

Sarhani, Malek; Voß, Stefan; Jovanovic, Raka

Article — Published Version

Initialization of metaheuristics: comprehensive review, critical analysis, and research directions

International Transactions in Operational Research

Provided in Cooperation with:

John Wiley & Sons

Suggested Citation: Sarhani, Malek; Voß, Stefan; Jovanovic, Raka (2022) : Initialization of metaheuristics: comprehensive review, critical analysis, and research directions, International Transactions in Operational Research, ISSN 1475-3995, Wiley, Hoboken, NJ, Vol. 30, Iss. 6, pp. 3361-3397,
<https://doi.org/10.1111/itor.13237>

This Version is available at:

<https://hdl.handle.net/10419/287957>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Initialization of metaheuristics: comprehensive review, critical analysis, and research directions

Malek Sarhani^{a,c} , Stefan Voß^{a,*}  and Raka Jovanovic^b

^a*Institute of Information Systems, University of Hamburg, Von-Melle-Park 5, Hamburg 20146, Germany*

^b*Institute of Physics, University of Belgrade, Pregrevica 118, Belgrade 11080, Serbia*

^c*School of Business Administration, Al Akhawayn University in Ifrane, Avenue Hassan II, P.O. Box 104, Ifrane 53000, Morocco*

E-mail: malek.sarhani@gmail.com [Sarhani]; stefan.voss@uni-hamburg.de [Voß]; rakaj@ipb.ac.rs [Jovanovic]

Received 26 January 2022; received in revised form 31 October 2022; accepted 5 November 2022

Abstract

Initialization of metaheuristics is a crucial topic that lacks a comprehensive and systematic review of the state of the art. Providing such a review requires in-depth study and knowledge of the advances and challenges in the broader field of metaheuristics, especially with regard to diversification strategies, in order to assess the proposed methods and provide insights for initialization. Motivated by the aforementioned research gap, we provide a related review and begin by describing the main metaheuristic methods and their diversification mechanisms. Then, we review and analyze the existing initialization approaches while proposing a new categorization of them. Next, we focus on challenging optimization problems, namely constrained and discrete optimization. Lastly, we give insights on the initialization of local search approaches.

Keywords: metaheuristics; initialization; evolutionary algorithms; swarm intelligence; local search

1. Introduction

Optimization is a powerful technique for obtaining suitable solutions for various engineering and planning problems. Over the past three decades, a wide range of metaheuristic algorithms have been developed by various researchers in order to solve complex optimization problems from engineering, business, etc. In fact, while gradient-based classical optimization algorithms are usually unable to deal with nonlinear, nonconvex as well as multimodal problems, metaheuristics can handle these types of problems more effectively. They are also useful when the methods that find optimal solutions cannot be applied due to their computational cost (Bennis and Bhattacharjya, 2020). These metaheuristics are usually iterative optimization algorithms, which most often share an algorithmic step, which is solution(s) initialization. It is widely accepted in the research community that

*Corresponding author.

© 2022 The Authors.

International Transactions in Operational Research published by John Wiley & Sons Ltd on behalf of International Federation of Operational Research Societies.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

this step plays an important role in the optimization process, because all the solutions generated thereafter depend, to a certain extent, on their preceding solutions and, eventually, on the initial solution or on the initial population of solutions. Nevertheless, measuring the degree of dependency is not straightforward. Moreover, when addressing the issue, it is crucial to illustrate the purpose of the approach, and whether it aims to improve quality and/or diversity. Diversity refers to the degree to which solutions effectively explore the entire search space, while quality often refers to the proximity of the generated solutions to a global optimum. For simplicity, we adopt this definition for quality and we will discuss it later. The process of searching for improving quality within certain limits of the search space is often known as intensification (also called exploitation) and enhancing diversity is often labeled as diversification (or exploration), where the search attempts to explore as yet unvisited parts of the solution space. The balance between both objectives is the most challenging issue when designing and improving metaheuristics. Nevertheless, the achievement of a balance between them is not fully explored, due to the difficulty of carrying out theoretical and customized work in this regard (e.g., Chen et al., 2009). Our goal in this paper is to bridge the gap between the progress in metaheuristics initialization and the aforementioned trade-off.

By analyzing the literature on metaheuristics initialization, we note that little attention has been paid to the synthesis and analysis of the initialization methods in a comprehensive and systematic way. To our knowledge, the most recent survey on the initialization of metaheuristics was proposed in Kazimipour et al. (2014), which is mainly devoted to population-based metaheuristics. The main contribution of that paper is to categorize them according to three different dimensions, which are randomness, compositionality, and generality. Our contribution compared to Kazimipour et al. (2014) is to provide a critical analysis of the proposed approaches and to highlight the main related topics that are crucial to tackle this issue, while providing insights into it. Moreover, we do not restrict ourselves to population-based approaches. In fact, to properly address these issues, it is necessary to consider the main metaheuristics challenges, especially the mentioned dilemma. Despite this, we note that many papers referenced in Kazimipour et al. (2014) do not provide real evidence why their initialization approaches work well and under which conditions.

The main purpose of this paper is to present a comprehensive review of the state of the art of metaheuristics initialization that situates it within the current progress and findings in the field. In other words, the field of metaheuristics has exerted a great development, particularly with respect to the exploration–exploitation balance management. In this paper, we show how it has been exploited either directly (i.e., for the papers that focus primarily on initialization) or indirectly (i.e., for papers that offer generic ideas beneficial for initialization). We also give ideas in this respect and highlight open challenges that have to be addressed for an effective and efficient initialization of metaheuristics.

In this paper, we are interested more in generic initialization approaches and not so much in problem-specific approaches. In fact, when detailed knowledge about the problem to be solved is available, a common way to generate an initial solution is to use a constructive heuristic. For instance, Ho and Gendreau (2006) adopted a stochastic heuristic to generate solutions for a routing problem and Sapkal and Laha (2011) proposed a constructive heuristic to obtain initial solutions to a scheduling problem. On the other hand, supervised machine learning (ML) approaches (Jo, 2021) can be adopted to learn from previous experiences (i.e., results of previous executions), and then to predict the best initialization approach, based on the best predicted performance, for each problem instance (Birattari, 2006). Such approaches could be beneficial when historical information about

the problem being solved is available. In this paper, our interest is in solving problems when no (referenced) previous knowledge is available.

In Kazimipour et al. (2014), the referenced papers are notably divided into two categories, namely noncompositional and compositional approaches. The latter methods require a preliminary step (e.g., clustering) before defining the initial values. In this paper, we rather propose a categorization based on the ideas behind the approaches and their technical concepts. This is the subject of Sections 3–5. Most of the literature in metaheuristics initialization can be included in this categorization. In addition, we are interested in particular problems and types of algorithms that are often not directly addressed in the initialization literature. This is the case with constrained problems, and single-solution algorithms, which are discussed in Sections 6 and 7, respectively.

One of the main issues in metaheuristic's research of the last decade was the antipodal discussion of possibly new approaches using certain metaphors (see, e.g., Sörensen, 2015) and the question about their real novelty (see, e.g., Camacho Villalón et al., 2020; de Armas et al., 2022; Camacho-Villalón et al., 2022). One of the possible ways to enhance the knowledge about metaheuristics is to explore their components to discriminate novelty from repetition. In this sense, we add toward the literature investigating specific components regarding the initialization of metaheuristics. In this paper, we mainly focus on papers published in well-known journals and conferences. The main repositories used are, for example, ScienceDirect, IEEE Xplore, ACM, and Springer. In addition, we highlight other articles that promote new ideas while noting the limits of their findings. This is done by searching the above-mentioned repositories plus some forward and backward search as well as Google scholar entries under “metaheuristic” and “initialization.”

The remainder of this paper is organized as follows. In the next section, we outline the background of metaheuristics. In Section 3, we focus on randomized approaches. Section 4 is dedicated to the learning concept. Section 5 is devoted to statistical and decomposition methods. Section 6 aims to shed light on the specificity with respect to the constraints and to provide suggestions for combinatorial optimization. The purpose of Section 7 is to project the problem onto single-solution methods. Finally, in Section 8, we summarize the findings of the paper and highlight promising research directions.

2. Metaheuristic methodology

The aim of this section is to provide a preamble that introduces the necessary background needed to situate, understand, and analyze the literature on metaheuristic initialization. More specifically, we begin by showing the different optimization problems and metaheuristic concepts, then we introduce the main metaheuristic methods while describing how they handle the diversification–intensification (exploration–exploitation) dilemma. Then, we show and analyze their basic approaches for initialization.

2.1. Optimization problems

Optimization problems are often differentiated according to the constraints or the nature of the variables. Regarding the former, for unconstrained optimization problems, the set of constraints

is empty. Otherwise, the problem is labeled as a constrained optimization problem. Regarding the latter, the solutions are encoded with real-valued variables for continuous optimization, and are encoded with discrete and binary variables for discrete and binary optimization problems, respectively. In this review, we start by exploring generic cases that are applicable or easily customized to different types of problems. In general, the above-mentioned approaches do not face any specific limitation for unconstrained continuous optimization problems, and the other cases are often more complicated and some of them could not be effectively personalized. Thereby, in Section 6, we draw attention to the particularity with respect to constraints and the nature of the variables, respectively. Concerning the nature of the variables, we are interested in combinatorial optimization, since it is the usual and most studied case of discrete and binary optimization in the literature. In this type of problem, the constraint of the binary variables is added to the other constraints related to the problem.

2.2. Metaheuristic concepts

Before diving into the issue of metaheuristic initialization, we introduce the concept of metaheuristics and the aforementioned dilemma. First of all, we note that a number of definitions have been proposed. According to Voß et al. (1999), a metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. Voß and Woodruff (2003) distinguish between a guiding process and an application process. The first process decides on possible (local) moves and forwards its decision to the second process, which then executes the chosen move. According to Sörensen (2015), metaheuristics are high-level problem-independent algorithmic frameworks that provide a set of guidelines or strategies to develop heuristic optimization algorithms. We can conclude from these and other definitions that metaheuristics are problem-independent and aim to guide the search process in an intelligent way.

More practically, it is widely admitted that the diversification–intensification (or exploration–exploitation) trade-off is a crucial aspect that has to be addressed by different metaheuristics. In fact, they are two fundamental components of different metaheuristics (Glover and Samorani, 2019) and the success of most of them depends on the proper handling of this compromise. The diversification–intensification (exploration–exploitation) dilemma is a crucial issue in the field of metaheuristics, which has been associated with it since its introduction and has been studied from its beginnings (Glover and Laguna, 1997). The former is responsible for the detection of the most promising regions in the search space, while the latter promotes convergence of solutions. In general, during an intensification stage, the search concentrates on the examination of the neighbors of selected solutions. The diversification stage, on the other hand, encourages the search process to examine unvisited regions and to generate solutions that differ in significant ways from those seen before. We refer to Blum and Roli (2003) for some definitions and extrapolations of this dilemma, which drives the various well-known and effective metaheuristics. We note that in the literature, the terms diversification and intensification (respectively, exploration and exploitation) are often associated with single-solution methods (respectively, population-based methods). Hence, we will use them in this paper based on this differentiation. Next, we expose the metaheuristic methods.

2.3. Metaheuristic approaches

First of all, we discuss the extent to which we can design generic initialization approaches, given that the number of metaheuristics is countless. In fact, one can guess that this could be a main reason for the absence of a unified review of the addressed topic. But, we note that the novelty of several metaheuristics introduced over the last decade has been questioned (or even denied) in numerous papers (e.g., Sörensen, 2015; Weyland, 2015). Such claims have also been supported by early pioneer researchers in the field in their recent publications (e.g., Sörensen and Glover, 2013; Camacho Villalón et al., 2020; de Armas et al., 2022). In particular, de Armas et al. (2022) pointed out that several different metaheuristics share similar components and most of them do not contain any novelty at all. In this part, we take a look at classical metaheuristics and aim to categorize them and highlight the different inspirations and philosophies beyond them, and how they deal with the dilemma.

Metaheuristic approaches could be divided in several ways (see, e.g., Caserta and Voß, 2009, as well as the template concept in Greistorfer and Voß, 2005). The best-known and most common approach is to divide them into single-solution metaheuristics and population-based metaheuristics. The former focus on modifying and improving a single candidate solution, while the latter maintain and improve multiple candidate solutions. Typical examples of the former are simulated annealing (SA), tabu search (TS), iterated local search (ILS), and the greedy randomized adaptive search procedure (GRASP). Examples of the latter are genetic algorithms (GA), particle swarm optimization (PSO), differential evolution (DE), ant colony optimization (ACO), and scatter search (SC).

