INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 22 (2015) 3–18
DOI: 10.1111/itor.12001

Metaheuristics—the metaphor exposed

Kenneth Sörensen

University of Antwerp, Operations Research Group ANT/OR, Prinsstraat 13 – B5xx, 2000 Antwerp, Belgium
E-mail: kenneth.sorensen@ua.ac.be [Sörensen]

Received 2 November 2012; received in revised form 12 November 2012; accepted 13 November 2012

Abstract

In recent years, the field of combinatorial optimization has witnessed a true tsunami of "novel" metaheuristic methods, most of them based on a metaphor of some natural or man-made process. The behavior of virtually any species of insects, the flow of water, musicians playing together – it seems that no idea is too far-fetched to serve as inspiration to launch yet another metaheuristic. In this paper, we will argue that this line of research is threatening to lead the area of metaheuristics away from scientific rigor. We will examine the historical context that gave rise to the increasing use of metaphors as inspiration and justification for the development of new methods, discuss the reasons for the vulnerability of the metaheuristics field to this line of research, and point out its fallacies. At the same time, truly innovative research of high quality is being performed as well. We conclude the paper by discussing some of the properties of this research and by pointing out some of the most promising research avenues for the field of metaheuristics.

Keywords: optimization; combinatorial optimization; metaheuristics; heuristics

1. Introduction

Imagine the following situation. On browsing your weekly copy of *Nature*, you stumble upon an article entitled "A novel food-based theory of particle physics." In this paper, the authors claim to offer truly new insight into the working of the universe on its smallest scale. The standard theory of particle physics, it is argued, has so far been rather unsuccessful in truly explaining the nature of the universe, like any of the other theories that precede it. By relating each force and each particle (called "ingredients" in the new theory) to its culinary equivalent, the author insists, a much more powerful explanation than all previous theories combined is exposed. Quarks, for example, the particles of which protons and neutrons are composed, are called "meat," whereas leptons (such as the electron) are "vegetables." Combining "meat" with "vegetables," the new theory suggests, evidently gives rise to a "dish" (an atom). Photons, the particles that transfer the electromagnetic force, the paper claims, can be best understood in terms of the "taste" that a dish produces. Similarly, bosons, fermions, gluons, and all other elementary particles are related to some cookbook terms.

Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

The infamous Higgs boson that gives other particles their mass is, of course, the "salt" particle in the new theory, which gives other "dishes" their "flavor."

There is not much doubt that the reaction of the scientific community over this article would be one of ridicule, if not outrage. In letters to the editor (who would probably be fired for allowing this article to be published), the question would rightfully be asked whether any new contribution was made other than a re-iteration of existing knowledge. The authors would be widely criticized for their attempt to change the vocabulary of the standard theory of particle physics, a tool that has been instrumental in allowing scientists across the world to communicate unambiguously. Steps would certainly be taken for this type of "research" never to be published again in a reputable journal.

The above story may strike the reader as very unlikely, yet it is not as far-fetched as it may seem at first sight. On the contrary, in the research field of optimization using metaheuristics, contributions akin to the hypothetical paper described above are frighteningly common. For a few decades, every year has seen the publication of several papers claiming to present a "novel" method for optimization, based on a metaphor of a process that is often seemingly completely unrelated to optimization. The jumps of frogs, the refraction of light, the flowing of water to the sea, an orchestra playing, sperm cells moving to fertilize an egg, the spiraling movements of galaxies, the colonizing behavior of empires, the behavior of bats, birds, ants, bees, flies, and virtually every other species of insects – it seems that there is not a single natural or man-made process that cannot be used as a metaphor for yet another "novel" optimization method. Invariably, the authors of such papers promise that their "new" method is superior to methods previously published in the literature. Rather than being scorned for the reasons mentioned above, more often than not such papers attract an impressive follow-up literature, in which a large number of optimization problems are subjected to the "novel" method, invariably with strikingly good results.

This paper is a call for a more critical view on such methods, which we will call "novel" methods (including the quotes) in the remainder of the paper, than the standard so far. We will argue that in a majority of cases they have outlasted their usefulness, which is not only unnecessary, but also harmful to the scientific quality and the external appearance of the research field. More specifically, we will attempt in this paper to answer the following questions:

- How and why were metaphors introduced to inspire the development of metaheuristics?
- What are the main fallacies of most metaphor-based research on metaheuristics?
- Why is the field of metaheuristics so vulnerable to this type of "research"? How can these methods pass the peer-review test and why can they all present such good results?

The paper will conclude with some positive paragraphs. Notwithstanding the above-mentioned problems, there is a lot of excellent research currently being undertaken in the field of metaheuristics, as researchers are gaining ever more insight into the development of more powerful methods. The final section of this paper examines some of (the properties of) these innovative research avenues.

2. History of metaphors in metaheuristics research

Algorithms for optimization can be roughly divided into two categories: exact algorithms and heuristics. The difference between the two categories is that exact algorithms are designed in such a way that it is guaranteed that they will find the optimal solution in a finite amount of time. Heuristics do not have this guarantee, and therefore generally return solutions that are worse than optimal. (An intermediate category is the so-called approximation algorithms that guarantee that the solution they find is within a certain percentage of the optimal solution. They can be thought of both as "heuristics with a guarantee" or "not-quite-exact algorithms.") Precisely, because of their guarantee to find the optimal solution, exact algorithms not only have to locate this solution in the solution space but also have to prove that they are optimal. To do this, exact algorithms must exhaustively examine every single solution in the solution space, unless they can explicitly determine that it does not need to be examined. Although many techniques exist to do this very efficiently, eliminating large numbers of solutions at each iteration, the fact remains that a lot of interesting real-life combinatorial optimization problems are not easily tackled by exact methods. Moreover, developing an efficient exact method is a nontrivial task, even for relatively easy problems.

