, *86*(3), 452–468. http://dx.doi.org/10.1016/0377-2217(95)98957-2.

Rinnooy Kan, A., & Timmer, G. (1987). Stochastic global optimization methods; part I: Clustering methods. *Mathematical Programming*, *37*, 27–56. http://dx.doi.org/10.1007/BF02592071.

Ríos-Mercado, R. Z., & Bard, J. F. (1998). Heuristics for the flow line problem with setup costs. *European Journal of Operational Research*, *110*(1), 76–98. http://dx.doi.org/10.1016/S0377-2217(97)00213-0.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*(4), 455–472. http://dx.doi.org/10.1287/trsc.1050.0135.

Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, *177*(3), 2033–2049.

Sharda, R., & Wang, J. (1996). Neural networks and operations research/management science. *European Journal of Operational Research*, *93*(2), 227–229. http://dx.doi.org/10.1016/0377-2217(96)00032-X.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming* (pp. 417–431). Springer, http://dx.doi.org/10.1007/3-540-49481-2_30.

Silver, E. A., Victor, R., Vidal, V., & de Werra, D. (1980). A tutorial on heuristic methods. *European Journal of Operational Research*, *5*(3), 153–162. http://dx.doi.org/10.1016/0377-2217(80)90084-3.

Simon, H. A., & Newell, A. (1958). Heuristic problem solving: The next advance in operations research. *Operations Research*, *6*(1), 1–10. http://dx.doi.org/10.1287/opre.6.1.1.

Skorin-Kapov, D., & Skorin-Kapov, J. (1994). On tabu search for the location of interacting hub facilities. *European Journal of Operational Research*, *73*(3), 502–509. http://dx.doi.org/10.1016/0377-2217(94)90245-3.

Smith-Miles, K., & Muñoz, M. A. (2023). Instance space analysis for algorithm testing: Methodology and software tools. *ACM Computing Surveys*, *55*(12), 1–31. http://dx.doi.org/10.1145/3572895.

Sofianopoulou, S. (1992). Simulated annealing applied to the process allocation problem. *European Journal of Operational Research*, *60*(3), 327–334. http://dx.doi.org/10.1016/0377-2217(92)90084-M.

Solis, F., & Wets, R. (1981). Minimization by random search techniques. *Mathematics of Operations Research*, (6(1)), 19–30.

Sörensen, K. (2015). Metaheuristics - the metaphor exposed. *International Transactions in Operational Research*, *22*(1), 3–18. http://dx.doi.org/10.1111/itor.12001.

Sörensen, K., & Glover, F. (2013). Metaheuristics. In S. I. Gass, & M. C. Fu (Eds.), *Encyclopedia of operations research and management science* (pp. 960–970). Boston, MA: Springer New York, http://dx.doi.org/10.1007/978-1-4419-1153-7_1167.

Sörensen, K., Sevaux, M., & Glover, F. (2018). A history of metaheuristics. In *Handbook of heuristics* (pp. 791–808). Springer, http://dx.doi.org/10.1007/978-3-319-07124-4_4.

Swan, J., Adriaensen, S., Brownlee, A. E., Hammond, K., Johnson, C. G., Kheiri, A., et al. (2022). Metaheuristics "in the large". *European Journal of Operational Research*, *297*(2), 393–406. http://dx.doi.org/10.1016/j.ejor.2021.05.042.

Taillard, E. D. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, *47*(1), 65–74. http://dx.doi.org/10.1016/0377-2217(90)90090-X.

Taillard, E. D. (2023). *Design of heuristic algorithms for hard optimization*. Springer International Publishing, http://dx.doi.org/10.1007/978-3-031-13714-3.

Taillard, E. D., Gambardella, L. M., Gendreau, M., & Potvin, J.-Y. (2001). Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, *135*(1), 1–16. http://dx.doi.org/10.1016/S0377-2217(00)00268-X.

Tam, K. Y. (1992). Genetic algorithms, function optimization, and facility layout design. *European Journal of Operational Research*, *63*(2), 322–346. http://dx.doi.org/10.1016/0377-2217(92)90034-7.

Tian, P., Ma, J., & Zhang, D.-M. (1998). Non-linear integer programming by darwin and Boltzmann mixed strategy. *European Journal of Operational Research*, *105*(1), 224–235. http://dx.doi.org/10.1016/S0377-2217(97)00024-6.

T'kindt, V., Monmarché, N., Tercinet, F., & Laugt, D. (2002). An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research*, *142*(2), 250–257. http://dx.doi.org/10.1016/S0377-2217(02)00265-5.

Tonge, F. (1960). Summary of a heuristic line balancing procedure. *Management Science*, *7*(1), 21–42. http://dx.doi.org/10.1287/mnsc.7.1.21.

Toso, R. F., & Resende, M. G. (2015). A C++ application programming interface for biased random-key genetic algorithms. *Optimization Methods & Software*, *30*(1), 81–93.

Tsang, E., & Voudouris, C. (1997). Fast local search and guided local search and their application to british telecom's workforce scheduling problem. *Operations Research Letters*, *20*(3), 119–127. http://dx.doi.org/10.1016/S0167-6377(96)00042-9.

Tsubakitani, S., & Evans, J. R. (1998). An empirical study of a new metaheuristic for the traveling salesman problem. *European Journal of Operational Research*, *104*(1), 113–128. http://dx.doi.org/10.1016/S0377-2217(96)00334-7.

Turkeš, R., Sörensen, K., & Hvattum, L. M. (2021). Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search. *European Journal of Operational Research*, *292*(2), 423–442. http://dx.doi.org/10.1016/j.ejor.2020.10.045.

Turkeš, R., Sörensen, K., Hvattum, L. M., Barrena, E., Chentli, H., Coelho, L. C., et al. (2020). Data for a meta-analysis of the adaptive layer in adaptive large neighborhood search. *Data in Brief*, *33*, Article 106568. http://dx.doi.org/10.1016/j.dib.2020.106568.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, *185*(4157), 1124–1131. http://dx.doi.org/10.1126/science.185.4157.1124.

Voudouris, C., & Tsang, E. (1999). Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research*, *113*(2), 469–499. http://dx.doi.org/10.1016/S0377-2217(98)00099-X.

Widmer, M., & Hertz, A. (1989). A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*, *41*(2), 186–193. http://dx.doi.org/10.1016/0377-2217(89)90383-4.

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. http://dx.doi.org/10.1109/4235.585893.

Zanakis, S. H., Evans, J. R., & Vazacopoulos, A. A. (1989). Heuristic methods and applications: A categorized survey. *European Journal of Operational Research*, *43*(1), 88–110. http://dx.doi.org/10.1016/0377-2217(89)90412-8.