On the one hand, single-solution metaheuristics are often based on a local search procedure. The aim of the underlined metaheuristics is to provide intelligent mechanisms to exploit local search moves (Caserta and Voß, 2009), as outlined next.

TS is a single-solution metaheuristic that takes a potential solution to a problem and checks its immediate neighbors. Its main contribution compared to classical local searches is that it considers the history of the search by introducing the concept of memory in order to diversify the solutions (Voß, 1993; Glover and Laguna, 1997). SA is based on another concept that mimics the cooling of metals by allowing to accept worse solutions based on a probabilistic factor, named temperature. This factor enables to control the aforementioned dilemma. ILS generates the starting solution for the next iteration by perturbing the local optimum found by adopting a local search, to enable a diversification of the search. GRASP is another multistart metaheuristic (the concept is described in Section 7) for combinatorial optimization problems, in which each iteration essentially consists of two phases: construction and improvement (or local search). The dilemma is managed by balancing both randomization and greedy parts of the method (Resende and Ribeiro, 2018). We note that in this paper, we also refer to single-solution metaheuristics by adopting the terms local searches or local search metaheuristics as widely embraced in the literature.

On the other hand, population-based metaheuristics are most often inspired by natural phenomena. Bio-inspired population-based metaheuristics are often divided into evolutionary algorithms and swarm intelligence algorithms. GA and DE are the most popular algorithms belonging to the first category, and PSO and ACO are the best-known examples of the second category. It is notable that Molina et al. (2020) emphasized that most of the bio-inspired algorithms proposed recently

Table 1
Metaheuristics and diversification/intensification

Algorithm	Diversification	Intensification	Control
TS	Prohibition list	Local search	Tabu list
SA	Temperature	Local search	Cooling rate
ILS	Restart	Local search	Perturbation
GRASP	Randomization	Local search	Restricted candidate list
GA	Mutation	Crossover	Rates
DE	//	//	Adaptation
PSO	Personal best	Global best	Inertia weight
ACO	Randomization	Neighbor	Pheromone
SC	Diversification	Improvement	Reference set

ACO, ant colony optimization; DE, differential evolution; GA, genetic algorithm; GRASP, greedy randomized adaptive search procedure; ILS, iterated local search; PSO, particle swarm optimization; SA, simulated annealing; SC, scatter search; TS, tabu search.

could be derived from PSO, DE, GA, and ACO (the rest could be derived from the artificial bee colony method; cf. Karaboga et al., 2012), which are the methods considered in this part. (This is also in line with the facts exposed in Camacho Villalón et al., 2020; de Armas et al., 2022.)

On the one hand, GA is a popular metaheuristic that is based on the idea of individuals competing for survival. At each iteration, GA selects and generates solutions based on three main operators, namely selection, mutation, and crossover. The dilemma management depends on mutation and crossover operators. GA is a typical case of evolutionary algorithms. DE, which belongs to the same category, adopts similar exploration and exploitation mechanisms. We note that other exploration approaches for evolutionary algorithms have been proposed (e.g., Oliveto et al., 2019). Additionally, several extensions of GA have been proposed to better handle the dilemma, such as the biased random-key GA (Gonçalves and Resende, 2010).

On the other hand, within the umbrella of swarm intelligence, PSO is the most adopted for continuous optimization and ACO is best known for combinatorial optimization. The first guides the search based on the location of high-quality solutions and the second focuses on the parts of solutions that frequently appear in high-quality solutions. They both consist of the interaction between a number of agents that share information about the solution space. In PSO, each agent (particle) is a candidate solution to the problem, and is represented by velocity, a position in the search space, and has a memory that helps remembering its previous positions. For the typical PSO, the dilemma is controlled by the values of the parameters (Shi and Eberhart, 1999). ACO is an algorithm for finding paths based on the behavior of ants foraging (Dorigo and Gambardella, 1997). The management of the balance depends on the values of the pheromone trails. SC draws its foundations from previous strategies for combining decision rules and constraints. Equilibrium is achieved in a manner similar to evolutionary algorithms.

In Table 1, we summarize the diversification and intensification mechanisms for the different metaheuristics as well as the control mechanisms for each algorithm. Table 1 does not provide complete information but just a simplistic overview.

We note that all these algorithms aim to balance the aforementioned dilemma through parameter tuning. The algorithms usually allow for immense diversification at the start of the search, and

tend to intensify in specific regions throughout the later stages of the search. This is the typical case of setting the PSO inertia weight parameter (Shi and Eberhart, 1999) and the SA temperature parameter (Henderson et al., 2003), and is widely used for the GA and DE mutation operator (e.g., Hassanat et al., 2019). Moreover, several method improvements aimed at strengthening this feature have been proposed. For example, clustering and cooperative approaches (Section 5) frequently aim to explore different regions of the search space in the first steps (diversification) and then to exploit them in the last stages (intensification).

To handle the dilemma in the most appropriate way, we can observe from the literature that much of the interest has been devoted to tuning the parameters of the algorithms. Conversely, less attention has been paid to relating the initialization phase to the dilemma. Nevertheless, we can argue that initialization can help achieving the same goal of the tuning process. In the remainder of the paper, we address this issue and show how to diversify the solutions in the same spirit of tuning advances.

2.4. Metaheuristic initialization

First of all, as stated earlier, a crucial question that needs to be addressed when proposing initialization approaches is whether they aim to diversify the initial solutions or have good objective values for them. As illustrated in the previous part, diversification is the most sought-after feature in the early stages of the different algorithms. Moreover, as indicated in Li et al. (2020), the average distance between the initial population of solutions and the real optimal solution does not have a significant correlation with the quality of the final solution for the algorithms.

Second, in the absence of prior information on the solution space, the generation of random numbers is the most classical approach for initialization. A main difference between the two above-mentioned approaches is that population-based metaheuristics generate initial solutions simultaneously, while single-solution metaheuristics generate the solutions sequentially.

Third, two issues that arise are to which extent we can design generic initialization approaches that are independent of the nature of the algorithms, and what is the impact of the initialization step on different algorithms. To address the first issue, we first note that some studies outlined similarities among these algorithms. For example, Taillard et al. (2001) affirmed that procedures such as SA, TS, or GA can be explained by means of an established structure named adaptive memory programming. Another structure, based on a pool template, has been proposed in Greistorfer and Voß (2005) to unify these methods. We can then conclude that these algorithms share some features and, as an initial guess, we can expect that the impact of initialization methods on them is correlated.

With respect to the second issue, we can observe that studying the impact of the initialization phase on metaheuristics has not been exhaustively reviewed. Nonetheless, some works have been proposed in this regard. Li et al. (2020) studied the sensitivity of five algorithms, including PSO, GA, and DE, to the initialization phase. The authors pointed out that some algorithms are more sensitive to initialization than others. More specifically, the authors compared several different initialization methods, based on different probability distributions and found, for example, that PSO performs differently for different initialization methods while DE is more robust when it comes to the initialization method. Moreover, the paper reports the impact of studying the quality and

diversity of the solutions on the performance of the algorithms. More details on the sensitivity issue and on the paper's findings will be provided later.

Other papers are interested in specific algorithms. For evolutionary algorithms, an example of a work that includes this issue for the case of GA is Paul et al. (2015). The authors evaluated seeding techniques using performance criteria such as the computation time, error rate, average convergence, and convergence diversity. Oman and Cunningham (2001) analyzed the impact of seeding the initial GA solutions with good ones (in terms of their objective value) for the traveling salesman problem (TSP) and the job-shop scheduling problem. For swarm intelligence, for instance, Helwig and Wanka (2008) studied the impact of different velocity initialization strategies on the population behavior of PSO. In particular, the authors showed that uniform velocity initialization causes many particles to leave the feasible search space.

Birattari et al. (2007) stressed the importance of this topic for ACO. In case of ACO, the initialization is related to the initial values of the pheromone trail. In that sense, no specific solutions need to be provided. That is, the authors noted that the initialization of the pheromone plays a critical role. In particular, for the algorithm to be invariant, the pheromone must be initialized invariantly. Several other studies of different initialization techniques have been proposed and they will be reviewed in the next corresponding sections. Our aim, in addition to the review, is to derive meaningful conclusions.

At the end, we note that almost all the papers that have directly addressed the issue of initialization are concerned with population-based algorithms, hence our interest in the next three sections is mainly devoted to them.

3. Randomization

Randomization may be used in different settings generating random solutions as incumbents or generating sequences of random solutions.

3.1. Generating random solutions

In the absence of prior information about the solution space, random number generation is the classic approach used by most population-based metaheuristics to generate an initial population of solutions. In practice, most method implementations use pseudo-random number generators (PRNGs) to generate initial solutions for them. Although PRNGs are widely adopted in a number of areas (e.g., cryptography), only a few works are interested in studying and improving PRNGs for metaheuristics. For instance, Ma and Vandenbosch (2012) studied the impact of random generators, such as standard Java and Matlab generators, on the initialization of positions and velocities of particles (for a PSO) as well as on their update. Moreover, Krömer et al. (2014) proposed an empirical comparison of the behavior of three common metaheuristics, namely GA, PSO, and DE, based on various PRNG methods. In particular, the authors stressed the importance of studying the properties of the distributions used in PRNGs. In fact, this issue is not sufficiently addressed for the initialization of metaheuristics. According to Alhalabi and Dragoi (2017), the performance of the algorithm varies considerably depending on the used distribution. In particular, the best

results were obtained using Binomial and Weibull distribution and the worst were obtained using the Beta distribution. (More information on these distributions can be found in Mun, 2015). In that paper, the influence of different distributions adopted in PRNGs, either continuous or binary, was compared for the differential search (DS) algorithm, which is a newly proposed metaheuristic. However, the authors did not provide evidence for the existence of a significant difference between these distributions. We, therefore, cannot generalize these results because of their randomness.

A typical approach to generate the random population is to use a uniform distribution as in Equation (1):

$$x_i^j = \underline{x}_j + rand_j(0, 1)(\bar{x}_j - \underline{x}_j), \quad (1)$$

where $rand_j(0, 1)$ is a uniform random number within $[0, 1]$ and \bar{x}_j and \underline{x}_j are the upper and lower bounds of the j th dimension of the problem, respectively.

Nevertheless, the most adopted PRNG is the Mersenne twister from Matsumoto and Nishimura (1998). The name of this PRNG, which is included in most software, comes from the fact that a Mersenne prime is chosen to be its period length. This approach has shown better performance than other PRNGs. However, its main problem, as with other PRNGs, is concerning its diversification capability. Indeed, randomization does not generally cover the search space in an optimal way because there is no guarantee of a significant differentiation of the solutions. For example, it is possible that many generated vectors have similar values or closer ones. For example, for binary problems, it is possible that almost all vectors are composed of $n/2$ zeros and $n/2$ ones. Such an initial population is not informative about the whole search space. In addition, it can cause a premature convergence of algorithms such as PSO (Trelea, 2003) or if a local search is used subsequently. That is, if all the solutions are close and one corresponds to a local optimum, then all the solutions will converge toward the local optimum.

Therefore, another exciting approach to build the initial population is to generate the individuals in sequence. Maaranen et al. (2004) proposed a quasi-random sequence of points, instead of pseudo-random numbers, for the same purpose. Another sequence for a GA has been proposed in Kimura and Matsumura (2005). The main advantage of these approaches is that individuals can take advantage of information from those previously initialized. In other words, the j th individual could be generated in such a manner to improve the quality or diversity based on the previous individuals (e.g., $(j - 1)$ th individual). An example of comparison between PRNGs and quasi-random generators, for the case of a DE, can be found in Sacco and Rios-Coelho (2018), which affirmed that the Mersenne twister is suitable for a small population while the Sobol' sequence (an example of a quasi-random sequence) is adequate for large population sizes. Also, another study has been proposed in Maaranen et al. (2006), which analyzed the properties of different point generators of a GA for continuous optimization problems. These results are in line with the previous statements. In other words, quasi-random sequences are suitable to cover large spaces and enlarge while PRNGs could be effective just for small search spaces, in which the diversification process might not be needed. This issue was studied in a broader way in Greistorfer et al. (2008), which compared the quality of sequential and simultaneous generation approaches. The paper focused on the solution quality but the idea could also be generalized to the diversity as we will see next. But first, we are interested in how to generate this sequence using chaos theory.