Because of the impracticality of exhaustive search, however cleverly done, heuristics have been developed for challenging problems throughout human history. People have always been applying (simple or complicated) rules of thumb to solve optimization problems. The advent of computers in the second half of the previous century did nothing to diminish this, as computational complexity theory had by then shown that many (*NP*-hard) optimization problems were likely to be forever intractable to exact algorithms, regardless of exponentially increasing computing power. Nevertheless, notwithstanding the fact that heuristics could achieve similar solutions in a fraction of the computing time for a large class of optimization problems, and that heuristics were the *only* way to find solutions to an even larger number of real-life optimization problems, acceptance of heuristics as a serious field of research has been slow, and research in this domain has long been frowned upon by the academic community. Even though George Pólya's influential book *How to Solve It?* (Pólya, 1945), which discussed a large number of techniques to create efficient heuristics, was over three decades old by the late seventies, a quote by Fred Glover reveals that the sentiment had not quite changed:

> "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" (Glover, 1997).

Early research on heuristics often focused on human intuition as a way of solving problems. Although slightly optimistic about the future, Simon and Newell (1958), e.g., propose that a theory of heuristic (as opposed to algorithmic or exact) problem-solving should focus on intuition, insight, and learning. Metaheuristics such as tabu search, in which some aspects of the search process are remembered and used to guide the search later on, are also built on this principle. Indeed, the very idea of the tabu list (i.e., make a list of the last changes made to a solution and make sure not to perform these changes again in later iterations) is very close to the way humans might intuitively solve a problem in which the search is likely to get stuck in a local optimum (as later terminology would call it).

Many modern (meta)heuristic algorithms similarly attempt to gain a deep insight into the structure of the problem that is to be solved. Some of the heuristic principles put forward by Pólya (1945), e.g., are actively being used by heuristics researchers today to develop effective algorithms. The principle of *analogy* tells the heuristic designer to look for analogous problems for which a solution technique is known and use this information to help solve the problem at hand. The principle of *induction* instructs to solve 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. These principles are well known to a designer of heuristics, who will look for similar problems in the literature and examine the best-known methods for them (analogy), who will try to solve some simple examples by hand and derive an intelligent solution strategy from this (induction), or who will try to decompose the problem into, e.g., a master problem and a subproblem and develop specialized techniques for both (auxiliary problem). The use of these techniques forces the heuristic designer to think about the optimization problem that is being faced and provides guidelines to come up with the most appropriate strategy.

The change in perception, which has put heuristics on equal footing with exact methods as a field of research, coincides in time with the advent of *metaheuristics*. The term was first coined in Glover (1986), but a concrete definition has been elusive. In Sörensen and Glover (in press), we have defined it as follows:

> 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.

It is perhaps unfortunate that the term "metaheuristic" has come to be used both for a general problem-independent algorithmic *framework* as well as for the specific algorithms built according to its specifications. In its general sense, however, a metaheuristic is not an algorithm, i.e., it is not a sequence of actions that needs to be followed such as a cooking recipe. Rather, it is a consistent set of ideas, concepts, and operators that can be used to design heuristic optimization algorithms. To stick with the analogy, a metaheuristic in the general sense – and it is this sense we use when we speak of metaheuristics, such as tabu search, genetic algorithms, and simulated annealing – is a cooking *style*, such as "French" or "Chinese" or "Cajun," and not a recipe such as "spaghetti carbonara à la Antonio Carluccio." In fact, it could be easily argued that the analogy does not quite hold, as most cooking styles are more general than metaheuristic frameworks. Perhaps a "general recipe" such as a sandwich or a pasta dish is a better analogy, but this just illustrates one of the main points of the paper: it is just a metaphor and metaphors, however accurate, always have their limitations.

The whole idea driving the development of metaheuristic frameworks was that a heuristic did not necessarily have to be completely problem-dependent, but that general optimization techniques could be developed that were applicable to a broad class of different problems. Rather than reinvent the wheel when developing a heuristic for each different problem, a metaheuristic offered several tools that were problem-independent and that could be used as the skeleton for the development of a heuristic. Tabu search, e.g., proposed a set of concepts that were meant to memorize parts of the trajectory that the search had taken through the solution space in the recent and not-so-recent past, so that this could be exploited in the future. Just like a person looking for water in the desert tries to avoid running in circles, tabu search offered several mechanism to avoid the search from cycling, i.e., visiting the same sequence of solutions over and over again. In a sense, many of the early metaheuristics were closely related to the way humans use their intelligence to solve problems.

From the early 1970s, however, metaheuristics began to be developed based not on insight into the problem structure or on the way in which an intelligent human would solve it, but on processes that at first sight seemed to have very little to do with optimization. The simulated annealing metaheuristic, introduced by Kirkpatrick et al. (1983) based on Metropolis et al. (1953), was one of the earliest metaphor-based metaheuristics (although the term "metaheuristic" did not exist at that time). To create a defect-free crystalline solid, such as a sheet of metal or glass from a molten substance, it needs to be cooled down in a carefully controlled process called annealing. Annealing often involves (locally) reheating the substance to relieve internal stresses and errors that may have occurred. *Simulated* annealing was innovative in using an analogy to the thermodynamic process underlying annealing: the atoms in the molten material move randomly, but as the temperature drops they attempt to settle into the lowest possible energy state (which corresponds to a flawless crystal structure). Sometimes, atoms cannot move to their optimal position because other atoms have settled in positions that prohibits this. Reheating the material provides more energy to these atoms to overcome these barriers and move into a position with lower energy. Simulated annealing mimics this process by equating solutions to certain atomic configurations and the random movements of atoms to random changes of these solutions. The total energy state of the atomic configuration is the quality of the solution. Similar to the fact that atoms can always move to positions that lower the energy, changes to the solution (later called "moves") are always accepted if they improve it. The true novelty of simulated annealing was that it allowed random moves that deteriorated the solution *with a certain probability* that decreased with the size of the deterioration and increased with an external parameter that was called the "temperature." Using the temperature parameter, the algorithm designer could influence the way in which the algorithm searched for better solutions: a higher temperature allowed more random moves, which would lead the search to investigate many different parts of the solution space. A lower temperature let the algorithm favor improving moves, thereby focusing the search on a specific part of the solution space. By setting out the evolution of the temperature parameter over time (called the "cooling schedule"), the designer of a simulated annealing algorithm could influence the effectiveness of the developed algorithm.

