A History of Metaheuristics

27

Kenneth Sörensen, Marc Sevaux, and Fred Glover

Contents

Abstract

This chapter describes the history of metaheuristics in five distinct periods, starting long before the first use of the term and ending a long time in the future.

The field of metaheuristics has undergone several paradigm shifts that have changed the way researchers look upon the development of heuristic methods.

K. Sörensen (✉)
University of Antwerp, Antwerp, Belgium
e-mail: kenneth.sorensen@uantwerpen.be

M. Sevaux
Université de Bretagne-Sud, Lab-STICC, CNRS, Lorient, France
e-mail: marc.sevaux@univ-ubs.fr

F. Glover
OptTek Systems, Inc, Boulder, CO, USA
e-mail: glover@opttek.com

R. Martí et al. (eds.), *Handbook of Heuristics*,
https://doi.org/10.1007/978-3-319-07124-4_4

Most notably, there has been a shift from the *method-centric* period, in which metaheuristics were thought of as algorithms, to the *framework-centric* period, in which researchers think of metaheuristics as more general high-level frameworks, i.e., consistent collections of concepts and ideas that offer guidelines on how to go about solving an optimization problem heuristically.

Tremendous progress has been made in the development of heuristics over the years. Optimization problems that seemed intractable only a few decades ago can now be efficiently solved. Nevertheless, there is still much room for evolution in the research field, an evolution that will allow it to move into the *scientific period*. In this period, we will see more structured knowledge generation that will benefit both researchers and practitioners.

Unfortunately, a significant fraction of the research community has deluded itself into thinking that scientific progress can be made by resorting to ever more outlandish metaphors as the basis for so-called "novel" methods. Even though considerable damage to the research field will have been inflicted by the time these ideas have been stamped out, there is no doubt that science will ultimately prevail.

Keywords
History

Introduction

Even though people have used heuristics throughout history, and the human brain is equipped with a formidable heuristic engine to solve an enormous array of challenging optimization problems, the *scientific study* of heuristics (and, by extension, metaheuristics) is a relatively young endeavor. It is not an exaggeration to claim that the field of (meta) heuristics, especially compared to other fields of study like physics, chemistry, and mathematics, has yet to reach a mature state. Nevertheless, enormous progress has been made since the first metaheuristics concepts were established. In this chapter, we will attempt to describe the historical developments this field of study has gone through since its earliest days.

No history is ever neutral, and a history of metaheuristics – or any other topic – can be written in many different ways. A straightforward (one could say "easy") history of metaheuristics would consist of an annotated and chronological list of metaheuristic methods. Useful as such a list may be, it suffers from a lack of insight into the development of the field as a whole. To illustrate this viewpoint, consider the list in Fig. 1 that appeared on Wikipedia until April 8, 2013, to illustrate the "most important contributions" in the field of metaheuristics. It is our opinion that such a list is not particularly enlightening (and neither was the article that contained it) when it comes to explaining the evolution of the field of metaheuristics.

Taking a bird's eye view of the field of metaheuristics, one has to conclude that there has been a large amount of progressive insight over the years. Moreover,

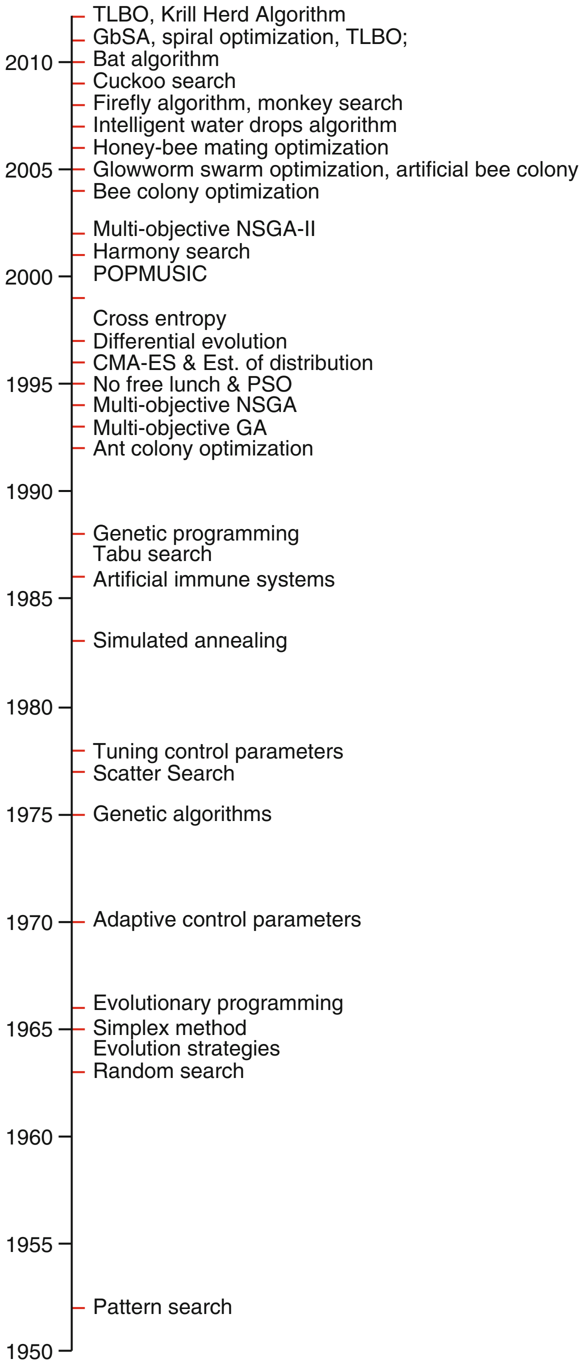

Fig. 1 "Most important contributions" list as it appeared on Wikipedia until April 8, 2013