3.2. Generating diversified sequences of random solutions

As stated earlier, the generation of solutions in sequence is a promising approach to avoid obtaining initial solutions with similar values or properties. For example, the sequence of solutions can be defined such that the x_i solution is a combination of x_{i-1}, \dots, x_1 solutions. An example of a mathematical formula for the Sobol' sequence can be found in Sobol' et al. (2011). However, the Sobol' sequence is not primarily designed to diversify solutions. In fact, when designing the population, it is more important that the points are distributed as uniformly as possible to imitate diverse random points. A common way to implement this feature is to adopt a chaotic initialization of the algorithm: Chaotic methods mimic the behavior of dynamic systems and are very sensitive to their initial conditions.

We can find in the literature that different forms of chaotic maps have been adopted to generate an initial population (e.g., tent, logistic, sinusoidal maps). But, as in Kazimipour et al. (2014) and Elsayed et al. (2017), we propose in Equation (2) a generic formula that can be used to generate an initial population for the algorithms:

$$x_{i,j}^k = f_{ch}(x_{i,j}^k), \quad (2)$$

where each j corresponds to the j th variable of the i th individual of the population (the first individual is generated randomly). f_{ch} is the mapping function (e.g., logistic, circle, sinus, tent). Such maps generate real values $\in [0, 1]$. k is the iteration number, which is equal to 1 as we are interested only in the first iteration (initialization phase).

We note that the use of chaotic approaches has been mainly embraced along with other approaches. For instance, Gao et al. (2012) adopted an initialization approach that uses a chaotic system and an opposition-based learning (OBL) scheme (which is described in Section 4.3) to generate an initial population. The authors adapted Equation (1) by replacing the uniform random number generation by a chaotically generated number.

A fairly similar idea was introduced in Tian (2017). In that paper, the proposed approach consists first of randomly initializing solutions. Then, if a variable plunges into the fixed points (as in the example above regarding binary problems), a tent map is adopted to add a very small positive random perturbation (the pseudo-code of the approach is available in that paper). Moreover, the author adopted a hybridization of a chaotic initialization of PSO and a Cauchy mutation (a type of a mutation operator). The motivation beyond hybridizing chaotic approaches is to attempt to deal with the exploration–exploitation dilemma. Indeed, Snaselova and Zboril (2015) emphasized the benefits of mixing chaotic and nonchaotic individuals. That is, the mixing of (chaotic) individuals, with better exploration capacity, and others, with good exploitation capability, is a propitious option to manage the aforementioned dilemma. Also, the coupling of a number of chaotic approaches could also be an emerging concept, as outlined in Ozer (2010), which compared the performance of several chaotic procedures. The authors' idea is to generate different chaotic variables according to a chaotic map formula (Equation (2)).

We can then conclude that chaos theory is a research line that could benefit the initialization of metaheuristics. Additionally, although sensitivity to initial conditions is often seen as a drawback of algorithms, it can be exploited to provide a more diverse population or to improve the search

process, as elaborated in Fischetti and Monaci (2014). The authors coined the concept of *erraticism*, which reflects the sensitivity of initial solutions (for a more detailed illustration of the concept, interested readers are referred to Lalla-Ruiz and Voß, 2016). More details on the options to exploit this concept can be found in Section 7.2.

On the other hand, a common approach to evaluate these chaotic solutions is to adopt entropy. Fuertes et al. (2019) claimed that chaotic maps with higher entropy show increased fitness density and generate better solutions. The authors founded their claims on experimental results. However, it would be of the utmost importance to explain the reasons for such behavior. Also, the impact of the entropy measurement may differ from one algorithm to another. In fact, Liu et al. (2007) intended to illustrate the chaotic behavior of the swarm intelligence. Their study shows that the PSO results are correlated with the maximal exponent of Lyapunov (Skokos, 2010). Likewise, according to Snaselova and Zboril (2015), the use of chaotic maps to generate the initial GA population avoids premature convergence to a local optimum. However, this characteristic is again mainly related to the dynamic system called Lyapunov exponent. In addition, other measures of the diversity of the initial population could be proposed. For example, Diaz-Gomez and Hougen (2007) suggested a metric based on the center of mass to measure diversity.

As far as entropy is concerned, it can be calculated on the basis of the problem data, as Vargas et al. (2018) did. The authors asserted a positive correlation between the entropy of the initial populations and the performance of a GA while solving the problem at hand. Nevertheless, it is unknown if this could be generalized to different problems. Also, Saroj et al. (2011) extended a GA by incorporating a form of entropy-based probabilistic initialization to automate the process of rule mining.

At the end of this section, we can conclude that the use of PRNGs is a typical way to generate an initial population. However, the generation of individuals in sequence holds more promise in either promoting the quality of solutions (in terms of their fitness function values) or promoting diversity (Gagnon et al., 2021). Chaos theory is therefore the most propitious way to generate diverse initial solutions, which is the most needed at this stage, as underlined in Section 2. The contribution of sequential generation is clearly shown for large-scale solution spaces and where there is no computational advantage (e.g., parallel computing) for other approaches.

In Table 2, we summarize the current literature related to randomized approaches. More precisely, we highlight the methodology adopted in each paper for the main papers above, as well as their contribution(s) and/or finding(s). If quality or diversity issues are involved in the paper, we highlight the most considered issues in parentheses. Finally, we show the algorithm(s) considered in the study (noted Alg.). In Table 2, the papers are sorted in chronological order to give an overview of the progress of the work on the subject. (We note that in the table, the contributions start with a verb while findings begin with a noun to differentiate them.)

4. Learning

Learning can be incorporated into metaheuristics in various ways. Specifically, we look at supervised learning, Markov models, OBL as well as some alternative of OBL.

Table 2

Studies of randomized approaches: methodology adopted, contributions and/or findings, and algorithm considered

Author(s)	Methodology	Contributions and/or findings	Alg.
Maaranen et al. (2004)	Quasi-random sequence of points	Improve the distribution of the solutions (diversity)	GA
Kimura and Matsumura (2005)	Low-discrepancy sequences	Propose a uniform distribution of the solutions (diversity)	GA
Maaranen et al. (2006)	PRNGs	Analyze the properties of different point generators Investigate the effects of the uniform coverage	GA
Liu et al. (2007)	Chaos theory	PSO with a high maximum Lyapunov exponent usually achieves better performance than other approaches	PSO
Ozer (2010)	Chaos maps	Analyze several chaotic maps	DE
Saroj et al. (2011)	Entropy	Generate an initial population that has relevant and informative attributes (quality)	GA
Ma and Vandenbosch (2012)	PRNGs	Study and compare the impact of PRNGs on PSO	PSO
Gao et al. (2012)	Chaotic maps and OBL	Enhance the convergence of a stochastic PSO variant (quality)	PSO
Snaselova and Zboril (2015)	Chaotic maps	Exploit a mixture of chaotic solutions (diversity) and convergent solutions (quality)	GA
Tian (2017)	Chaos theory	Generate uniformly distributed solutions (diversity)	PSO
Alhalabi and Dragoi (2017)	PRNGs	Study the influence of PRNGs on DS; Weibull distribution has the best results	DS
Sacco and Rios-Coelho (2018)	PRNGs	A PRNG is suitable for a small population while a quasi-random sequence is adequate for large population sizes	DE

DE, differential evolution; DS, differential search; GA, genetic algorithm; OBL, opposition-based learning; PRNGs, pseudo-random number generators; PSO, particle swarm optimization.

4.1. Supervised learning

In the introduction, we outlined that supervised ML approaches are useful to learn from previously solved problem instances. Such approaches can also be adopted when prior knowledge is not available. The idea beyond the use of supervised ML is to generate subsequent initial solutions by learning the characteristics of the solutions previously found. In other words, on the basis of the first solutions generated, the goal is to determine the remaining solutions based on the information extracted from the previous solutions. For a while, the adoption of ML for this purpose was proposed using case-based reasoning (Ramsey and Grefenstette, 1993). Surrogate optimization is a concept that can adopt supervised learning to build models from sample points (e.g., Kim and Boukouvala, 2020). An example of an ML algorithm that can be used for this purpose is support vector machines (e.g., Keedwell et al., 2018). Moreover, Zhou et al. (2007) adopted the concept of time series forecasting (in which supervised ML approaches are widely used) for this problem.

Supervised ML can be adopted for the initialization of different metaheuristics, as is done, for example, in Ren et al. (2018) for a cooperative co-evolution (COCO). Nevertheless, its practical

use is hardly arguable due to its impact on the computational time (e.g., the training process), except for very large-scale optimization problems. It is also important to leverage recent technological computational advances for an efficient supervised learning-based initialization. Another idea that could be explored further is to use these learning techniques for only parts of a population, as investigated in Li et al. (2021b), which adopted a modular neural network for this purpose.

In a different setting, one may also extend conventional metaheuristic approaches to incorporate related learning. An example is the fixed set search of Jovanovic et al. (2019) superimposed on the GRASP. Another idea that could be subsumed under this category, though developed in a different context, is target analysis as described, for example, in Laguna and Glover (1993).

4.2. Markov models

Another practical way of using ML is through Markov models. Markov models are stochastic mathematical models that are used to model randomly changing systems. The main assumption is that future states depend only on the current state, not on the previous states. It has shown success in modeling several metaheuristics in which the mathematical formulation fits with this assumption. Depending on the problem type, a Markov model can be named a Markov chain or a Markov decision process. An example of modeling a metaheuristic through Markov models can be found in Simon et al. (2011). Regarding metaheuristic's initialization, Caserta and Voß (2014) used a Markov chain based cross-entropy scheme that adopts a maximum likelihood estimator to generate an initial transition probability matrix that reflects the chances of obtaining high-quality solutions. The authors are primarily interested in maximizing the structural diversity of the initial solutions and adopt a smoothing factor for this purpose.

The approach is used to get one (or more) incumbent solution(s) for the corridor method (see, e.g., Sniedovich and Voß, 2006; Caserta et al., 2011), but it could be generalized. We note that although this idea was not directly extended, Markov chains and cross entropy are main ingredients of the ML reinforcement learning approach which is attracting a lot of interest today. It was used for metaheuristics initialization in de Lima Junior et al. (2007) and Cai et al. (2019). In addition, Hsu and Phoa (2018) came up with an initialization idea, which they called swarm initialization, for a swarm intelligence method they developed earlier. Their idea is to use a Markov chain Monte Carlo (MCMC) pool, and the authors claimed that it can accelerate the convergence of the algorithm.

4.3. Opposition-based learning

Another way to integrate ML for the initialization of metaheuristics is through the use of the concept of OBL (Tizhoosh, 2005). This idea has been exploited in many problems. In particular, it can be observed from the literature that it is the most-cited generic approach for the initialization of metaheuristics. To be more specific, Rahnamayan et al. (2007) proposed an initialization approach using OBL to generate an initial population for evolutionary algorithms. The main idea behind OBL is to consider both the generated random solution and its opposite. More precisely,

the algorithm generates n random individuals and calculates n opposite points, then selects the n fittest (best) individuals of the two populations. This proposed scheme was initially incorporated into DE in Rahnamayan et al. (2007), but it can also be incorporated into other population-based metaheuristics such as PSO (Wang et al., 2011). However, we note that the notions of an opposite solution and a distance can differ according to each algorithm, but it must conform to the following mathematical formula: let $P(x_1, \dots, x_n)$ be a point in an n -dimensional space, where $x_1, \dots, x_n \in R$ and $x_i \in [a_i, b_i] \forall i \in \{1, \dots, n\}$. The opposite point of P is defined by $\tilde{P}(\tilde{x}_1, \dots, \tilde{x}_n)$, where

$$\tilde{x}_i = a_i + b_i - x_i. \quad (3)$$

The numerical experiments carried out in that paper revealed that replacing random initialization with an opposition-based population initialization accelerates the convergence speed of the optimization algorithm. For a more detailed description of OBL, one can refer to Rahnamayan et al. (2008a).