By far the most successful metaphor used in the development of optimization algorithms has been that of natural evolution. By equating solutions to individuals in a population competing for survival, this metaphor provided the inspiration for a wide range of computational processes that could be used to locate good solutions of an optimization problem. An "evolutionary" or "genetic" algorithm would first generate a "population" (set) of random "individuals" or "chromosomes" (solutions). Individuals would then be selected from the population favoring those who had better "fitness" (objective function value). Selected solutions would then be combined into new ones by a process called "crossover." "Mutation" (random change to a solution) would be applied to some solutions to increase the diversity of individuals in the population. Genetic algorithms introduced several innovative algorithmic operators that could be effectively used to create intelligent optimization methods. In a way, the evolutionary metaphor provided researchers with a consistent toolkit that they could mix-and-match. Although many of the components introduced by genetic algorithms were truly new, the literature on evolutionary algorithms also used different terms for several common concepts. In this way, solutions became "individuals" or "chromosomes," and moves became "mutations."

The metaheuristic concept was extremely compelling, partially at least due to the intuitive appeal of evolutionary algorithms and simulated annealing, and offered a powerful way for researchers to develop more efficient heuristics. As a result, metaheuristics became more widely accepted as a viable alternative for exact methods in some situations and as the *only* alternative in others. For a while, early theoretical results even seemed to hold the promise that a set of general-purpose methods could be created of which the performance could be proved mathematically. Laarhoven and Aarts (1987), e.g., found that – under a few relatively simple conditions – simulated annealing would always converge, i.e., it was shown to have a 100% probability of finding the optimal solution. The building block hypothesis (Goldberg et al., 1989) in the genetic algorithm literature suggested that genetic algorithms would automatically be able to locate the good parts of solutions and share them with other solutions. In the end, these good "building blocks" would become prevalent in the population and the population would automatically and always converge toward an optimal solution.

Another metaphor-based metaheuristic, ant colony optimization, was introduced in Dorigo (1992) and modeled the optimization process upon the behavior of a colony of ants searching for food. As many of the metaheuristics that preceded it, ant colony optimization introduced a consistent framework for the development of optimization algorithms. At its core, ant colony optimization uses a set of agents (called) ants that heuristically construct solutions in parallel. After each construction round, the agents exchange information on the quality of their solutions through a memory structure that holds a measure of the value of each element and is called *pheromone* (a pheromone is a chemical factor that triggers a social response in the same species). In this way, elements that appear often in good solutions obtain a higher probability of being selected for inclusion in future solutions. The imagery of "ants" solving optimization problems proved to be extremely strong, and approaches based on ant colony optimization received widespread attention, including in the popular press (Anonymous, 2010), something no other metaheuristic had achieved before. The success of ant colony optimization consequently spawned an enormous number of related methods.

By the second half of the 1990s, however, it became more 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, 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.

This did not stop people from inventing new methods based on metaphors. To quote Fred Glover (Glover and Laguna, 1997, Chapter 1) again:

> "Models of nature that are relied upon for inspiration [in creating new metaheuristics] are ubiquitous, and it is easy to conjure up examples whose metaphorical possibilities have not yet been tapped. To take an excursion in the lighter side of such possibilities (though not too far from the lanes currently traveled), we may observe that a beehive offers a

> notable example of a system that possesses problem solving abilities. Bees produce hives of exceptional quality and complexity, coordinate diverse tasks among different types of individuals, perform spatial navigation, and communicate via multiple media. (It is perhaps surprising in retrospect that the behavior of bees has not been selected as a basis for one of the 'new' problem solving methods.)"

The final sentence of this quote proved truly clairvoyant, because – as if Fred Glover's words were meant to be anything but sarcastic – in 2004 the world was introduced to the bees optimization algorithm (Nakrani and Tovey, 2004), followed a year later by the artificial bee colony algorithm (Karaboga, 2005), and yet another year later by the honey bee mating optimization algorithm (Haddad et al., 2006).

In fact, the mid-2000s turned out to be a time so plentiful of "novel" metaphors as to make the Cambrian explosion pale in comparison. Just in the area of social insects, "novel" algorithms were introduced involving ants, honey bees (Karaboga, 2005), flies (Abidin et al., 2010), fruit flies (Pan, 2011), termites (Hedayatzadeh et al., 2010), fireflies (Łukasik and Żak, 2009), glow worms (Krishnanand and Ghose, 2005), and probably some other that the author is unaware of. The differences between these various social insect algorithms proved marginal at best. Moreover, discerning the novelty present in the method, if any existed at all, was difficult to do because of the opaque insect-based vocabulary used throughout the paper. Although most social insect based metaheuristics made only a small dent in the research literature, none of them coming close to achieving the notoriety of ant colony optimization, they still managed to get published.

Perhaps because the list of remaining social insects had started shortening (very likely, some researchers were actively perusing the biology literature to check for new insect discoveries), researchers turned to ever more exotic domains for inspiration. Algorithms that have received some attention in the literature are the intelligent water drops algorithm (Shah-Hosseini, 2009) (inspired by the flow of water to the sea), cuckoo search (Yang and Deb, 2009) (inspired by cuckoos laying their eggs in other birds' nests), and harmony search (Geem et al., 2001), but these are just a few outliers among a myriad other "novel" methods that did not gain traction.