this progressive insight has not reached its end point: the way researchers and practitioners look at metaheuristics is still continually shifting. Even the answer to the question what a metaheuristic is has changed quite a lot since the word was first coined in the second half of the 1980s. In our view, it is this shifting viewpoint that deserves to be written down, as it allows us to truly understand the past and perhaps learn a few lessons that could be useful for the future development of research in metaheuristics. We did not limit our discussion in this chapter to metaheuristics that have been formally written down and published. When studying the history of metaheuristics with an open mind, one has to conclude that people have been using heuristics and metaheuristics long before the term even existed.

We have therefore adopted a different approach to write "our" history of metaheuristics. Our approach starts well before the term "metaheuristic" was coined and is based on the premise that people have looked at metaheuristics through different sets of glasses over the years. The way in which people – not only researchers – have interpreted the different metaheuristic concepts has shaped the way in which the field has been developing. To understand the design choices that people have been making when developing metaheuristic optimization algorithms, it is paramount that these choices are understood in relationship to the trends and viewpoints of the time during which the development took place.

Our history divides time in five distinct periods. The crispness of the boundaries between each pair of consecutive periods, however, is a gross simplification of reality. The real (if one can use that word) time periods during which the paradigm shifts took place are usually spaced out over several years, but it is difficult, if not impossible, to trace the exact moments in time at which the paradigm shifts began and ended. More importantly, not every researcher necessarily makes the transition at the same time.

- The *pre-theoretical* period (until c. 1940), during which heuristics and even metaheuristics are used but not formally studied.
- The *early* period (c. 1940–c. 1980), during which the first formal studies on heuristics appear.
- The *method-centric* period (c. 1980–c. 2000), during which the field of metaheuristics truly takes off and many different methods are proposed.
- The *framework-centric* period (c. 2000–now), during which the insight grows that metaheuristics are more usefully described as frameworks and not as methods.
- The *scientific* period (the future), during which the design of metaheuristics becomes a science instead of an art.

Until recently, a clear definition of the word metaheuristic has been lacking, and it could be argued that it is still disputed. In this chapter, we adopt the definition of Sörensen and Glover [39].

> A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. The term is also used to refer to a problem-specific implementation of a heuristic optimization algorithm according to the guidelines expressed in such a framework.

The term “metaheuristic” has been used (and is used) for two entirely different things. One is a high-level framework, a set of concepts and strategies that blend together and offer a perspective on the development of optimization algorithms. In this sense, variable neighborhood search [28] is nothing more (or less) than the idea to use different local search operators to work on a single solution, together with a perturbation operator once all neighborhoods have reached a local optimum. There is a compelling motivation, as well a large amount of empirical evidence, as to why multi-neighborhood search is indeed a very good idea. This motivation essentially comes down to the fact that a local optimum for one local search operator (or one neighborhood structure) is usually not a local optimum for another local search operator. The idea to switch to a different local search operator once a local optimum has been found is therefore both sensible and in practice extremely powerful.

The second meaning of the term “metaheuristic” denotes a specific implementation of an algorithm based on such a framework (or on a combination of concepts from different frameworks) designed to find a solution to a specific optimization problem. The variable neighborhood search (-based) algorithm for the location-routing problem by Jarboui et al. [23] is an example of a metaheuristic in this sense.

In this chapter, we will use the term “metaheuristic framework” to refer to the first sense and “metaheuristic algorithm” to refer to the second sense of the word “metaheuristic.”

As mentioned, a history of any topic is not a neutral. We therefore do not attempt to hide the fact that certain ways in which the field has been progressing seem to us less useful and sometimes even harmful to the development of the field in general. For example, many of the entries that appear on the list in Fig. 1 are, in our view, not “important contributions” at all but rather marginal additions to a list of generally useless “novel” metaphor-based methods that are best forgotten as quickly as possible.

Period 0: The Pre-theoretical Period

Optimization problems are all around us. When we decide upon the road to take to work, when we put the groceries in the fridge, and when we decide which investments to make so as to maximize our expected profit, we are essentially solving an optimization problem (a shortest path problem, a packing problem, and a knapsack problem, respectively). For human beings (and many animal species), solving an optimization problem does not require any formal training, something which is immediately clear from the examples given here. The difference between exact solutions and approximate solutions and the difference between easy and hard optimization problems or between fast (polynomial) and slow (exponential) algorithms are all moot to the average problem-solver.