It is notable that the main motivation beyond the adoption of OBL is to improve the chances of producing solutions closer to the global optimum. Moreover, Rahnamayan et al. (2008b) attempted to establish a mathematical proof that opposite points (solutions) have a higher probability of being closer to the best solution than a second random point (solution). However, as indicated before, diversification is the most desired feature for initial solutions, and can then be achieved by introducing the notion of diversification-based learning. Despite this issue not yet having been systematically addressed, in Section 4.4 we give insights on how to address it. In fact, as noted in Glover and Hao (2019), OBL corresponds to the notion of a complemented solution in binary optimization. An example of OBL improvement with respect to its diversity was proposed in Park and Lee (2016), but was not yet projected to the initialization of the solutions. The issue was recently addressed in Ghannami et al. (2021), which combined the approach with a sampling technique. We can then conclude that OBL, despite its wide use, lacks diversification mechanisms and needs to be supplemented by other approaches, which are described in Section 5. Another issue is that OBL was mainly designed for unconstrained problems and found application in continuous problems. To our knowledge, no extension and adaptation of OBL has been proposed to deal with constrained and discrete problems.

4.4. Diversification-based learning

The notion of diversification-based learning has been explicitly introduced in Glover and Hao (2019), which provides some alternatives to the OBL, highlighted next. In general, population-based metaheuristics seek initial solutions that are meaningfully opposed to all other solutions in the population. That is, a single solution has to be the farthest point to the collection of all other solutions. In other words, the authors replaced the notion of an opposite solution with the notion of a diverse (opposite) collection of solutions.

We note that despite that this concept is not formally well known as a concept associated with metaheuristics initialization, it is integrated in some metaheuristics. For example, the idea is explicitly included in the generation phase of SC (Laguna and Martí, 2003). In the three next sections, we show how this concept could be integrated into different types of problems. More concretely,

Table 3
Studies of learning approaches

Author(s)	Methodology	Contributions and/or findings	Alg.
Ramsey and Grefenstette (1993)	Case-based reasoning	Guide the search in changing environments	GA
Zhou et al. (2007)	Time series prediction	Reinitialize the algorithm when a change occurs in the fitness function	Dynamic MOO
Rahnamayan et al. (2007)	OBL	Improve the fitness function of the generated population (quality)	Dynamic MOO
de Lima Junior et al. (2007)	Reinforcement Learning (RL)	Reinitialize the algorithm when a change occurs	GA and GRASP
Caserta and Voß (2014)	Markov chain based cross-entropy	Generate an initial transition probability matrix that reflects the chances of obtaining diversified solutions (diversity)	Corridor method
Hsu and Phoa (2018)	MCMC pool	Accelerate the convergence (quality)	Swarm intelligence
Ren et al. (2018)	A surrogate model	Evaluate only promising solutions	COCO
Keedwell et al. (2018)	Support Vector Machines (SVM)	Reduce the computational cost (diversity)	DE
Cai et al. (2019)	RL	Propose a RL extension, that builds upon proximal policy optimization, to better learn an initialization for algorithms	SA
Glover and Hao (2019)	Diversification-Based Learning (DBL)	Motivate that DBL is more promising than OBL. Illustrate that some typical approaches for initialization fall under the umbrella of DBL	-
Li et al. (2021b)	Neural network	Generate some solutions when a change occurs, and generate others at random to maintain diversity (quality)	Dynamic MOO
Ghannami et al. (2021)	Sampling and OBL	Path length diversity is more important during the initialization phase. A repair function can introduce diversity to the population (diversity)	PSO

COCO, cooperative co-evolution; DE, differential evolution; GA, genetic algorithm; GRASP, greedy randomized adaptive search procedure; OBL, opposition-based learning; MCMC, Markov chain Monte Carlo; MOO, multiobjective optimization; PSO, particle swarm optimization; SA, simulated annealing.

in Section 5, we are interested in the implementation of this concept for unconstrained continuous problems, which is the simplest case. In Section 6, we outline the main issues with respect to constrained problems.

In Table 3, we summarize the work on learning approaches in the same way as for randomized approaches.

5. Sampling and decomposition

In Section 3, we focused on the probabilistic approaches that are used to generate pseudo-random initial solutions. Here, we first look at sampling before we focus on clustering and cooperation.

5.1. Sampling

In this section, we show statistical approaches that aim to sample the search space in the most appropriate way. In general, these approaches aim to diversify the search for solutions to be generated within the search space instead of the probabilistic sampling methods that were presented in that section.

A typical statistical concept that could be used to generate diverse solutions is the experimental design (or design of experiments). It aims to determine the relationship between the factors and the output. Uniform design is an approach that aims to sample a small set of points, from a given set of points, that are uniformly scattered. It was applied, for example, in Leung and Wang (2000) to generate an initial population for a GA in the multiobjective optimization (MOO) space. Another type, which was defined in Zhang and Leung (1999), is the orthogonal design. The motivation of using an orthogonal array is to specify a small number of combinations of solutions that are scattered uniformly over the space of all the possible combinations. It was applied in Leung and Wang (2001), along with a quantization technique, to generate an initial population for a GA in a generic way for different types of problems. The concept of orthogonal design was also adopted in Gong et al. (2009) to uniformly scan the neighborhood around each solution for the artificial immune system (AIS) metaheuristic. We note here that both Leung and Wang (2000) and Leung and Wang (2001) observed that some major steps of a GA can be considered as an experimental design. Indeed, the aim of the first iterations of a GA, and other metaheuristics, is often to explore the search space and to locate promising regions. Then, it is reasonable to adapt the suitable statistical concepts to improve the exploration process. At the time of developing their ideas, the authors confirmed positive outcomes. However, those approaches have not been extended over the past decade, as nowadays much of the emphasis is on learning and advanced data analytic concepts instead of basic statistical methods.

Nevertheless, despite the rise of learning approaches, appropriate statistical tools are also worth studying due to their computational advantage. Latin hypercube sampling (LHS) is an example of a statistical approach that has shown its effectiveness, and it is nowadays the most adopted for metaheuristic's initialization. It divides a domain into intervals in each dimension, then places sample points so that each interval in each dimension contains only one sample point. In other words, LHS is a spatial filling mechanism that creates a grid in the search space by dividing each dimension into equal interval segments, and then generates random points within an interval. It was adopted in Mousavirad et al. (2019) for the initialization of the population of a DE along with OBL. Mahdavi et al. (2016) proposed population initialization strategies that attempt to generate points around a central point with different schemes. Li et al. (2020) compared LHS with other probabilistic initialization approaches (mentioned in Section 3). The authors recommended its use for PSO. Although the number of papers that focused on LHS for initialization is limited, several papers adopted it when proposing their approaches (e.g., Tian et al., 2020). In addition, Georgieva and Jordanov (2009) proposed an advanced approach that exploits this idea. The authors suggested an initial seeding of a GA by choosing the centers of regions and then seeding the neighborhood of each solution.

Other papers, such as Rahnamayan and Wang (2009), proposed other sampling strategies. However, they did not provide enough evidence on the effectiveness of their approaches and their

findings need to be further clarified. Another idea is to combine sampling and learning approaches as in de Melo and Delbem (2012). The authors define a concept they named smart sampling for the initialization of a DE. Their approach consists of using an ML algorithm to identify promising and nonpromising solutions, to guide the resampling procedure to smaller areas that include high-quality solutions.

At the end of this section, we note that more specific studies, tailored to particular algorithms such as GA, have been conducted. For example, in Maaranen et al. (2006), in addition to a random sequence initialization, the paper proposed a sequential inhibition process in order to produce points that are either diverse or have good uniform coverage.

5.2. Clustering

In general, the statistical sampling-based approaches highlighted in Section 5.1 are suited for small and medium optimization problems. However, for large-scale problems, it is often impractical to sample the entire search space. In such a case, a typical way to ensure the diversity of the generated population is through the clustering of the solution space. There are a number of clustering approaches used in the literature for metaheuristic initialization. Bajer et al. (2016) employed a clustering method that first generates uniformly random population solutions. The resulted cluster centers are then used to identify promising regions of the search space. A mutation operator is then applied to generate new individuals around these centers while promoting the best. According to the authors, although the approach requires more time to generate the initial population, it exhibits an increased convergence rate resulting in a lower execution time in most of the functions tested. The incorporation of clustering has also been proposed from different perspectives. For instance, Elsayed et al. (2017) proposed to decompose the search domain of each decision variable into segments, then generate combinations of different segments of all the variables such that different points of different areas of the search space can be selected. Main contribution of the approach is that the fitness values of the initial population could provide information on the pattern of the function's behavior. A three-step approach has been proposed in Poikolainen et al. (2015). In the first step, two fast local search algorithms are applied. In the second step, the solutions are clustered to identify the basins of attraction (this concept is explained in Section 7.5). In the third step, the solutions are sampled from the clusters to identify the most promising basins.

We can conclude that clustering methods are promising for metaheuristics initialization. However, they face the challenge of the additional computational time required to run them.

For this, other simplistic clustering approaches could be adopted. An example is the Voronoi tessellations (VT). VT, as proposed by Du et al. (1999), can be described as a way to compartmentalize an area. In particular, in centroidal VT, the generating solution of each Voronoi cell is also its centroid. The approach was examined for an evolutionary algorithm, namely estimation of distribution algorithm (EDA) in Muelas et al. (2010). The authors defined a partition set of the solution space in which each island or node will start its own exploration. More concretely, the initialization is based on the isolation of the initial search space of each island and the use of a heuristic method to uniformly cover each region of the search space. Regarding swarm intelligence, Shatnawi and Nasrudin (2011) investigated this idea for cuckoo search, but to our knowledge, no in-depth

studies have been conducted to evaluate this approach. Nonetheless, we note that the approach shares similarities with LHS, which is gaining more attention.

On the other hand, we note that the concept could be indirectly incorporated in different population-based algorithms. For evolutionary algorithms, Samorani et al. (2019) proposed a clustering strategy that aims to isolate promising regions of the search space, and then to select parents in different promising regions in order to generate new solutions. Additionally, the concept is mostly exploited in swarm intelligence as it is based on the cooperation between different individuals, which ideally explores the search space in a divided way.

5.3. Cooperative approaches

The aim of this section is to give insights on how to generate and design the initial population of swarm intelligence approaches in such a way that the individuals cover the search space in the best manner. In swarm intelligence, the concept of clustering is often presented under the term of cooperation or cooperative optimization (or learning).

First, a typical approach is to divide it into subpopulations. Such an approach is frequently used for the parallelization of algorithms. In fact, Przewoźniczek (2020) claimed that the random initialization of some population-based metaheuristics may lead to what he called Long-Way-To-Stuck. Moreover, the author proposed to divide the population into subpopulations. However, it was affirmed that some subpopulations will be unable to contribute to the improvement of the solution if they are randomly initialized.

A classical multipopulation strategy is problem-oriented in the sense that each subpopulation independently optimizes a part of the problem (e.g., Sarhani and Voß, 2022). Other advanced multipopulation optimization methods are used to improve the search diversity by splitting the entire population into groups, in which each one has a specific role. But it is important that the split of the population will result in a guaranteed improvement in diversity. Moreover, the concept could also be used to achieve both diversity and quality. This was investigated, for example, in Zhan et al. (2009), in which each particle's group has its specific mathematical formula (e.g., velocity update) depending on its positioning in the exploration–exploitation dilemma. We can conclude that the proposition of different initialization formulas for each subpopulation is a promising area that could be further explored and benefit from recent advanced approaches.

Second, the concept of cooperation is explicitly implemented in ACO (Dorigo and Gambardella, 1997; Sondergeld and Voß, 1999). In ACO, the pheromone trail is associated with elements of the solutions (e.g., edges for the case of the TSP). One typical approach is to initialize pheromone trails at random. However, some studies have shown that other approaches may be more effective. According to Blum and Dorigo (2004), it is customary to initialize pheromone trails, when dealing with problems such as the TSP, at small numerical values. According to the authors, this is the case with most ACO variants. But Stützle and Hoos (2000) recommended to initialize the pheromone trails to the maximum value available, thus achieving a higher exploration of solutions at the start of the algorithm. Other papers were rather interested in the improvement of the exploitation of the algorithm. Indeed, Bellaachia and Alathel (2014) and Dai et al. (2009) proposed two different initialization strategies to improve ACO's ability to convergence.