3. The fallacies of "novel" metaphor-based methods

Although simulated annealing, evolutionary algorithms, and ant colony optimization changed the vocabulary of optimization to some extent, rendering themselves less accessible to researchers in optimization, they at least added some new ideas to the repertoire of optimization methods. The more recent surge of "novel" methods, on the other hand, seems to have very little on offer besides a new way of selling existing ideas.

Although it is perhaps a little unfair to single it out, the *harmony search* metaheuristic contains all the ingredients necessary to explain what is wrong with the stream of "novel" metaheuristic methods. Introduced by Geem et al. (2001), harmony search is a metaheuristic framework based on the principle of jazz musicians playing together. It completely changes the terminology of optimization into a musical one (although it oddly retains the biology-based word "fitness" as a synonym of objective function value). It is therefore not uncommon to read sentences such as the following in a paper on harmony search:

- "The Random Selection Rate measures the probability that the proposed new value for a note is drawn from an uniform distribution in the range $[0, 2\pi)$."
- "The harmony memory is updated whenever any of the new improvised harmonies at a given iteration sounds better (under the fitness criterion) than any of the remaining harmonies from the previous iteration."

The above sentences start to make sense only when one realizes that a "harmony" is just another word for "solution," that the value of a "note" (which is also called "pitch") is the value of a "decision variable" in the solution, that "sounds better (under the fitness criterion)" means "has a better objective function value" and that the "harmony memory" is just a set of solutions, that would be called "population" in an evolutionary algorithm. Not only does this change in vocabulary make papers on harmony search difficult to read for someone with a more traditional background in optimization, the metaphor itself also seems to make very little sense on its own: a musician playing or composing music does not "draw notes from a uniform random distribution" and the concept of "harmony memory" has no equivalent in real life. In other words, the use of the harmony search metaphor leads to an algorithm that is difficult to understand and, moreover, the metaphor itself seems to offer very little in the way of a consistent set of concepts on which a sensible optimization approach can be based. Harmony search is definitely not alone in stretching its metaphor to, and sometimes far beyond, its limits. "Flies" that appear in random positions, "intelligent" water drops, "bats" that do things no real-life bat would consider, the inspiration of some metaphor-based researchers seems to extend well beyond the confines of the metaphor they supposedly employ. In such cases, one could ask what is the purpose of using a metaphor, when the metaheuristic based on it does not even remotely have to resemble the process it is modeled on. In any case, it obviously does nothing in the way of making the resulting method easier to understand.

The harmony search algorithm essentially generates a set of random initial solutions and tries to find better solutions by combining existing ones and changing the values of decision variables in some of the solutions in the set. The reader who would think that this does not differ too much from the workings of an evolutionary algorithm, would not be mistaken. In a critical paper, Weyland (2010) offers compelling evidence that the harmony search algorithm is nothing else but a special case of $(\mu + 1)$ evolution strategies, a metaheuristic belonging to the evolutionary family, which was proposed by Rechenberg (1973) just short of 30 years prior to the introduction of harmony search. The rebuttal of Geem (2010), the creator of harmony search, is less than fully convincing: "Most importantly, *when I searched Wikipedia* [emphasis added], I could not find the structure $(\mu + 1)$-ES which, the protester claimed, equals that of HS." In other words, while inadvertently admitting to being completely unaware of the literature on evolution strategies, Geem et al. (2001) have reinvented a method published almost 30 years earlier. Like so many other methods, the terminology used in the paper has completely obfuscated the commonalities with existing methods. The author leaves it to the research community to determine the link of his method with other methods, something which the research community has obviously failed to do.

Notwithstanding these problems, the harmony search metaheuristic has spawned an impressive number of papers applying this method to numerous optimization problems. In fact, Wayland's paper seems to have had very little, if any, effect on the productiveness of the harmony search community. A Google Scholar search on the term "harmony search" (including the quotes) by Weyland in 2010 yielded somewhat over 500 results. The same search executed today, only 2 years later, has this number almost 3000. Although the active promotion of harmony search by its creator has probably a lot to do with this (he maintains, e.g., an extensive website on everything harmony search), it remains remarkable that a method, the contribution of which to the field of metaheuristics can only be considered marginal, has attracted such a following.

In fact, Geem's main research area, the design of water distribution networks, seems to be particularly prone to the fallacies of metaphor-based metaheuristic design. In a review paper (De Corte and Sörensen, 2012), we list the different methods that have been applied to this challenging optimization problem. The list includes genetic algorithms, memetic algorithms, the shuffled frog leaping algorithm, cross-entropy search, differential evolution, simulated annealing, the immune algorithm, and particle swarm harmony search. The most complicated of these algorithms takes almost seven pages to explain. The algorithms are tested on a very small number of instances (three or four), all of which are very small in size, and unsurprisingly, all algorithms achieve excellent performance on this unchallenging instance set. Yet, as we show in De Corte and Sörensen (2012), none of the mentioned methods is able to outperform a simple constructive heuristic that can be explained in a single paragraph.

The development of ever more "novel" metaheuristic methods has another disadvantage, in that it distracts attention away from truly innovative ideas in the field of metaheuristics. The idea to use more than one neighborhood structure to escape from local optima (usually called variable neighborhood search, Mladenović and Hansen, 1997), the idea to introduce a controlled amount of randomness to a greedy heuristic (greedy randomized adaptive search procedure or GRASP, Resende et al., 2010), the idea to use a constructive "pilot" heuristic that calculates the cost of a solution to determine the quality of a heuristic choice (the pilot method, Duin and Voß, 1999), are all fundamental ideas. At least in some domains of the metaheuristics literature, however, they seem to have been completely overlooked in favor of the idea of modeling the search process on the behavior of a frog, a termite, or a monkey. Understandably, many newcomers to the field of heuristic optimization are drowned by the tsunami of "novel" methods, which perhaps explains why these methods seem to be more prevalent in applications that are further from the core of the optimization community (e.g., water distribution network design versus vehicle routing). One need only look at the Wikipedia page on metaheuristics to appreciate the urgency for sanity to return to this research field. If the reader is not convinced, he/she is urged to have a look at Fig. 1.