Indeed, the human mind seems to be formidably equipped from early childhood on to solve an incredible range of problems, many of which could be easily modeled as optimization problems. Most likely, the ability to solve optimization problems adequately and quickly is one of the most important determinants of the probability of survival in all sentient species and has therefore been favored by evolution throughout time. Clearly, the human (and animal) mind solves optimization problems *heuristically* and not exactly, i.e., the solutions produced by the brain are by no means guaranteed to be optimal. Given what we now know about exact solution procedures, this makes perfect sense. When determining the trajectory of a spear to hit a mammoth, it is much more important that this trajectory be calculated quickly rather than optimally. Given our knowledge on exact solution methods, we can now say that the calculation of the exact solution (let us say the solution that has the highest probability to hit the mammoth exactly between the eyes given its current trajectory, the terrain in front of it, its anticipated trajectory changes, the current wind direction, etc.) would almost certainly be found only *after* our target has disappeared on the horizon. Moreover, it would almost certainly require too much computing power from the brain, quickly depleting the body's scarce energy resources.

Given the diversity of problems the human mind must solve, including problems with which it has no prior experience, there is very little doubt that the human mind has the capacity (whether evolved or learned) to use *meta*-heuristic strategies. Just like the metaheuristics for optimization that form the subject of this book, such strategies are not heuristics in themselves but are used to derive heuristics from. For example, when confronted with a new problem to which a solution is not immediately obvious (e.g., determining the trajectory of a spear to hit a mammoth), the human mind will automatically attempt to find similar problems it has solved in the past (e.g., determining the trajectory of a stone to hit a bear) and attempt to derive the rules it has learned by solving this problem. This strategy is called *learning by analogy* [3]. Another example is called means-end analysis [36] and can be summarized as follows: given a current state and a goal state, choose an action that will lead to a new state that is closer to the goal state than the current state. This rule is iteratively applied until the goal state has been reached or no other state can be found closer to the goal state than the current state. Obviously, this strategy is a more general counterpart of all formal optimization heuristics that can be categorized as local search, in which a solution is iteratively improved using small, incremental operations we have come to call *moves*. The technique of *path relinking* [15], in which an incumbent solution is transformed, one move at a time, into a guiding solution, is another example of a formalized means-end analysis strategy.

Whereas heuristics (and even metaheuristics) are completely natural to us humans, exact methods seem to be a very recent invention, coinciding with the introduction of the field of Operations Research around WWII. On the other hand, even though heuristics have been applied since the first life on earth evolved, the *scientific study* of heuristics also had to wait until the twentieth century. It could be hypothesized that heuristics are so natural to us that we had to wait until a formal theory of optimization, especially of linear programming, had to be developed before anyone considered it a topic worthy of study.

Period 1: The Early Period

In 1945, immediately after WWII, the Hungarian mathematician George Pólya, then working at Stanford University, published a small volume called "How to Solve It" [30]. In his book, he argued that problems can be solved by a limited set of generally applicable strategies, most of which serve to make the problem simpler to solve. The book's focus was not on optimization problems but on the more general class of "mathematical" problems, i.e., problems that can be modeled and solved by mathematical techniques. Nevertheless, most of the solution strategies proposed in his book are equally applicable to develop optimization algorithms.

The "analogy" principle, e.g., tells the problem-solver to look for another problem that closely resembles the problem at hand and to which a solution method is known. By studying the similarities and differences between both problems, ideas can be garnered to solve the original problem. The principle of "induction" consists in solving a problem by deriving a generalization from some examples. The "auxiliary problem" idea asks whether a subproblem exists that can help to solve the overall problem. Even though it is a bit of stretch to call these principles "*meta*-heuristics," it is clear that the start of the field of OR also marks the age during which people start thinking about more general principles that are useful in the design of heuristic algorithms (or solution methods for other types of problems). A case can be made for the fact that many of Pólya's principle are still heavily used today by heuristic designers. Looking for similar problems in the literature or elsewhere, and modifying the best-known methods for them to suit the problem at hand (analogy), is an extremely common strategy to arrive at a good heuristic fast. Solving some simple examples by hand, and using the lessons learned from your own (or someone else's) perceived strategy to derive an intelligent solution strategy from (induction), is also a useful technique. Finally, decomposing a problem into smaller subproblems and developing specialized techniques for each of them (auxiliary problem) has proven to be a powerful heuristic design strategy on a large number of occasions.

What is important is that none of Pólya's strategies actually solve any problem, nor can they be called "algorithms" in themselves. Instead, they are high-level, meta-strategies that are useful to influence the way a heuristic designer thinks about a problem. In that sense at the very least, they are very like the more advanced and specialized metaheuristic frameworks that we have today.

Several very high-level algorithmic ideas also came about around this period. The fact that good solutions can be reached by a constructive procedure, for example, is one of them. A constructive algorithm is one that starts from an empty solution and iteratively adds one element at a time until a complete solution has been formed. Simple rules for selecting this element from the set of all potential elements have led to different types of algorithms. The *greedy* selection rule selects the best (value for each) element at each iteration. Kruskal's or Prim's algorithm for the minimum spanning tree problem, Dijkstra's algorithm for the shortest path problem, etc., are all examples of greedy heuristics [7]. *Regret* algorithms present a similar class of optimization procedures that select, at each iteration, the element for which *not* selecting its best value results in the highest penalty cost. Vogel's approximation method [35] for the transportation problem is a well-known example. Again, calling the greedy idea or the regret idea "metaheuristics" is a bit of a stretch, but they *are* high-level strategies, and they are *not* algorithms themselves.