Table 4
Studies of sampling and decomposition approaches

Author(s)	Methodology	Contributions and/or findings	Alg.
Leung and Wang (2000)	Uniform design	Propose initial solutions that are uniformly scattered toward the Pareto frontier in the MOO space (diversity)	GA
Stützle and Hoos (2000)	Randomization	Recommend to initialize the pheromone trails to the maximum available value (quality)	ACO
Leung and Wang (2001)	Orthogonal design	Select solutions that are uniformly scattered over the feasible solution space (diversity)	GA
Dai et al. (2009)	Minimum spanning tree	Improve the convergence of the algorithm and its fitness values (quality)	ACO
Gong et al. (2009)	Orthogonal design	Locate good points (diversity)	AIS
Muelas et al. (2010)	VT	The partition of the search space reduces the modality of the constrained regions	EDA
de Melo and Delbem (2012)	Smart sampling	Finds regions with high possibility of containing a global optimum (diversity)	DE
Bellaachia and Alathel (2014)	Local pheromone initialization	Use information available locally for ants to calculate the initial pheromone level on uninitialized edges. Improve the convergence (quality)	ACO
Poikolainen et al. (2015)	Clustering, local search	Sample solutions from clusters to identify the most promising basins of attraction (diversity)	DE
Mahdavi et al. (2016)	LHS	Generate points around a central point with different schemes (quality)	COCO
Bajer et al. (2016)	Clustering	Increase the convergence rate of the solutions (quality)	DE
Elsayed et al. (2017)	Clustering	Cover the search space uniformly (diversity)	DE
Li et al. (2020)	LHS, PRNGs	LHS is recommended for PSO	PSO
Przewoźniczek (2020)	Linkage learning-driven subpopulation	Show the negative influence of the Long-Way-To-Stuck effect. Propose new subpopulation initialization procedure	GA

ACO, ant colony optimization; AIS, artificial immune system; COCO, cooperative co-evolution; DE, differential evolution; EDA, estimation of distribution algorithm; GA, genetic algorithm; LHS, Latin hypercube sampling; PRNGs, pseudo-random number generators; PSO, particle swarm optimization; VT, Voronoi tessellations.

In Table 4, we summarize the works described in this section in the same manner as for the previous works.

6. Constrained optimization

We note that the methods outlined in the previous sections are generally applicable to constrained optimization. However, many of them need to be adapted to ensure the feasibility of the solutions generated. Indeed, randomly generated solutions are unlikely to provide feasible solutions for most constrained problems because, in general, the feasible domain is tiny, especially for combinatorial

optimization problems. Our aim in this section is to shed light on specific techniques and tricks that can be used to initialize constrained problems in the best and most efficient manner.

6.1. Generation of initial solutions

First, we are interested in how to ensure the feasibility of the generated initial solutions. In fact, there are few papers that have addressed the problem of constraint handling in the initialization process. For example, in Sacco and Henderson (2011), as an alternative to random initialization, the authors used the Luus–Jaakola algorithm (a kind of heuristic) to generate an initial sampling with points belonging to the feasible region. Another idea that could be adopted is inverse optimization. This approach is often used to infer unknown problem parameters such as constraints. In Ghobadi and Mahmoudzadeh (2021), the authors adopted it to infer linear feasible solutions and to provide a baseline for the initial filtering of future solutions based on their feasibility.

Despite keeping the feasibility of the solutions is the obvious approach for effective generation of initial solutions, the relaxation of some problem constraints is a promising concept. The idea is to stipulate that X represents a set of solutions derived from a problem relaxation. This problem can be taken as a starting point for metaheuristics that generate fully feasible solutions (Glover and Hao, 2019). The most challenging issue is to drive the search to feasibility when it is lost. A common approach for this is to use a penalty function. Tometzki and Engell (2011) adopted a positive penalty coefficient that guides the search in infeasible regions toward a feasible region. In that paper, in case of infeasible initialization, the population is driven toward feasibility by a penalty function. Also, Dengiz et al. (1997) embraced a repair function as well as a stochastic depth-first algorithm to generate the initial population. However, such an approach could involve problem knowledge, and more research is needed on how to guide initial solutions to the feasible area in the least costly manner.

For constrained optimization, an important question that arises is the impact of the feasibility of the initial solutions on the outcome of the methods. According to Azad (2017), metaheuristics with randomly initialized solutions failed in most cases to locate feasible solutions in the early stages of the optimization. The authors then proposed to seed the initial population with feasible solutions. However, the authors did not take into account the relaxation approaches outlined above. Olikar and Bekhor (2020) proposed an approach for the transit route network design problem. It first starts from infeasible solutions that assign all transit routes with the maximal frequency, and then eliminates routes and decreases frequencies of the less attractive solutions. Elsayed et al. (2012) attempted to analyze the effect of the number of feasible individuals, in the initial population, on the performance of a DE algorithm. Their analysis aims to help judge both whether the initial population should include feasible individuals and also whether to increase the number when there are very few. However, work on this issue is still insufficient and further studies are needed to validate and extend previous work.

6.2. Diversification of solutions

Another important issue is how to generate diverse solutions for constrained optimization. In this case, the initial population needs to be uniformly scattered over the feasible solution space, so that

the algorithm can explore the entire solution space evenly. Clustering could also be an option to select diverse solutions for mixed-integer programming problems (Danna and Woodruff, 2009). However, the relationship between clustering methods and the formation of sets of diverse solution vectors is a rich area for further exploration.

Moreover, experimental design could also be customized for constrained optimization. In Leung and Wang (2001), the authors' approach consists of dividing the feasible search space into a number of subspaces based on a defined formula. The next step is to quantize each subspace and select the best solutions.

Another way to generate initial feasible solutions is to adopt approaches beyond metaheuristics. Indeed, finding good feasible initial solutions is an even more general issue in operations research and a common problem addressed in different optimization methods. For instance, the classical transportation problem is one of the most considered problems in operations research, and a large number of papers are devoted to finding initial feasible solutions (e.g., Juman and Hoque, 2015; Karagul and Sahin, 2020) that are then possibly improved. Often, however, a solid evaluation whether and how the initial feasible solutions improve the overall time-to-target quality is occasionally missing. More in-depth studies need to consider specific problem settings and related solution approaches to be evaluated. For example, we refer to the case of transit network design (and frequency assignment), where a classical idea is to apply a route construction as a seed for improvement and then to apply some sort of route repair; see, for example, Iliopoulou et al. (2019). Similar ideas also come up in case of problem settings, where disturbances need to be considered and related repair mechanisms might be applied, where a given (disturbed) solution is taken as initialization for the subsequent optimization process; see, for example, Ge et al. (2022) for some references.

We note that the use of relaxation is particularly interesting for combinatorial optimization problems (Glover and Hao, 2019). Next we provide a suggestion on how to deal with such a problem.

6.3. Initialization using single solution based metaheuristics for combinatorial optimization

In the metaheuristics community, single-solution metaheuristics are widely used to initialize population-based metaheuristics. In other words, when researchers and practitioners want to implement the initial values of algorithms, they often adopt rapid local searches as alternatives to randomly generated solutions. The reason is often to supply them with good feasible solutions. The contribution of these approaches is appealing in combinatorial optimization as they are associated with the notion of a neighborhood. That is, a local search approach usually involves choosing the best solution in a neighborhood. The question that may arise in this context is what local search approach should be considered in order to generate a diverse population of solutions. Although it is not straightforward to provide affirmative options due to the free lunch theorem for optimization (Wolpert and Macready, 1997), the literature could provide rules of thumb that could be useful in practice. More concretely, multiple papers have shown that GRASP may be an appropriate tool to initialize population-based methods for combinatorial optimization. In Ahuja et al. (2000), a GRASP heuristic is applied to generate the initial population for a GA. Moreover, it was considered as a typical example for the diversification-based learning scheme (Section 4.4) proposed in Glover and Hao (2019) and was incorporated into the first design of SC. That paper refers to

Table 5

Studies of constrained and combinatorial optimization approaches

Author(s)	Methodology	Potential and/or findings	Alg.
Ahuja et al. (2000)	Single-solution approach	Generate diversified solutions using GRASP	GA
Danna and Woodruff (2009)	A local search & Sequential screening	Suggest ideas to generate a small subset of solutions that maximizes their diversity (diversification)	-
Tometzki and Engell (2011)	Mixed-integer program	Drive the population toward feasibility, in case of infeasible initializations, by a penalty function	GA
Elsayed et al. (2012)	Chaos maps	Analyze the effect of the number of feasible individuals, in the initial population, on the algorithm's performance	DE
Juman and Hoque (2015)	Heuristic	Adopt approaches beyond metaheuristic literature for initialization. Develop a heuristic technique for obtaining initial feasible solutions to the transportation problem	-
Ghobadi and Mahmoudzadeh (2021)	Inverse optimization	Inform the desirable properties of the feasible region based on user preference and historical data	-

DE, differential evolution; GA, genetic algorithm; GRASP, greedy randomized adaptive search procedure.

Campos et al. (2001), where the authors developed and tested several diversification generation methods (most of them are based on GRASP variants). The reason of these selections is that GRASP has shown effectiveness in providing diverse solutions compared to simple local searches such as TS. For example, Aringhieri et al. (2007) found that GRASP is better at solving the maximum diversity problem. The problem, which was defined in Glover et al. (1998), consists of selecting a number of elements in a set so as to maximize the sum of the distances between the chosen elements. The reason could be due to its sampling capacity. Indeed, according to Feo and Resende (1995), GRASP construction can be seen as a repetitive sampling technique, in which each iteration produces a sample solution from an unknown distribution; see also Hart and Shogan (1987) for a related discussion of the repetitive nature of the approach (though not yet using the notion of GRASP). It is of interest that the pilot method also, to a large extent, relies on repetition; see Duin and Voß (1999). GRASP was by design used in the initialization of the recently proposed fixed set search metaheuristic (Jovanovic et al., 2019), which has shown favorable results in quite a few problems (e.g., Jovanovic and Voß, 2021).

In Table 5, we summarize the work corresponding to both constrained and combinatorial optimization approaches.

7. Single-solution metaheuristics

The aim of this section is to give some helpful insights, which can be used to generate the sequence of initial solutions in the best way for the next iterations of a multistart single-solution metaheuristic. In fact, from the previous sections, it is clear that the interest in metaheuristic initialization is

dedicated to population-based methods, and that this issue is rarely addressed for single-solution methods. The reason is that it appears to be illogical to search for the best first single solution when no knowledge is available on the search space. Nevertheless, initialization can be incorporated here using multiple runs or a sequence of solutions as described next. We start with motivation before delving into specific approaches.

7.1. Motivation and contextualization

In this section, we are interested in how to design the sequence of initial solutions as it has an impact on the outcome of the algorithm and its diversification. The aim of this part is then to give an overview on how to generate a diverse sequence of initial solutions instead of a simple local search. In other words, rather than proposing a local search that works in the same way for the whole process, we propose insights on the first iterations that can be exploited later regardless of the local search adopted.

In general, local search algorithms have a good intensification capacity, while their drawbacks are mainly due to their lack of capacity to diversify the search. In fact, local searches typically focus on the exploration of the neighbors of the previously visited solutions, and therefore the exploration of new regions requires several iterations and needs the handling of different local optima (Hao and Solnon, 2020). Nevertheless, the algorithms presented in Section 2.3 provide mechanisms to maintain the diversity of the search. Moreover, several improvements have been proposed in this regard. The hybridization with population-based metaheuristics is a suitable option for this purpose. In Lozano and García-Martínez (2010), several examples were reviewed. Other approaches are based on tuning the algorithms in an intelligent and adaptive way. An example for the case of TS was proposed in Neveu et al. (2004). As before, our objective is to project these tools for the initialization phase of local searches and to highlight those that are suitable for this phase. Such approaches often need to diversify the search and obtain (possibly in a multiobjective setting) as much information as possible in a very short time, by improving and extending the ideas presented in Section 2.3.

Next, we focus on the generation of the sequence of initial solutions in the best informative way for the next iterations. The issue is often addressed in the literature by the terms reinitialization, restart, bet and run, or multistart (the latter is the most generic and encompasses the different approaches depicted in Section 7.2).

We should note in passing that many of these methods may use some random restart. In this section, we formalize approaches used for such repeated runs that are in essence related to initialization. Many of these metaheuristics do not have multiple starts per se as part of their definition but that is how they are often used.