In conclusion, "novel" metaheuristics based on new metaphors should be avoided if they cannot demonstrate a contribution to the field. To stress the point: renaming existing concepts does not count as a contribution. Even though methods may be called "novel" by their originator, many present no new ideas, except for the occasional marginal variant of an already existing method. Moreover, these methods take up the space of truly innovative ideas and research, for example in the analysis of existing heuristics. Because these methods invariably change the vocabulary, they are difficult to understand. Combined with the fact that the authors of these methods usually neglect to properly position "their" method in the metaheuristics literature, such methods present a loss of time and a step backward rather than forward.

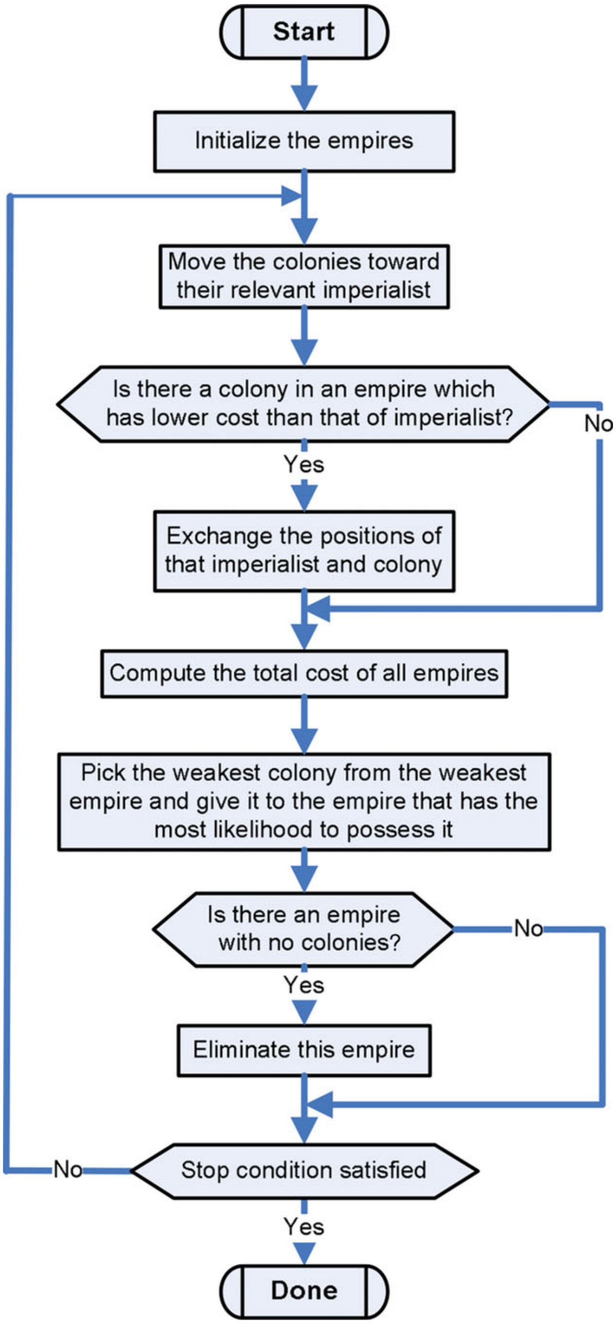

Fig. 1. Another “novel” metaheuristic: the imperialist competitive algorithm (Atashpaz-Gargari and Lucas, 2007).

4. The vulnerability of metaheuristics research

The field of optimization is perhaps unique in that natural or man-made processes completely unrelated to optimization can be used as inspiration, but other than that what has caused the research field to shoot itself in the foot by allowing the wheel to be invented over and over again? Why is the field of metaheuristics so vulnerable to this pull in an unscientific direction? The field has shifted from a situation in which metaheuristics are used as inspiration to one in which they are used as *justification*, a shift that has far-reaching negative consequences on its credibility as a research area.

The field's fetish with novelty is certainly a likely cause. Of course, researchers always aim to make significant new scientific discoveries, and the acclaim that initial metaphor-based metaheuristics such as simulated annealing, evolutionary algorithms, and ant colony optimization received, has certainly inspired many to attempt to achieve similar status. It is not surprising by the way that many of the novel methods are vigorously marketed by a single researcher, who often co-authored a sizeable fraction of all papers on "his" method. (The use of the masculine pronoun is not coincidental: unless some methods have been overlooked, and without wishing to draw any conclusions, the fact stands that it is never "her" method.) A recent marketing strategy is to maintain social media website profiles for "novel" methods. The reader might, for example, be interested to learn that the intelligent water drops algorithm officially "likes" the galaxy-based search algorithm.

A second reason for this research to pass is the fact that the research literature in metaheuristics is positively obsessed with playing the *up-the-wall game* (Burke et al., 2009). There are no rules in this game, just a goal, which is to get higher up the wall (which translates to "obtain better results") than your opponents. Science, however, is not a game. Although some competition between researchers or research groups can certainly stimulate innovation, the ultimate goal of science is to understand. True innovation in metaheuristics research therefore does not come from yet another method that performs better than its competitors, certainly if it is not well understood why exactly this method performs well.