Also during this period, Simon and Newell [37] see heuristics specifically as fit to solve what they call "ill-structured" problems. Contrary to well-structured problems, such problems cannot be formulated explicitly or solved by known and feasible computational techniques. Their predictions in 1958 have turned out to be slightly optimistic, but it cannot be denied that heuristics have turned out to be more flexible problem-solving strategies than exact methods.

Even though the heuristics developed in the early period were very simple, the realization that high-level strategies existed that could be used as the basis for the development of heuristics for *any* optimization problem led to insights that paved the way for more complex meta-strategies. Together with the widespread availability of computers, these developments took the field of heuristics into the next period in this history, the method-centric period.

Period 2: The Method-Centric Period

Even though the frameworks and ideas developed during what we have called the early period lacked the comprehensiveness of the later developed metaheuristic frameworks like tabu search [14], it is not too far-fetched to call them early metaheuristics. Like later metaheuristic frameworks, these methods offered – in the form of some generally applicable strategies – inspiration for the development of optimization algorithms. Of course, these principles still needed to be instantiated for each different optimization problem, but at least the process of coming up with an optimization strategy did not have to start from scratch.

Much of the work done in the early period can be characterized under the umbrella term of *artificial intelligence* because it involves mimicking human problem-solving behavior and learning lessons from this behavior on a more abstract level. Starting in the 1960s, however, an entirely different line of research into problem-solving methods came to life. These methods used an analogy with life's main problem-solving method: evolution.

Evolution by natural selection has been called "the best idea ever" [4]. No single idea explains as much as Darwin's realization that species evolve over time to adapt to their environment. The way in which this happens, by natural selection of inheritable characteristics, is both so clever and so simple it begs the question why the world needed to wait until the second half of the nineteen century before someone thought of it. Nevertheless, it took another century and the advent of the computer before researchers would become interested in simulating the process of natural evolution.

Although researchers in the late 1950s and early 1960s had developed what we would now label as evolutionary algorithms, their main aim was not to solve optimization problems but to study the phenomenon of natural evolution. The insight that the principles of natural evolution could be used to solve optimization problems in general came in the early 1960s, when researchers like Box, Friedman, and several others had independently developed algorithms inspired by evolution for function optimization and machine learning. One of the first methods to receive some share of recognition was the so-called evolution strategy (as later reported in [31]). Evolution strategy was still quite far from what we would call an evolutionary algorithm: it did not use a population or crossover. One solution (called the parent) was mutated and the best of the two solutions became the parent for the next round of mutation.

Evolutionary programming, introduced a few years later [11], represented solutions as finite-state machines but also lacked the concepts of both a population and crossover. The true start of the field of evolutionary algorithms came with the seminal work of John Holland [20], who was the first to recognize the importance of both concepts. With his schemata theory, which essentially states that high-quality schemata ("parts") of solutions will increase in frequency in successive iterations of the algorithm, Holland was also among the pioneers of theory-building in metaheuristics. The schemata theory was criticized later for its limited use and lack of general applicability, but it demonstrated that the field of metaheuristics needed not forever be devoid of theoretical underpinning.

It was perhaps the book by David Goldberg [18] (a student of John Holland) that truly sparked the evolutionary revolution. Evolutionary methods became extremely popular, journals and conferences specifically devoted to this topic sprouted, and an exponentially increasing number of papers appeared in the literature. A large number of variants were proposed, each with its own specific characteristics. Extraordinary claims were made, not necessarily grounded in empirical evidence. The quest for a generic heuristic optimization method that could solve any problem efficiently, without requiring problem-specific information, seemed finally to be on the right track.

In the 1980s, the first papers start to appear that introduced general problem-solving frameworks not based on natural evolution. One of the first used another metaphor: annealing, the controlled heating and cooling process used in metallurgy and glass production to remove stresses from the material [24]. Simulated annealing used random solution changes and "accepted" these if they improved the solution or, if they did not, with a probability inversely proportional to the solution quality decrease and proportional to an external parameter called the "temperature."

For a while, it might have seemed that the development of metaheuristics was all about finding a suitable process to imitate. The 1980s, however, also saw the development of several methods that reached back to the early period and used ideas derived from human problem-solving. One of the most powerful ideas was that solutions could be gradually improved by iteratively making small changes, called *moves*, to them. To this end, an algorithm would investigate all or some of the solutions that could be reached from the *current* solution by executing a single move. Together, these solutions form the *neighborhood* of the current solution.

Threshold accepting [9], a simple variant of simulated annealing demonstrated that a metaphor was certainly not necessary to develop a powerful general-purpose optimization framework. The great deluge method and record-to-record travel [8] differed from threshold accepting only by the way in which they accepted new solutions. Still, each of these was seen as a different method.

Perhaps the most influential of the AI-based methods was tabu search [14]. The basic premise of this framework is that a local search algorithm could be guided toward a good solution by using some of the information gathered during the search in the past. To this end, the tabu search framework defined a number of memory structures that captured aspects of the search. The most emblematic is without a doubt the *tabu list*, a list that records attributes of solutions and prohibits, for a certain number of iterations, any solutions that exhibit an attribute on the tabu list.