7.2. Multistart approaches

One promising way to escape a local optimum when using a local search is to generate a new starting solution and then reinitiate the process. This approach, exploited in ILS, consists in restarting the algorithm when no progress is observed, which reflects that the algorithm is stuck in a local

optimum. This idea has been integrated in different algorithms to avoid premature convergence. For example, Alfonzetti et al. (2006) adopted it for the case of an SA when facing a local optimum. The typical way to do it is through randomization. However, randomized approaches could be questioned, as outlined next.

First, as indicated in Charon and Hudry (2001), the efficiency of ignoring the previous solutions when building a new, entirely random initial solution is questionable. The authors recommended generating new starting solutions for local search in a smart way rather than just providing random restart solutions.

Second, from a computational perspective, the random reinitialization often needs a considerable amount of time to reach a good solution. Another approach that could be used to enhance solution's diversity is to apply a local search multiple times in parallel, with initial solutions chosen arbitrarily. The initialization phase in this case is to perform a random search several times in parallel. For large-scale optimization, some advanced approaches could be introduced. For instance, Tseng and Chen (2008) used a multiple trajectory search, which adopts multiple agents to search the solution space concurrently and where each agent adopts an ILS. Clustering is also an option to divide the search space before applying single-based solutions in parallel, as investigated, for example, in Oliveira and Lorena (2007). However, we note that this approach is similar to the case of having a population of solutions. Despite being able to exploit parallel computing, its computational time needs to be further improved.

Moreover, multistart strategies could aggregate different hybridization of metaheuristics, which use them sequentially as investigated, for instance, in Fuad (2013). However, the first algorithm must often be fast and help increase the diversity of the latter. Multistart approaches may also utilize the previously determined search history so as to avoid revisiting already known solutions in a TS as proposed and applied by Sondergeld and Voß (1996).

Also, another crucial issue concerns the selection of exploitable solutions among those generated. In fact, the basic approach consisting in retaining the solution corresponding to the best local minimum obtained does not often lead to the overall best solution. For this purpose, an alternative option, which was proposed in Fischetti and Monaci (2014), is to provide several random runs to explore the search space and select the most promising one. The contribution of this approach compared to those above is that it aims to limit the use of local search on only promising areas and to explore only their neighborhoods. The authors affirmed that the approach was able to find good solutions. Their motivation is that local searches are sensitive to initial solutions (*erraticism*) and that the selection of good runs can have a significant impact on the performance of the algorithm. But the authors stressed the importance of generating diversified runs and selecting the most promising. We have covered the first problem above and will focus on the second problem next.

7.3. Intelligent and adaptive approaches

First, we note that Martí et al. (2013) divided multistart methods into memory-based and memoryless procedures, which are described in this part and the previous part, respectively. Regarding memory, the aim of intelligent and adaptive approaches is to exploit memory to restart the process in the most efficient way. In the literature, the implementation of intelligent and adaptive multistart

strategies is often achieved either by generating new starting solutions for the local search in an intelligent way instead of randomization, or allowing worsening moves.

The first approach has been incorporated in the literature in GRASP and ILS, as highlighted next. For example, Boese et al. (1994) proposed a multistart method where the starting points for greedy descent are deceptively derived from the best local optimum previously found. Another approach that can be used for GRASP is the pilot method (Duin and Voß, 1999). In fact, the authors noted that the big mistakes in the optimization process are usually done in the initialization phase, and proposed the pilot method to address this issue. Related results for the pilot method can be found, for example, in Voß et al. (2005).

Weise et al. (2019) incorporated the notion of performance prediction. The authors adopted a method that aims to predict the future performance of the initial runs in order to select the most promising while considering the corresponding time budget. Brandão (2006) adopted an intelligent technique that generates initial solutions from pseudo-lower bounds. Their approach consists in restarting the search from the best-known feasible (or even infeasible if none is feasible) solution.

The second approach could be incorporated in SA and TS by tuning the corresponding parameter, shown in Table 1. Another promising idea is to adaptively generate the sequence of solutions. One approach that could be exploited in this regard is the reactive TS (Battiti and Tecchiolli, 1994). Reactive TS aims at the automatic adaptation of the tabu list length. The idea is to increase the tabu list length when the tabu memory indicates that the search is revisiting previously visited solutions. A similar idea, which was named prohibition-based diversification in Battiti and Brunato (2018), consists of prohibiting previously visited solutions and guiding the search space beyond them. We note here that the integration of other algorithms such as SA could provide a better diversification strategy for reactive TS, as shown, for example, in Voß and Fink (2012). In this case, if the search seems to repeat an excessive number of solutions too often, then the search is diversified by performing a number of random movements proportional to a moving average of the cycle length. Moreover, the aforementioned distances for combinatorial optimization can also be adopted to diversify the solutions, in an adaptive manner, as proposed for example in Voß (1995).

We note that, although these approaches are not limited to the initialization phase, they are particularly interesting for guiding the search in the most efficient way at this stage. An interesting deviation from these ideas may be to use the same starting solution in a repetitive way; however, with different information about parameters or the history of the search. For instance, Sondergeld and Voß (1996) propose to restart a TS (more specifically, the reverse elimination method) with modified lists representing the search, actually guiding toward diversified solutions.

On the other hand, we note that these restart strategies could also be incorporated into population-based metaheuristics. For example, in Tian et al. (2020), the authors proposed a reinitialization strategy based on the maximal focus distance, which can generate uniformly distributed initial particles (for a PSO). Simon et al. (2014) proposed a reinitialization technique for another swarm intelligence algorithm. An adaptive strategy was proposed for ACO in Birattari et al. (2007), which suggested that an effective initialization strategy of pheromone for ACO could be inversely proportional to the cost of the solution obtained by the nearest neighbor heuristic. Moreover, the authors proposed to seed the algorithm with a solution based on previous knowledge and observe its impact.

At the end of this part, we note that Martí et al. (2010) proposed another categorization, based on three elements, to classify multistart methods that are randomization, memory, and the degree

of rebuilding. In Sections 7.2 and 7.3, we discussed randomization and memory issues, respectively. The degree of rebuilding reflects the number or proportion of solutions that remain fixed. This issue is addressed in Section 7.4.

7.4. Measuring modification costs

The aim of this part is to introduce an issue that must be dealt with in the initialization phase. The problem is to specify what is the appropriate cost, in time and/or number of iterations, to allocate to the initialization phase. For example, for population-based methods, an important issue is to choose the population size, which is often a parameter that needs to be tuned. This problem could be addressed in a similar manner and in combination with the other parameters (e.g., Aoun et al., 2018). However, we note a lack of connection of this issue with the strategies presented above.

Another issue is the comparison between sequential and simultaneous generation of initial solutions, and we argued that the former is more suitable for their diversification. Nevertheless, there is a need to embrace the issue while also considering the computational time. Greistorfer et al. (2008) asserted that sequential generation frequently needs less computational time and produces solutions that are at least as good as those produced by simultaneous generation. Further research, which considers both recent algorithms and computational advances (e.g., parallelization), is required to obtain a unified assessment of the approaches.

In particular, this issue is crucial for multistart approaches. It was addressed in Weise et al. (2019), which extended Fischetti and Monaci (2014). The paper discussed the issue of effective management of the time budget and in particular the selection of the time required for initialization. As indicated previously, their approach, applied to a stochastic local search (SLS), intends to allow for each run a corresponding time according to its expected performance. A fairly similar idea was proposed in György and Kocsis (2011), which suggested to start multiple instances of a local search algorithm, and to allocate processing time to them depending on their behavior.

For sequential initialization methods, and in particular for constrained and combinatorial optimization problems, it is essential to evaluate the cost of generating and updating a solution. In Duin and Volgenant (2006), the authors sought to evaluate the modifications of the cost function adopted to generate a feasible solution and to find the cheapest adaptation of the cost. The objective is to optimize the difference between the initial solution and the following solutions.

Additionally, we refer to Watson (2010) who pointed out that sampling the search space should attempt to provide as wide a coverage of the search space as possible within the limits of an acceptable computational cost. For a recent approach to restrict the sampling to a subspace, see, for example, Li et al. (2021a). Watson (2010) asserted that the computational cost of sampling should be significantly lower than the cost of solving the problem with randomly generated individuals. The assessment of the needed cost depends on each algorithm and problem. But, as a rule of thumb, we can expect that single-solution approaches require more diverse initial solutions than population-based methods. Indeed, in general, population-based methods have better diversification mechanisms than simple local searches. Hence, local searches generally require more diverse initial solutions to address this limitation.

At the end, we note that Watson (2010) also emphasized the promises of the fitness landscape analysis (FLA), which is a next.

7.5. Fitness landscape analysis

Finally, we aim to highlight a concept that can be exploited in defining initial solutions, which is FLA (Reeves, 1999). FLA aims to characterize optimization problems and improve knowledge of their properties (also called functionalities or characteristics). Its motivation in our context is that the characterization of a problem could lead to a better understanding of it and a better choice of initial solutions for each instance of the problem. More specifically, FLA makes it possible to sample the search space and extract the relevant and available features that could drive the search and reinitialize it based on feature values. For example, if the sample suggests a rugged landscape (which corresponds to a large number of local optima), the algorithm should accept worsening moves or exploring the different regions. The initialization strategy could also leverage specific findings, such as Ochoa and Veerapen (2017) who suggested a big-valley structure for typical TSP instances. The TSP is the best-known combinatorial optimization problem, and in this case local optima are clustered around one central global optimum.

For a comprehensive review of the different features that could be exploited, we refer to Malan and Engelbrecht (2013). In this paper, we are particularly interested in how to generate a representative sample allowing to guide the search space according to the extracted characteristics. Next, we highlight some works on this issue.

For example, in Muñoz et al. (2014), LHS was used to generate a sample. In Malan and Engelbrecht (2014), multiple random walks starting in different zones on the boundary of the search space were used as a basis for obtaining sufficient information. Jana et al. (2016) adopted random walks as a technique to characterize the features of an optimization problem such as ruggedness and smoothness. We note that the calculation of the aforementioned features differs according to the problem and the nature of the variables. Nevertheless, the same methodology could be adopted. Also, there are specific features for each kind of problem such as the basins of attractions for combinatorial optimization problems.

In fact, for local searches, a fruitful approach to generate a good solution is to target basins of attraction. These are the areas that lead to a certain local optimum. Indeed, it has been shown, for example, in Traonmilin and Aujol (2020), that their identification is useful for local searches. To achieve this goal, initial solutions should be diversified and uniformly distributed over the search space in order to sample basins of attraction of all local optima (Mehdi et al., 2010). Compared to the randomized multistart approaches, this is more computationally efficient because it restricts the adoption of local searches in their promising areas. A more in-depth and generic study was proposed in Prugel-Bennett and Tayarani-Najaran (2012), which focused on this issue and on the analysis of other features. FLA can improve the computational time also using an empirical hardness model as in Malone et al. (2018).

In Table 6, we summarize the work corresponding to single-solution approaches. In this table, as in the previous tables, we have not shown all of the above papers and chosen a representative article for papers with similar ideas.

At the end, we note that the approaches outlined above for FLA are also of interest for population-based methods. In particular, inspired from evolutionary biology, several features have been proposed to quantify the population of evolutionary algorithms through FLA (e.g., Wang et al., 2018), and sampling approaches have been adopted to measure these features.

Table 6
Studies of single-solution approaches

Author(s)	Methodology	Potential and/or findings	Alg.
Boese et al. (1994)	Adaptive multistart	Underline that random multistart approaches are insufficient in high-dimensional problems. Propose an adaptive approach that uses best-known locally minimum solutions to generate subsequent starting points	GRASP
Alfonzetti et al. (2006)	Restart	Propose an improvement to prevent the algorithm from becoming trapped in local minima (diversity)	SA
Tseng and Chen (2008)	Multistart	Adopt multiple agents to search the solution space concurrently	ILS
Mehdi et al. (2010)	FLA	Propose an initial population that aims to sample basins of attraction of all local optima (diversity)	GA, DE
Martí et al. (2010)	Multistart	Classify multistart methods based on randomization, memory, and degree of rebuild	-
Watson (2010)	FLA	Motivate the use of FLA to improve the theoretical understanding of local search. Point out that sampling the search space should attempt to provide as wide a coverage of the search considering the allowed computational cost	-
Malan and Engelbrecht (2014)	Random walks	Propose a progressive random walk algorithm to sample neighborhood structure (diversity)	-
Fischetti and Monaci (2014)	Restart	Coin the concept of erraticism. Exploit that concept to make a number of short sample runs with randomized initial conditions (diversity)	SLS
Weise et al. (2019)	Bet and run	Predict the future performance of the initial runs in order to select the most promising. Allow for each run a corresponding time according to its expected performance	SLS

DE, differential evolution; FLA, fitness landscape analysis; GA, genetic algorithm; GRASP, greedy randomized adaptive search procedure; ILS, iterated local search; SA, simulated annealing; SLS, stochastic local search.