This is not how the metaheuristics literature works, however. Papers on metaheuristics are not published if they contain an interesting insight, but if the methods they present are successful players in the up-the-wall game, and either perform better on average or beat some best-known result. How many of us have tried to publish some interesting insightful analysis of a method, only to obtain a short review report claiming that the presented results should lead to methods that are equivalent in performance to the best-known results? The result is that researchers focus all their efforts on producing algorithms that (appear to) achieve good performance on the standard set of benchmark instances, even though these are often not representative of realistic instances of the optimization problem at hand. Those authors that manage to tune their algorithm's parameters to just the right values for it to perform well on the standard benchmark sets are published, the others not. As a result, it is not uncommon for researchers to tune the parameters of their method on an instance-specific basis, or even to consider the random seed of their random number generator as a parameter and "tune" it with the other parameters.

In general, the testing of algorithms is done in an ad-hoc way, and no serious (statistical) testing is generally required for a claim of good performance to be accepted (Hooker, 1995). The lack of a generally agreed upon method to test algorithmic performance leads to a situation in which authors can present the results of their algorithm in such a way that makes the algorithm look good. Some algorithms will perform well on average, others will improve upon some best-known result, or find a larger number of best-known solutions, but the statistical validity of such conclusions is often questionable at best.

5. High-quality research in metaheuristics

Although an impression to the contrary might have been created, the purpose of this paper has certainly not been to claim that no new interesting developments in the metaheuristics literature are being made. On the contrary, notwithstanding the gloomy message of the previous sections, there exists a lot of excellent research on metaheuristics, that is taking the field forward in promising directions. This section is not by any means meant as a catalogue of or a roadmap for "good" research in metaheuristics, its only purpose is to point out some properties that the author considers to be "good research practices," and some promising areas in which a lot of research is still needed.

Clearly, any research on metaheuristics should be adequately framed in the general literature on metaheuristics and optimization in general, not just the literature on the specific method that is being developed or used. Adequately framing a method entails deconstructing it, showing which components it consists of, examining in which other metaheuristics these and similar components appear, and how these components were adapted to the specific problem that is being solved. For this to be at all possible, a new metaheuristic algorithm should be explained using the general optimization terminology (a solution should be called a "solution," for example), and not in the language of some obscure metaphor.

In general, all metaheuristic design should return to a situation in which methods are developed based on insight into the structure of the problem. Especially, research in metaheuristics should be applauded if it yields insight into the reasons why specific methods work well on specific problems. In the component-based view of metaheuristics, operators from one or a set of different metaheuristic frameworks can be combined into ever more powerful methods. While this has occasionally given rise to the development of "Frankenstein methods," i.e., overly intricate methods with many different operators, where the contribution of these operators to the final quality of the solutions found may be poorly understood (Michalewicz and Fogel, 2004), tools and techniques are available to avoid this. Using a deconstruction process (Watson et al., 2006), it is possible to gain insight into the contribution of each component, which in turn allows the metaheuristic designer to remove the parts that are not essential to the functioning of the metaheuristic. The result of such a process is, on the one hand, a "lean" metaheuristic, from which all nonessential parts have been cut, and, on the other hand, a deep insight into which components are responsible for the core optimization power of the overall method. Potentially, such analyses allow the metaheuristic designer to draw important conclusions on *why* the method works as well as it does, by proving a relationship between the properties of the metaheuristic (operator) and the structure of the problem that is being solved.

It does not make sense to ask general questions such as "Is tabu search better than simulated annealing?" as such questions would be akin to asking "Is the Chinese kitchen better than the French?" or "Is a motorcycle faster than a car?" The answer to such questions can only be "it depends." This is not to say that all metaheuristic frameworks are equally powerful, or that it is impossible to obtain meaningful insight into whether a specific metaheuristic framework is more suitable for solving a specific class of optimization problems than another. Viewing metaheuristics as sets of general concepts rather than as cookbook recipes allows a broader view on the literature and allows for the discovery of similarities between the structure and inner workings of methods that remain opaque if only the label the author of the method has chosen for it is considered. This is certainly true in the modern view of metaheuristics, in which methods may combine ideas and operators from different frameworks and the framework that is used to name the method is a matter of the author's personal taste. Should an "evolutionary algorithm" that uses eight different local search operators not be called a "variable neighborhood search" algorithm, for example?

In the component-based view of metaheuristics, some limited but valid conclusions can be drawn on which frameworks are more suitable for optimization problems. For example, the concept of using different types of local search operators is generally linked to the variable neighborhood search metaheuristic framework. However, this concept is far more common in the metaheuristics literature and appears in many other algorithms, called "variable neighborhood search." In Sörensen et al. (2008), we investigate the prevalence of the multiple neighborhood mechanism in the vehicle routing literature and find that it is a critical component of almost all methods worth their salt. Such conclusions are far more meaningful and far more useful in practice than claims that metaheuristic frameworks X or Y are particularly good at solving problem Z: metaheuristic frameworks generally consist of a large number of components that can be potentially used and any successful method will consist of a selection of components.

The pinnacle of the component-based view of metaheuristics can be found in a relatively new area that has been called *matheuristics*, and that attempts to combine exact algorithms from the mixed-integer programming paradigm, such as branch-and-bound and branch-and-cut, with (mainly local search) heuristics. The resulting method usually integrates existing exact procedures to solve subproblems and guide a higher level heuristic (Raidl and Puchinger, 2008; Dumitrescu and Stützle, 2009). Similarly, *constraint programming techniques* are being integrated with metaheuristics (Van Hentenryck and Michel, 2009) where appropriate. Matheuristics present a best-of-both-worlds approach, in which ideas and operators, from two fields traditionally separated by a very high wall, are intertwined to create ever more powerful (usually heuristic) methods. Unlike the "novel" methods from the metaphor-based literature, many of these methods truly present a direction in the metaheuristics literature that is worthwhile to pursue.