The same paper that introduced tabu search also coined the word "*meta*-heuristic" [14]. However, not everybody agreed with this term and a push was made to use the (more modest) term "modern heuristics" instead. Clearly, not everybody agreed that the limited set of metaheuristics proposed by the 1980s had a higher-level aspect to them. Many still viewed them essentially as (admittedly, more complicated than their simple counterparts) algorithms, i.e., unambiguous step-by-step sets of operations to be performed. Indeed, it is very common in the late 1980s for a "new metaheuristic" to be described based on a flowchart or another typical algorithmic representation. The widespread realization that metaheuristics could and should be viewed as general frameworks rather than as algorithms would come during the next period, the framework-centric period.

Interestingly, neural networks [21] were among the limited list of metaheuristics proposed by the late 1980s. These methods imitate the functioning of a brain (including neurons and synapses) and were originally proposed in the context of pattern recognition (for which they are still mostly used).

By 1995, research in metaheuristics had grown to a level that could sustain its own conference series, and thus the MIC (Metaheuristics International Conference) series was established. In the same year, the first issue of the Journal of Heuristics (http://link.springer.com/journal/10732/1/1/page/1), the only journal dedicated to publishing research in metaheuristics, was published.

Several other frameworks that had been proposed around the early 1990s also gained increasing interest during the mid-1990s. The innovation proposed in the GRASP (greedy randomized adaptive search procedure) framework was to modify a greedy heuristic by selecting at each iteration not necessarily the best element but one of the best elements randomly [10]. Similarly, ant colony optimization [5] proposed not only to mix deterministic and stochastic information but also proposed a way for solutions to exchange information.

Evolutionary algorithms invariably introduce a large amount of randomness in the search process. Some authors argued that it might be beneficial to reduce the reliance on randomness, but rather create algorithms that perform a more systematic search of the solution space. Scatter search and path relinking, both introduced in Glover [15], are the most notable examples in which the principles of evolutionary algorithms were used with the randomness removed from them [17].

By the second half of the 1990s, however, it gradually became clear that metaheuristics based on metaphors would not necessarily lead to good approaches.

The promised black box optimizers that would always "just work" and that had attracted so much attention seemed elusive. Even the theoretical studies lost some of their shine. The convergence results obtained for simulated annealing [19], because they only worked when an infinite running time was available, were not as compelling for practical situations as initially thought. Similarly, the automatic detection of good building blocks by genetic algorithms only really worked if such building blocks actually existed and if they were not continually being destroyed by the crossover and mutation operators operating on the solutions. Even though the early metaheuristic frameworks offered some compelling ideas, they did not remove the need for an experienced heuristic designer. The advent of metaheuristics had not changed the simple fact that a metaheuristic that extensively exploited the characteristics of the optimization problem at hand would almost always be superior to one that took a black box approach, regardless of the metaheuristic framework used.

In general, researchers during the method-centric period proposed *algorithms*, i.e., formalized structures that were meant to be followed like a cookbook recipe. More often than not, the "new metaheuristic" was given a name, even when the difference between the new method and an existing method was small.

Period 3: The Framework-Centric Period

The insight that metaheuristics could be more usefully described as high-level algorithmic *frameworks*, rather than as algorithms, was a natural thing to happen. The main indicator that this mindset change was taking place – a change that has given rise to a period that we have dubbed the *framework-centric* period – is the increasing popularity of so-called "hybrid" metaheuristics during the early 2000s. Indeed, this period could by rights have been called the "hybrid metaheuristic period." Whereas earlier researchers used to restrain themselves to a single metaheuristic framework, more and more researchers around the turn of the century combined ideas from different frameworks into a single heuristic algorithm. Some combinations became more popular than others, like the use of a constructive heuristic to generate an initial solution for a local search algorithm or the use of GRASP to generate solutions that are then combined using path relinking.

One type of hybrid metaheuristic even received a distinct name: the use of local search (or any "local learning" approach) to improve solutions that are obtained by an evolutionary algorithm was called a *memetic* algorithm [29]. In 2004, the term "hybrid metaheuristic" had become common, and a new conference series with the same name was started.

The hybridization of metaheuristics, however, did not restrict itself to a combination of a metaheuristic with another metaheuristic. Opening up the individual algorithmic frameworks allowed researchers to combine a metaheuristic with any auxiliary method available. Constraint programming, linear programming, and mixed-integer programming were all used in combinations with ideas from metaheuristics. The combination of metaheuristics and exact methods was coined "matheuristics" [27] (though these methods too had many antecedents in the metaheuristics literature). In 2006, the first edition of the *Matheuristics* conference took place.

Soon after its introduction, the term "hybrid" metaheuristic would become obsolete, as researchers made a general transition from seeing metaheuristics as algorithms of which some components could be borrowed by other metaheuristics to general sets of concepts ("frameworks"). The "metaheuristic framework" concept entailed that metaheuristics were nothing more (or less) than a more or less coherent set of ideas, which could, of course, be freely combined with other ideas. Today, many researchers develop metaheuristics using their experience and knowledge about which methods will work well for certain problems and which most likely will not.