8. Conclusion

The initialization of metaheuristics is a crucial topic that lacks a comprehensive and systematic review of the state of the art. The reason for the lack of such a review is that it requires an in-depth study of various related topics. In fact, it is not enough to search only the papers that focus completely on the initialization. Indeed, it is also needed to extract insightful ideas that can be found in papers which are interested in the initialization phase while proposing their metaheuristics (e.g., Georgieva and Jordanov, 2009) and on generic approaches for diversification that are useful for initialization (e.g., Glover and Hao, 2019). In this paper, we conducted a literature review on this subject, which made the connection between the work proposed for the initialization of metaheuristics and the broader advances in the fields of metaheuristics and optimization. In other words, in this paper we have reviewed the current literature on metaheuristic initialization, and analyzed and evaluated it based on its contribution to the algorithms, particularly with respect to enhancing the capacity of diversification and/or intensification.

More concretely, we began by showing in Section 2 the needed background of the topic. In that part, we have focused on the dilemma of diversification/intensification and highlighted how the main metaheuristics handled it, with the aim of drawing inspiration from it in the initialization phase. Next, we reviewed the current state of the art of metaheuristic initialization taking into account the advances in the field. In particular, in Section 3, we reviewed the approaches that are based on randomization techniques. In Section 4, we were interested in learning methods. In Section 5, we focused on the other generic methods based mainly on sampling, clustering and cooperation. Then we switched to one of the most challenging types of optimization problems, namely constrained optimization, covered in Section 6. Particular interest was devoted to combinatorial optimization in that section. In Section 7, we projected the topic of initialization onto local search approaches and provided information on how to design a diverse and informative sequence of initial solutions. Next, we highlight some paper findings and issues that should be noted.

First, as we mentioned earlier, random initialization is the classical way for initialization. This is due to the fact that it is a widespread opinion in the metaheuristic community that randomization is associated with diversification. That is, to improve the diversification capacity of a particular algorithm, the most common approach is to add a random operation. However, the notion of diversification is different from randomization. In fact, the goal of diversification is to produce solutions that differ significantly from each other (Glover and Hao, 2019).

In particular, we note that learning-based methods are getting a lot of attention these days. We advocate that the use of learning-based methods should not be conducted arbitrarily and should be consistent with the proper handling of the dilemma.

Second, according to Kazimipour et al. (2014), the role of this step is to provide an initial guess of solutions. We have argued in this paper that the initial phase should primarily aim at diversifying the solutions in order to explore the different regions of the search space. We can, therefore, conclude that the measure of the quality of the initial solutions should not be associated with the objective or fitness function value. In our paper, we adopted the term quality, for example, in the tables, for the sake of simplicity but we recommend that it should be used differently and be associated with diversity in practice.

Particularly, we underline that depending on each metaheuristic, different diversification approaches have been proposed. While providing a literature review on this subject is beyond the scope of this paper, we suggest, as in Glover and Hao (2019), that more theoretical studies are needed in this regard, and the adoption of advanced diversity control techniques (e.g., Park and Ryu, 2010) can be beneficial for the initialization phase.

Third, an important issue is to assess the advantages and disadvantages of different approaches based on the time required for initialization. In particular, advanced approaches, such as clustering and learning, are useful for high dimensional problems. For these problems, simple randomization and sampling approaches are unable to cover the search space appropriately, as noted, for example, in Maaranen et al. (2006) for the case of a GA. Also, as a complement to the second point, we note that the quality can be associated with the fitness value in a particular case. That is, if the generation and evaluation of a solution is expensive, it is important to have a good objective value for the initial population, albeit at the expense of diversity, because few solutions can be evaluated.

Fourth, we have highlighted similarities between the initialization of single-solution metaheuristics and population-based metaheuristics. In general, in both cases, the goal is to provide diversified solutions. For population-based approaches, we have argued that the generation of solutions in

sequence is a suitable way for initialization. This motivates the use of single-solution approaches. But, in the literature, the algorithms are mostly initialized using an arbitrary algorithm without convincing justification. In Section 6.3, we have provided an example study in this regard. Likewise, population-based methods can be used to initialize single-solution approaches (e.g., by clustering the solution space, as mentioned in Section 7.2). The number of hybrid approaches that combine the two types in the literature is countless. However, most of them are not based on actual evidence. In this paper, we have provided food for thought and we advocate for more practical research.

Fifth, in Tables 2–4, we summarized the approaches that have been used for algorithm initialization. We note that most of these approaches have been applied to one of the three algorithms, namely GA, DE, and PSO. In this paper, we also looked at ACO, which is a widely used algorithm, and SC because it has a different inspiration. The other population-based algorithms mentioned in this paper can be derived or share similar components with the algorithms described in Section 2.3, as indicated in de Armas et al. (2022) and Molina et al. (2020). Therefore, the results obtained can be projected onto typical algorithms and vice versa by examining the similarities between the algorithms.

Sixth, in this paper, we reviewed and provided insights that will help research in constrained and discrete (especially combinatorial) optimization problems. We can see that there is an overlap between the work on these two problem types. Additionally, the research on initialization of metaheuristics for MOO problems is very limited, some examples can be found in Zhou et al. (2007), Rahnamayan et al. (2007), and Friedrich and Wagner (2015). Due to the high computational cost of solving such problems, metaheuristics are extensively used with great success. Because of this, this problem type needs further attention. In particular, more research effort should be dedicated to finding diversified initial solutions for it. This issue was not considered in the existing papers.

Acknowledgment

Malek Sarhani was supported by the Alexander von Humboldt Foundation.

Open access funding enabled and organized by Projekt DEAL.

References

- Ahuja, R.K., Orlin, J.B., Tiwari, A., 2000. A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research* 27, 10, 917–934.
- Alfonzetti, S., Dilettos, E., Salerno, N., 2006. Simulated annealing with restarts for the optimization of electromagnetic devices. *IEEE Transactions on Magnetics* 42, 4, 1115–1118.
- Alhalabi, W., Dragoi, E.N., 2017. Influence of randomization strategies and problem characteristics on the performance of differential search algorithm. *Applied Soft Computing* 61, 88–110.
- Aoun, O., Sarhani, M., El Afia, A., 2018. Particle swarm optimisation with population size and acceleration coefficients adaptation using hidden Markov model state classification. *International Journal of Metaheuristics* 7, 1, 1–29.
- Aringhieri, R., Cordone, R., Melzani, Y., 2007. Tabu search versus GRASP for the maximum diversity problem. *4OR* 6, 1, 45–60.
- de Armas, J., Lalla-Ruiz, E., Tilahun, S.L., Voß, S., 2022. Similarity in metaheuristics: a gentle step towards a comparison methodology. *Natural Computing* 21, 265–287.
- Azad, S.K., 2017. Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. *Engineering Optimization* 50, 1, 89–105.

- Bajer, D., Martinović, G., Brest, J., 2016. A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates. *Expert Systems with Applications* 60, 294–310.
- Battiti, R., Brunato, M., 2018. Reactive search: machine learning for memory-based heuristics. *Handbook of Approximation Algorithms and Metaheuristics* (2nd edn). CRC, Boca Raton, FL, pp. 327–344.
- Battiti, R., Tecchiolli, G., 1994. The reactive tabu search. *ORSA Journal on Computing* 6, 2, 126–140.
- Bellaachia, A., Alathel, D., 2014. A local pheromone initialization approach for ant colony optimization algorithm. *IEEE International Conference on Progress in Informatics and Computing*. IEEE, Piscataway, NJ, pp. 133–138.
- Bennis, F., Bhattacharjya, R.K., 2020. *Nature-Inspired Methods for Metaheuristics Optimization*. Springer, Berlin.
- Birattari, M., 2006. *Tuning Metaheuristics: A Machine Learning Perspective*, Studies in Computational Intelligence, Vol. 197. Springer, Berlin.
- Birattari, M., Pellegrini, P., Dorigo, M., 2007. On the invariance of ant colony optimization. *IEEE Transactions on Evolutionary Computation* 11, 6, 732–742.
- Blum, C., Dorigo, M., 2004. The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* 34, 2, 1161–1172.
- Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization. *ACM Computing Surveys* 35, 3, 268–308.
- Boese, K.D., Kahng, A.B., Muddu, S., 1994. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* 16, 2, 101–113.
- Brandão, J., 2006. A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research* 173, 2, 540–555.
- Cai, Q., Hang, W., Mirhoseini, A., Tucker, G., Wang, J., Wei, W., 2019. Reinforcement learning driven heuristic optimization. <https://arxiv.org/pdf/1906.06639.pdf>.
- Camacho-Villalón, C.L., Dorigo, M., Stützle, T., 2023. Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors. *International Transactions in Operational Research* 30, 2945–2971.
- Camacho Villalón, C.L., Stützle, T., Dorigo, M., 2020. Grey wolf, firefly and bat algorithms: three widespread algorithms that do not contain any novelty. *Lecture Notes in Computer Science* 12421, 121–133.
- Campos, V., Glover, F., Laguna, M., Martí, R., 2001. An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization* 21, 4, 397–414.
- Caserta, M., Voß, S., 2009. Metaheuristics: intelligent problem solving. In *Matheuristics*. Springer, Berlin, pp. 1–38.
- Caserta, M., Voß, S., 2014. A hybrid algorithm for the DNA sequencing problem. *Discrete Applied Mathematics* 163, 87–99.
- Caserta, M., Voß, S., Sniedovich, M., 2011. Applying the corridor method to a blocks relocation problem. *OR Spectrum* 33, 4, 915–929.
- Charon, I., Hudry, O., 2001. The noising methods: a generalization of some metaheuristics. *European Journal of Operational Research* 135, 1, 86–101.
- Chen, J., Xin, B., Peng, Z., Dou, L., Zhang, J., 2009. Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans* 39, 3, 680–691.
- Dai, Q., Ji, J., Liu, C., 2009. An effective initialization strategy of pheromone for ant colony optimization. *Fourth International Conference on Bio-Inspired Computing*. IEEE, Piscataway, NJ, pp. 1–4.
- Danna, E., Woodruff, D.L., 2009. How to select a small set of diverse solutions to mixed integer programming problems. *Operations Research Letters* 37, 4, 255–260.
- Dengiz, B., Altıparmak, F., Smith, A., 1997. Local search genetic algorithm for optimal design of reliable networks. *IEEE Transactions on Evolutionary Computation* 1, 3, 179–188.
- Diaz-Gomez, P.A., Hougen, D.F., 2007. Initial population for genetic algorithms: A metric approach. *2007 International Conference on Genetic and Evolutionary Methods*, Las Vegas, Nevada, pp. 43–49.
- Dorigo, M., Gambardella, L., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1, 1, 53–66.
- Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review* 41, 4, 637–676.