Not only do metaheuristics generally consist of different components, the contribution of which needs to be determined, most components have several parameters that need to be set. So far, the most common practice is to set these parameters manually, and without relying on a specific technique to do so. Nonetheless, the scientific value of most papers on metaheuristics would increase considerably if the step of parameter-tuning is be done in a transparent way. Moreover, this would certainly allow to create robust methods that perform well across a broad range of instances. By setting up a statistical experiment, the main and interaction effects of the different algorithmic parameters on the solution quality produced by the metaheuristic can be determined in a statistically valid way, and the optimal combination of parameter levels can be determined. Methods and guidelines to perform this step in the algorithm design are readily available (Coy et al., 2001; Adenso-Diaz and Laguna, 2006), but have not caught on in any significant way. The research software Bonesa (http://tuning.sourceforge.net) claims to be "an open source, user-friendly interface for tuning the numerical parameters of metaheuristics," but does not appear to be widely used, perhaps due to its limited documentation. *Self-adaptive* metaheuristics, which automatically tune their parameters, also present an interesting line of future research (Cotta et al., 2008). Generally, these methods are self-adaptive variants of existing metaheuristics such as GRASP (Prais and Ribeiro, 2000), evolutionary algorithms (Kramer, 2008), or tabu search (Nonobe and Ibaraki, 2001, 2002).

As mentioned, comparing different metaheuristic algorithms has so far been a largely unstructured affair, with testing procedures being determined on the fly and sometimes with a specific outcome in mind. Although several authors have developed procedures to make a statistically sound comparison (see, e.g., Barr et al., 1995; Rardin and Uzsoy, 2001), widespread acceptance of such procedures is lacking. Perhaps a set of tools is needed, i.e., a collection of statistical programs or libraries specifically designed to determine the relative quality of a set of algorithms on a set of problem instances. These should both be easy to use, and their results should be easy to interpret. Until such tools are available and a specific comparison protocol is enforced by journal editors and reviewers, the door is left open for researchers to select the method of comparison that proves the point it is intended to prove.

The good news is that the component-based view of metaheuristics, combined with a rigorous statistical protocol to deconstruct and compare algorithms, allows researchers to focus on specific aspects of a specific metaheuristic framework. Moreover, such contributions can be published even if they do not contain any "novel" method or a method that "outperforms all existing approaches." More and more papers are published that thoroughly investigate mundane aspects of a heuristic algorithm such as its termination criterion (i.e., when it should stop searching for better solutions) that have traditionally been completely neglected. To name just one out of a great many of such contributions, Ribeiro et al. (2011) discover that the solutions produced by GRASP tend to follow a normal distribution and derive effective rules to terminate the search based on the statistical properties of this distribution. Fortunately, the authors were not submerged in the "novel" method paradigm, or they might have noticed that this is exactly the way in which donkeys decide to stop mating. The result would no doubt have been a paper entitled "The donkey mating method – a novel metaheuristic for difficult optimization problems."

6. Conclusions

This paper has argued that most "novel" metaheuristics based on a new metaphor take the field of metaheuristics a step backward rather than forward. This paper therefore is a call for a more critical evaluation of such methods. In the author's view, there are more than enough "novel" methods, and there is no need for the introduction of new metaphors just for the sake of it. The time has come for consolidation, to allow the research community to discover the true mechanics underlying these "novel" methods and to concentrate on more promising research directions in the metaheuristics literature.

On the other side of the spectrum, researchers in metaheuristics are picking up ideas traditionally found in exact methods only, such as an intelligent decomposition of the problem and the use of exact methods to solve subproblems. In some problem domains such as vehicle routing or scheduling, a consensus is starting to condense on which ideas and which methods work and which do not. This is the result of many years of development of more powerful metaheuristics, combined with a careful study of the combinatorial properties of the problem. Additionally, deconstructing a method, combined with statistical testing of the components and parameters of a metaheuristic can reveal those components and parameter settings that truly contribute to the performance.

The field of metaheuristics is moving in two directions at once, but only one of these leads toward the development of more powerful methods. It is imperative that metaheuristicians keep on pushing forward in this direction, and avoid falling into the "novel" method trap. Rather than being a reason for publication, new methaphors should be exposed for what they are – a distraction from real research that is both embarrassing and detrimental to the field.

References

Abidin, Z.Z., Ngah, U.K., Arshad, M.R., Ping, O.B., 2010. A novel fly optimization algorithm for swarming application. IEEE Conference on Robotics, Automation and Mechatronics, 27–28 June 2010, Singapore, pp. 425–428.

Adenso-Diaz, B., Laguna, M., 2006. Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research* 54, 1, 99–114.

Anonymous, 2010. Riders on a swarm. *The Economist*.

Atashpaz-Gargari, E., Lucas, C., 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. IEEE Congress on Evolutionary Computation, 25–28 September 2007, Singapore, pp. 4661–4667.

Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., Stewart, W.R., 1995. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics* 1, 1, 9–32.

Burke, E.K., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., Vazquez-Rodriguez, J.A., 2009. Towards the decathlon challenge of search heuristics. Workshop on Automated Heuristic Design – In conjunction with the Genetic and Evolutionary Computation Conference (GECCO-2009), 8–12 July 2009, Montreal, Canada.

Cotta, C., Sevaux, M., Sörensen, K., 2008. *Adaptive and Multilevel Metaheuristics*. Springer-Verlag, Berlin.

Coy, S.P., Golden, B.L., Runger, G.C., Wasil, E.A., 2001. Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics* 7, 1, 77–97.

De Corte, A., Sörensen, K., 2012. Optimisation of water distribution network design: a critical review. Working paper 2012/016, University of Antwerp, Faculty of Applied Economics, Antwerp, Belgium.

Dorigo, M., 1992. Optimization, learning and natural algorithms. PhD thesis, Politecnico di Milano, Milan, Italy.