Some general patterns started to appear in the literature on which methods work well for which problems, and the community gravitated toward approaches that always delivered. For almost any variant of the vehicle routing problem, e.g., a large majority of approaches use some form of local search as their main engine, generally in a multi-neighborhood framework like variable neighborhood search [28]. The use of several different local search operators or the use of several different constructive procedures in general is now a well-regarded strategy and often used as the first choice by heuristic designers. Clearly, variable neighborhood search presented a framework within which the use of multiple neighborhoods could be captured, but many other ways of combining several local search operators in a single heuristic are possible. The use of several constructive heuristics (usually combined with several destructive operators) in a single heuristic became known under the name large neighborhood search [1, 33, 43].

Crucial in this period, which is still ongoing, is that researchers do not have to propose a "new algorithm" anymore to get their papers published. By combining the most efficient operators of existing metaheuristic frameworks, and carefully tuning the resulting heuristic, algorithms can be created that solve any real-life optimization problem efficiently. Researchers can now focus on studying a single, mundane aspect of a metaheuristic framework in detail like, e.g., its stopping rules [32].

Recently, a focus can be observed on frameworks that present a much simpler approach that – in many cases – is not much less effective. Iterated local search [25], which proposes to use a single local search operator in alternation with a single perturbation operator, has a long history and dates back to at least the early 1980s [2], yet is still as popular as ever. Its constructive counterpart, often called iterated greedy [12, 34], has recently gained a large amount of traction, despite the fact that it can be seen as a restricted form of strategic oscillation (SO), a technique introduced in Glover [13] often employed in the context of tabu search. SO offers a multitude of concepts and ideas to allow the search to move within and between regions demarcated by various boundaries within the search space, such as those defining feasibility or local optimality or thresholds for various functions (such as objective functions or sums of variables). Reference to such boundaries enriches the search process by introducing different types of moves and evaluations depending on which side of the boundary the search lies on and on whether the search is moving toward or away from the boundary. Moves leading toward or away from a boundary are joined by moves launched at the boundary and at selected distances from the boundary which involve more complex searches (e.g., by employing exchange moves in place of constructive or destructive moves). Successful applications of strategic oscillation are reported in Glover and Laguna [16], Hvattum et al. [22], Lozano et al. [26], and Corberán et al. [6]. An apparent paradox is that the simplification of a metaheuristic to a few simple rules, which may restrict both its scope and its power, seems to increase its popularity. A possible explanation is that it renders the framework more accessible to non-expert users, who – contrary to the scientific community – value robustness, ease of development, and flexibility over performance.

Traditionally, the metaheuristic community has put a heavy focus on *performance*. Research is only considered good if (and only if) it produces a heuristic algorithm that "performs" well with respect to some benchmark, such as another heuristic or a lower bound. This has been called the "up-the-wall game" (though it might also be called the "one-upmanship game"). All other contributions (e.g., heuristics that are many times simpler than the best-performing heuristic in the literature, studies on heuristics that should perform well but for some reason do not, etc.) are much more difficult to publish. However, several researchers have pointed out the adverse effects of this paradigm (which effectively reduces science to a game), and some recent contributions that go beyond the up-the-wall game demonstrate the framework-centric period is gradually transforming into the *scientific* period. In this period the study of metaheuristics will shift its focus from performance to understanding. Unfortunately, however, not all of the metaheuristic community makes the transition to the framework-centric period, and we are forced to report on a period which essentially runs in parallel with this period.

The Metaphor-Centric Period

Starting in the 1980s, a subfield has arisen of research (we hesitate to put quote marks around the term for reasons that will be explained below) that focuses on the development of new metaheuristic methods based on metaphors of natural or man-made processes. In our history, this period has not been assigned a number because it does not fit chronologically between the other periods, but rather is a side step that happened (and is still happening) in parallel to the framework-centric period.

Although metaphors had been useful in the development of early metaheuristics as a source of inspiration for the development of novel frameworks, it has always been evident to many that a metaphor is only a metaphor and always breaks down at a certain point. It is therefore useful for inspiration, but not necessarily everything about it usefully translates to a metaheuristic framework. Importantly, a metaphor is not enough to *justify* metaheuristic design choices or to create a foundation for completely new metaheuristics.

In recent years, however, a different attitude seems to have taken hold of a subfield of the metaheuristic community. The aim of the "metaphor-based" subfield seems to center entirely around the development of "novel" metaphors that can be used to motivate new metaheuristics. The list of natural and man-made processes that have inspired such metaheuristic frameworks is huge. Ants, bees, termites, bacteria, invasive weed, bats, flies, fireflies, fireworks, mine blasts, frogs, wolves, cats, cuckoos, consultants, fish, glowworms, krill, monkeys, anarchic societies, imperialist societies, league championships, clouds, dolphins, Egyptian vultures, green herons, flower pollination, roach infestations, water waves, optics, black holes, the Lorentz transformation, lightning, electromagnetism, gravity, music making, "intelligent" water drops, river formation, and many, many more, have been used as the basis of a "novel" metaheuristic technique.

Moreover, there does not seem to be any restriction on the type of process that can be translated into a metaheuristic framework. One would expect that, at the very least, the process that is to become the basis for a metaheuristic should *optimize* something (e.g., an annealing process *minimizes* the energy level, natural evolution *minimizes* the discrepancy between the characteristics of a species and the requirements of this species' environment, ants *minimize* the distance between their nest and their source of food). Nevertheless, many metaheuristic frameworks can now be found based on processes that by no stretch of imagination can be said to optimize anything, like fireworks, mine blasts, or cloud formation.

Both the causes and the consequences of this "metaphor fallacy" have been extensively dealt with in a number of other publications [38, 42] (short summary: it is not science), and this is not the place to repeat all the arguments why metaphor-based metaheuristics are a bad idea. Nevertheless, metaphor-based "novel" metaheuristics take up a (dark) page in the history of metaheuristics, a page that should be turned quickly.

Period 4: The Scientific Period?

For a long time, the field of metaheuristics has had difficulties to be taken seriously. In 1977, one of the authors of this chapter wrote "[exact] algorithms are conceived in analytic purity in the high citadels of academic research, heuristics are midwifed by expediency in the dark corners of the practitioner's lair [...] and are accorded lower status" [13]. Traditionally, the theoretical underpinning of heuristics and metaheuristics has not been on par with that of other areas in OR, more specifically exact methods. The development of heuristic optimization algorithms, whether using a metaheuristic framework or not, is guided by experience, not theory. Early attempts to firmly ground the development of metaheuristics in theory have not delivered upon their promises. Understanding the behavior of metaheuristics on a fundamental level has proven to be a difficult task, notwithstanding several noteworthy efforts (e.g., [40, 41]).

Nevertheless, it is hard to argue with success. The obvious usefulness of metaheuristics in practical optimization problems has drawn researchers to improve the frameworks and methods developed. To solve a large majority of real-life optimization problems, heuristics are and will remain the only option, whether developed using a metaheuristic framework or not. Nevertheless, there are many things that can be improved about the way the metaheuristic community operates. To list just a few:

- The establishing of adequate testing protocols, to ensure that algorithms perform as well as they are claimed to do.
- The introduction of meta-analysis (i.e., a review of a clearly formulated question that uses systematic methods to identify, select, and evaluate relevant research, as well as to collect and analyze data from relevant studies) to the field of metaheuristics (Hvattum, 2015, Personal communication).
- The requirement to disclose source code, so that researchers can check and build on each other's work without in a more efficient way, without reinventing the wheel.
- The development of powerful general-purpose heuristic solvers to decrease development time, like CPLEX or Gurobi, but then heuristically. LocalSolver seems to be on the right track.
- Supporting these general-purpose solvers, the development of a powerful and generally accepted modeling language, geared more toward the development of heuristics and less toward the MIP paradigm.
- . . .

Most importantly, the change from a performance-driven community to a community in which scientific understanding is more important will take place during the scientific period. Without doubt, this will lead to the development of even better heuristics, even more efficient, but it will also lead to heuristics that are usable outside of the developer's lab environment.

Conclusions

Describing the history of the field of metaheuristics in a few pages is not an easy undertaking, and completeness is a goal that simply cannot be achieved. In this chapter, we have attempted to clarify the evolution in this field by not focusing exclusively on important events or publications but by attempting to identify the important paradigm shifts that the field has dealt with. What is certain is that the use of metaheuristics is older, much older, than the term itself. As mentioned, our brain itself houses some powerful metaheuristics that have helped humans survive from the dawn of mankind. The scientific study of metaheuristics, however, had to wait until the second half of the previous century.

Scientific communities invariably develop a conceptual framework within which a few axioms are held to be true. This can also be said of the metaheuristic community. It is those shared truths that we have attempted to uncover in this chapter. Even though the field of metaheuristics is still young, it has already undergone several paradigm shifts that have changed the way researchers look upon the development of heuristic optimization methods.

The transition from the method-centric to the framework-centric period has been beneficial for the entire community, and there is no doubt that the transition toward the scientific period can take the field further into the right direction. Metaheuristics are a fascinating area of study with highly significant practical ramifications, and the field will certainly keep on evolving in the foreseeable future. There is no doubt that a more scientific, less dogmatic, and broader point of view can help us all in achieving our goals: the development of efficient methods to solve the most challenging and important real-life optimization problems.

References

1. Ahuja RK, Ergun Ö, Orlin JB, Punnen AP (2002) A survey of very large-scale neighborhood search techniques. Discret Appl Math 123(1):75–102
2. Baxter J (1981) Local optima avoidance in depot location. J Oper Res Soc 32:815–819
3. Carbonell JG (1983) Learning by analogy: formulating and generalizing plans from past experience. In: Machine learning. Springer Science & Business Media, Berlin/Heidelberg, pp 137–161. https://doi.org/10.1007/978-3-662-12405-5_5
4. Chu T (2014) Human purpose and transhuman potential: a cosmic vision for our future evolution. Origin Press, San Rafael. ISBN:978-1-57983-0250
5. Colorni A, Dorigo M, Maniezzo V (1992) Distributed optimization by ant colonies. In: Varela FJ, Bourgine P (eds) Proceedings of the first European conference on artificial life. MIT Press, Cambridge, pp 134–142
6. Corberán Á, Peiró J, Campos V, Glover F, Martí R (2016) Strategic oscillation for the capacitated hub location problem with modular links. J Heuristics 22(2):221–244
7. Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. MIT Press, Cambridge. ISBN:978-0-262-03384-8
8. Dueck G (1993) New optimization heuristics. J Comput Phys 104(1):86–92. https://doi.org/10.1006/jcph.1993.1010
9. Dueck G, Scheuer T (1990) Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. J Comput Phys 90(1):161–175. https://doi.org/10.1016/0021-9991(90)90201-b
10. Feo TA, Resende MGC (1995) Greedy randomized adaptive search procedures. J Glob Optim 6(2):109–133. https://doi.org/10.1007/bf01096763
11. Fogel LJ, Owens AJ, Walsh MJ (1966) Artificial intelligence through simulated evolution. Wiley, New York
12. García-Martínez C, Rodriguez FJ, Lozano M (2014) Tabu-enhanced iterated greedy algorithm: a case study in the quadratic multiple knapsack problem. Eur J Oper Res 232(3):454–463
13. Glover F (1977) Heuristics for integer programming using surrogate constraints. Decis Sci 8(1):156–166. https://doi.org/10.1111/j.1540-5915.1977.tb01074.x
14. Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13(5):533–549. https://doi.org/10.1016/0305-0548(86)90048-1
15. Glover F (1998) A template for scatter search and path relinking. In: Artificial evolution. Lecture notes in computer Science. Springer Science & Business Media, pp 1–51. https://doi.org/10.1007/bfb0026589
16. Glover F, Laguna M (1997) Tabu search. Kluwer Academic Publishers/Springer, Boston