Duin, C., Voß, S., 1999. The Pilot method: A strategy for heuristic repetition with application to the Steiner problem in graphs. *Networks* 34, 3, 181–191.

Dumitrescu, I., Stützle, T., 2009. Usage of exact algorithms to enhance stochastic local search algorithms. In Maniezzo, V., Stützle, T., and Voß, S. (eds.) *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, Vol. 10 of *Annals of Information Systems*. Springer, Berlin.

Geem, Z.W., 2010. Survival of the fittest algorithm or the novelest algorithm?: The existence reason of the harmony search algorithm. *International Journal of Applied Metaheuristic Computing* 1, 4, 76–80.

Geem, Z.W., Kim, J.H., Loganathan, G.V., 2001. A new heuristic optimization algorithm: harmony search. *Simulation* 76, 2, 60–68.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 13, 533–549.

Glover, F., 1997. Heuristics for integer programming using surrogate constraints. *Decision Sciences* 8, 1, 156–166.

Glover, F., Laguna, M., 1997. *Tabu Search*. Kluwer Academic, Boston, MA.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

Haddad, O.B., Afshar, A., Marino, M.A., 2006. Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization. *Water Resources Management* 20, 5, 661–680.

Hedayatzadeh, R., Akhavan Salmassi, F., Keshtgari, M., Akbari, R., Ziarati, K., 2010. Termite colony optimization: a novel approach for optimizing continuous problems. 18th Iranian Conference on Electrical Engineering (ICEE), IEEE, pp. 553–558.

Hooker, J.N., 1995. Testing heuristics: We have it all wrong. *Journal of Heuristics* 1, 1, 33–42.

Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University Press, Erciyes.

Kirkpatrick, S., Gelatt, C.D. Jr., Vecchi, M.P. 1983. Optimization by simulated annealing. *Science* 220, 4598, 671–680.

Kramer, O., 2008. *Self-adaptive Heuristics for Evolutionary Computation*. Springer-Verlag, Berlin.

Krishnanand, K.N., Ghose, D. 2005. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. *Swarm Intelligence Symposium*, IEEE, New York, pp. 84–91.

Laarhoven, P.J.M., Aarts, E.H.L. 1987. *Simulated Annealing: Theory and Applications*. Springer, Berlin.

Łukasik, S., Żak, S., 2009. Firefly algorithm for continuous constrained optimization tasks. *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, Springer, Berlin, pp. 97–106.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., et al., 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 6, 1087.

Michalewicz, Z., Fogel. D.B., 2004. *How to Solve It: Modern Heuristics*. Springer-Verlag, New York.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. *Computers & Operations Research* 24, 11, 1097–1100.

Nakrani, S., Tovey, C., 2004. On honey bees and dynamic server allocation in internet hosting centers. *Adaptive Behavior* 12, 3-4, 223–240.

Nonobe, K., Ibaraki, T., 2001. An improved tabu search method for the weighted constraint satisfaction problem. *INFOR* 39, 2, 131–151.

Nonobe, K., Ibaraki, T., 2002. Formulation and tabu search algorithm for the resource constrained project scheduling problem. In Ribeiro, C.C., Hansen, P. (eds.), *Essays and Surveys in Metaheuristics*, Kluwer Academic, Boston, MA, pp. 557–588.

Pan, W.T., 2011. A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems* 26, 69–74.

Pólya, G. 1945. *How to Solve It: A New Aspect of Mathematical Model*. Princeton University Press, Princeton, NJ.

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.

Raidl, G.R., Puchinger, J., 2008. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In Blum, C., Blesa Aguilera, M.J., Roli, A., Sampels, M. (eds.), *Hybrid Metaheuristics: An Emerging Approach to Optimization, Studies in Computational Intelligence*, Vol. 114, Springer, Berlin.

Rardin, R.L., Uzsoy, R., 2001. Experimental evaluation of heuristic optimization algorithms: a tutorial. *Journal of Heuristics* 7, 3, 261–304.

Rechenberg, I., 1973. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart.

Resende, M.G.C., Martí, R., Gallego, M., Duarte, A., 2010. GRASP and path relinking for the max-min diversity problem. *Computers and Operations Research* 37, 498–508.

Ribeiro, C., Rosseti, I., Souza, R., 2011. Effective probabilistic stopping rules for randomized metaheuristics: GRASP implementations. *Learning and Intelligent Optimization*, conference, Rome, Italy, January 17–21, 2011, pp. 146–160.

Shah-Hosseini, H., 2009. The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation* 1, 1, 71–79.

Simon, H.A., Newell, A., 1958. Heuristic problem solving: The next advance in operations research. *Operations Research* 6, 1, 1–10.

Sörensen, K., Glover, F., 2013. Metaheuristics. In Gass, S., Fu, M. (eds.), *Encyclopedia of Operations Research and Management Science*, 3rd ed., Springer, London.

Sörensen, K., Sevaux, M., Schittekat, P., 2008. "Multiple neighbourhood search" in commercial VRP packages: evolving towards self-adaptive methods, *Adaptive and Multilevel Metaheuristics*, Vol. 136, Springer, London, pp. 239–253.

Van Hentenryck, P., Michel, L., 2009. *Constraint-Based Local Search*. MIT Press, Cambridge, MA.

Watson, J.P., Howe, A.E., Darrell Whitley, L., 2006. Deconstructing Nowicki and Smutnicki's i-TSAB tabu search algorithm for the job-shop scheduling problem. *Computers & Operations Research* 33, 9, 2623–2644.

Weyland, D., 2010. A rigorous analysis of the harmony search algorithm – How the research community can be misled by a "novel" methodology. *International Journal of Applied Metaheuristic Computing* 1–2, 50–60.

Yang, X.S., Deb, S., 2009. Cuckoo search via lévy flights. World Congress on Nature & Biologically Inspired Computing, IEEE, New York, pp. 210–214.