17. Glover F, Laguna M, Martí R (2000) Fundamentals of scatter search and path relinking. Control and Cybern 29(3):653–684
18. Goldberg D (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading
19. Granville V, Krivanek M, Rasson J-P (1994) Simulated annealing: a proof of convergence. IEEE Trans Pattern Anal Mach Intell 16(6):652–656. https://doi.org/10.1109/34.295910
20. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
21. Hopfield J (1982) Neural networks and physical systems with emergent collective computational capabilities. Proc Natl Acad Sci 79(8):2254–2558
22. Hvattum LM, Løkketangen A, Glover F (2004) Adaptive memory search for boolean optimization problems. Discret Appl Math 142(1):99–109
23. Jarboui B, Derbel H, Hanafi S, Mladenović N (2013) Variable neighborhood search for location routing. Comput Oper Res 40(1):47–57. https://doi.org/10.1016/j.cor.2012.05.009
24. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680. https://doi.org/10.1126/science.220.4598.671
25. Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. Int Ser Oper Res Manag Sci 57:321–354
26. Lozano M, Glover F, García-Martínez C, Rodríguez FJ, Martí R (2014) Tabu search with strategic oscillation for the quadratic minimum spanning tree. IIE Trans 46(4):414–428
27. Maniezzo V, Stützle T, Voß S (eds) (2010) Matheuristics. Springer. https://doi.org/10.1007/978-1-4419-1306-7
28. Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100. https://doi.org/10.1016/s0305-0548(97)00031-2
29. Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts – towards memetic algorithms. Technical Report 826, Caltech Concurrent Computation Program, Pasadena
30. Polya G (2014) How to solve it: a new aspect of mathematical method. Princeton university press, Princeton
31. Rechenberg I (1989) Evolution strategy: nature's way of optimization. In: Optimization: methods and applications, possibilities and limitations. Springer Science & Business Media, Berlin/New York, pp 106–126. https://doi.org/10.1007/978-3-642-83814-9_6
32. Ribeiro CC, Rosseti I, Souza RC (2011) Effective probabilistic stopping rules for randomized metaheuristics: GRASP implementations. In: Learning and intelligent optimization. Lecture notes in computer science, vol 6683. Springer Science & Business Media, pp 146–160. https://doi.org/10.1007/978-3-642-25566-3_11
33. Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 40(4):455–472
34. Ruiz R, Stützle T (2008) An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. Eur J Oper Res 187(3):1143–1159
35. Shore HH (1970) The transportation problem and the VOGEL approximation method. Decis Sci 1(3-4):441–457. https://doi.org/10.1111/j.1540-5915.1970.tb00792.x
36. Simon HA (1996) The sciences of the artificial, 3rd edn. MIT Press, Cambridge
37. Simon HA, Newell A (1958) Heuristic problem solving: the next advance in operations research. Oper Res 6(1):1–10. https://doi.org/10.1287/opre.6.1.1
38. Sörensen K (2015) Metaheuristics–the metaphor exposed. Int Trans Oper Res 22(1):3–18. https://doi.org/10.1111/itor.12001
39. Sörensen K, Glover FW (2013) Metaheuristics. In: Gass SI, Fu MC (eds) Encyclopedia of operations research and management science, 663 3rd edn, pp 960–970, Springer, Boston, MA
40. Watson J-P, Barbulescu L, Whitley LD, Howe AE (2002) Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance. INFORMS J Comput 14(2):98–123. https://doi.org/10.1287/ijoc.14.2.98.120

41. Watson J-P, Beck JC, Howe AE, Whitley LD (2003) Problem difficulty for tabu search in job-shop scheduling. Artif Intell 143(2):189–217. https://doi.org/10.1016/s0004-3702(02)00363-6
42. Weyland D (2010) A rigorous analysis of the harmony search algorithm. Int J Appl Metaheuristic Comput 1(2):50–60. https://doi.org/10.4018/jamc.2010040104
43. Yagiura M, Iwasaki S, Ibaraki T, Glover F (2004) A very large-scale neighborhood search algorithm for the multi-resource generalized assignment problem. Discret Optim 1(1):87–98