- Duin, C., Volgenant, A., 2006. Some inverse optimization problems under the Hamming distance. *European Journal of Operational Research* 170, 3, 887–899.
- 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.
- Elsayed, S., Sarker, R., Coello, C.A.C., 2017. Sequence-based deterministic initialization for evolutionary algorithms. *IEEE Transactions on Cybernetics* 47, 9, 2911–2923.
- Elsayed, S.M., Sarker, R.A., Essam, D.L., 2012. The influence of the number of initial feasible solutions on the performance of an evolutionary optimization algorithm. *Lecture Notes in Computer Science* 7673, 1–11.
- Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6, 2, 109–133.
- Fischetti, M., Monaci, M., 2014. Exploiting erraticism in search. *Operations Research* 62, 1, 114–122.
- Friedrich, T., Wagner, M., 2015. Seeding the initial population of multi-objective evolutionary algorithms: a computational study. *Applied Soft Computing* 33, 223–230.
- Fuad, M.M.M., 2013. A preinitialization stage of population-based bio-inspired metaheuristics for handling expensive optimization problems. In *Advanced Data Mining and Applications*. Springer, Berlin, pp. 396–403.
- Fuertes, G., Vargas, M., Alfaro, M., Soto-Garrido, R., Sabattin, J., Peralta, M.A., 2019. Chaotic genetic algorithm and the effects of entropy in performance optimization. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29, 1, 013132.
- Gagnon, I., April, A., Abran, A., 2021. An investigation of the effects of chaotic maps on the performance of metaheuristics. *Engineering Reports* 3, 8.
- Gao, W.F., Liu, S.Y., Huang, L.L., 2012. Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique. *Communications in Nonlinear Science and Numerical Simulation* 17, 11, 4316–4327.
- Ge, L., Voß, S., Xie, L., 2022. Robustness and disturbances in public transport. *Public Transport* 14, 191–261.
- Georgieva, A., Jordanov, I., 2009. Global optimization based on novel heuristics, low-discrepancy sequences and genetic algorithms. *European Journal of Operational Research* 196, 2, 413–422.
- Ghannami, A., Li, J., Hawbani, A., Al-Dubai, A., 2021. Stratified opposition-based initialization for variable-length chromosome shortest path problem evolutionary algorithms. *Expert Systems with Applications* 170, 114525.
- Ghobadi, K., Mahmoudzadeh, H., 2021. Inferring linear feasible regions using inverse optimization. *European Journal of Operational Research* 290, 3, 829–843.
- Glover, F., Hao, J.K., 2019. Diversification-based learning in computing and optimization. *Journal of Heuristics* 25, 4–5, 521–537.
- Glover, F., Kuo, C.C., Dhir, K.S., 1998. Heuristic algorithms for the maximum diversity problem. *Journal of Information and Optimization Sciences* 19, 1, 109–132.
- Glover, F., Laguna, M., 1997. *Tabu Search*. Springer US, New York.
- Glover, F., Samorani, M., 2019. Intensification, diversification and learning in metaheuristic optimization. *Journal of Heuristics* 25, 4, 517–520.
- Gonçalves, J.F., Resende, M.G.C., 2010. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics* 17, 5, 487–525.
- Gong, M., Jiao, L., Liu, F., Ma, W., 2009. Immune algorithm with orthogonal design based initialization, cloning, and selection for global optimization. *Knowledge and Information Systems* 25, 3, 523–549.
- Greistorfer, P., Løkketangen, A., Voß, S., Woodruff, D.L., 2008. Experiments concerning sequential versus simultaneous maximization of objective function and distance. *Journal of Heuristics* 14, 6, 613–625.
- Greistorfer, P., Voß, S., 2005. Controlled pool maintenance for metaheuristics. In *Metaheuristic Optimization Via Memory and Evolution*. Kluwer, Boston, MA, pp. 387–424.
- György, A., Kocsis, L., 2011. Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research* 41, 407–444.
- Hao, J.K., Solnon, C., 2020. Meta-heuristics and artificial intelligence. In *A Guided Tour of Artificial Intelligence Research*. Springer, Berlin, pp. 27–52.
- Hart, J.P., Shogan, A.W., 1987. Semi-greedy heuristics: an empirical study. *Operations Research Letters* 6, 3, 107–114.

- Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., Prasath, V.B.S., 2019. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information* 10, 12, 390.
- Helwig, S., Wanka, R., 2008. Theoretical analysis of initial particle swarm behavior. In *Parallel Problem Solving from Nature*. Springer, Berlin, pp. 889–898.
- Henderson, D., Jacobson, S.H., Johnson, A.W., 2003. The theory and practice of simulated annealing. In *Handbook of Metaheuristics*. Kluwer, Dordrecht, pp. 287–319.
- Ho, S.C., Gendreau, M., 2006. Path relinking for the vehicle routing problem. *Journal of Heuristics* 12, 1–2, 55–72.
- Hsu, T.C., Phoa, F.K.H., 2018. A smart initialization on the swarm intelligence based method for efficient search of optimal minimum energy design. International Conference on Swarm Intelligence, Springer, Berlin, pp. 78–87.
- Iliopoulou, C., Kepaptsoglou, K., Vlahogianni, E., 2019. Metaheuristics for the transit route network design problem: a review and comparative analysis. *Public Transport* 11, 487–521.
- Jana, N.D., Sil, J., Das, S., 2016. Continuous fitness landscape analysis using a chaos-based random walk algorithm. *Soft Computing* 22, 3, 921–948.
- Jo, T., 2021. *Machine Learning Foundations*. Springer International, Berlin.
- Jovanovic, R., Tuba, M., Voß, S., 2019. Fixed set search applied to the traveling salesman problem. *Lecture Notes in Computer Science* 11299, 63–77.
- Jovanovic, R., Voß, S., 2021. Fixed set search application for minimizing the makespan on unrelated parallel machines with sequence-dependent setup times. *Applied Soft Computing* 110, 107521.
- Juman, Z., Hoque, M., 2015. An efficient heuristic to obtain a better initial feasible solution to the transportation problem. *Applied Soft Computing* 34, 813–826.
- Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2012. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review* 42, 1, 21–57.
- Karagul, K., Sahin, Y., 2020. A novel approximation method to obtain initial basic feasible solution of transportation problem. *Journal of King Saud University—Engineering Sciences* 32, 3, 211–218.
- Kazimipour, B., Li, X., Qin, A.K., 2014. A review of population initialization techniques for evolutionary algorithms. In 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp. 2585–2592.
- Keedwell, E., Breuvilliers, M., Idoumghar, L., Lepagnot, J., Rakhshani, H., 2018. A novel population initialization method based on support vector machine. IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, Piscataway, NJ, pp. 751–756.
- Kim, S.H., Boukouvala, F., 2020. Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. *Optimization Letters* 14, 4, 989–1010.
- Kimura, S., Matsumura, K., 2005. Genetic algorithms using low-discrepancy sequences. Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, pp. 1341–1346.
- Krömer, P., Zelinka, I., Snášel, V., 2014. Behaviour of pseudo-random and chaotic sources of stochasticity in nature-inspired optimization methods. *Soft Computing* 18, 4, 619–629.
- Laguna, M., Glover, F., 1993. Integrating target analysis and tabu search for improved scheduling systems. *Expert Systems with Applications* 6, 3, 287–297.
- Laguna, M., Martí, R., 2003. Scatter search. In *Metaheuristic Procedures for Training Neural Networks*. Springer US, New York, pp. 139–152.
- Lalla-Ruiz, E., Voß, S., 2016. Improving solver performance through redundancy. *Journal of Systems Science and Systems Engineering* 25, 3, 303–325.
- Leung, Y.W., Wang, Y., 2000. Multiobjective programming using uniform design and genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30, 3, 293–304.
- Leung, Y.W., Wang, Y., 2001. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 5, 1, 41–53.
- Li, Q., Bai, Y., Gao, W., 2021a. Improved initialization method for metaheuristic algorithms: a novel search space view. *IEEE Access* 9, 121366–121384.
- Li, Q., Liu, S.Y., Yang, X.S., 2020. Influence of initialization on the performance of metaheuristic optimizers. *Applied Soft Computing* 91, 106193.
- Li, S., Yang, S., Wang, Y., Yue, W., Qiao, J., 2021b. A modular neural network-based population prediction strategy for evolutionary dynamic multi-objective optimization. *Swarm and Evolutionary Computation* 62, 100829.

- de Lima Junior, F.C., de Melo, J.D., Neto, A.D.D., 2007. Using q-learning algorithm for initialization of the GRASP metaheuristic and genetic algorithm. *International Joint Conference on Neural Networks*. IEEE, Piscataway, NJ, pp. 1243–1248.
- Liu, H., Abraham, A., Clerc, M., 2007. Chaotic dynamic characteristics in swarm intelligence. *Applied Soft Computing* 7, 3, 1019–1026.
- Lozano, M., García-Martínez, C., 2010. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research* 37, 3, 481–497.
- Ma, Z., Vandenbosch, G.A., 2012. Impact of random number generators on the performance of particle swarm optimization in antenna design. 6th European conference on antennas and propagation (EUCAP). IEEE, Piscataway, NJ, pp. 925–929.
- Maaranen, H., Miettinen, K., Mäkelä, M.M., 2004. Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications* 47, 12, 1885–1895.
- Maaranen, H., Miettinen, K., Penttinen, A., 2006. On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization* 37, 3, 405–436.
- Mahdavi, S., Rahnamayan, S., Deb, K., 2016. Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization. 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, Piscataway, NJ, pp. 3557–3565.
- Malan, K.M., Engelbrecht, A.P., 2013. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* 241, 148–163.
- Malan, K.M., Engelbrecht, A.P., 2014. A progressive random walk algorithm for sampling continuous fitness landscapes. IEEE Congress on Evolutionary Computation (CEC). IEEE, Piscataway, NJ, pp. 2507–2514.
- Malone, B., Kangas, K., Järvisalo, M., Koivisto, M., Myllymäki, P., 2018. Empirical hardness of finding optimal Bayesian network structures: algorithm selection and runtime prediction. *Machine Learning* 107, 1, 247–283.
- Martí, R., Moreno-Vega, J.M., Duarte, A., 2010. Advanced multi-start methods. In *Handbook of Metaheuristics*. Springer US, New York, pp. 265–281.
- Martí, R., Resende, M.G., Ribeiro, C.C., 2013. Multi-start methods for combinatorial optimization. *European Journal of Operational Research* 226, 1, 1–8.
- Matsumoto, M., Nishimura, T., 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 8, 1, 3–30.
- Mehdi, M., Melab, N., Talbi, E.G., Bouvry, P., 2010. Interval-based initialization method for permutation-based problems. IEEE Congress on Evolutionary Computation. IEEE, Piscataway, NJ, pp. 1–8.
- de Melo, V.V., Delbem, A.C.B., 2012. Investigating smart sampling as a population initialization method for differential evolution in continuous problems. *Information Sciences* 193, 36–53.
- Molina, D., Poyatos, J., Ser, J.D., García, S., Hussain, A., Herrera, F., 2020. Comprehensive taxonomies of nature- and bio-inspired optimization: Inspiration versus algorithmic behavior, critical analysis recommendations. *Cognitive Computation* 12, 5, 897–939.
- Mousavirad, S.J., Bidgoli, A.A., Rahnamayan, S., 2019. Tackling deceptive optimization problems using opposition-based DE with center-based Latin hypercube initialization. 14th International Conference on Computer Science & Education (ICCSE), IEEE, Piscataway, NJ.
- Muelas, S., Peña, J.M., LaTorre, A., Robles, V., 2010. A new initialization procedure for the distributed estimation of distribution algorithms. *Soft Computing* 15, 4, 713–720.
- Mun, J., 2015. Understanding and choosing the right probability distributions. In *Advanced Analytical Models: Over 800 Models and 300 Applications from the Basel II Accord to Wall Street and Beyond*. John Wiley & Sons, Inc., Hoboken, NJ, pp. 899–917.
- Muñoz, M.A., Kirley, M., Halgamuge, S.K., 2014. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Transactions on Evolutionary Computation* 19, 1, 74–87.
- Neveu, B., Trombettoni, G., Glover, F., 2004. ID walk: a candidate list strategy with a simple diversification device. In *Principles and Practice of Constraint Programming – CP 2004*. Springer, Berlin, pp. 423–437.
- Ochoa, G., Veerapen, N., 2017. Mapping the global structure of TSP fitness landscapes. *Journal of Heuristics* 24, 3, 265–294.

- Oliker, N., Bekhor, S., 2020. An infeasible start heuristic for the transit route network design problem. *Transportmetrica A: Transport Science* 16, 3, 388–408.
- Oliveira, A.C.M., Lorena, L.A.N., 2007. Hybrid evolutionary algorithms and clustering search. In *Hybrid Evolutionary Algorithms*. Springer, Berlin, pp. 77–99.
- Oliveto, P.S., Sudholt, D., Zarges, C., 2019. On the benefits and risks of using fitness sharing for multimodal optimisation. *Theoretical Computer Science* 773, 53–70.
- Oman, S., Cunningham, P., 2001. Using case retrieval to seed genetic algorithms. *International Journal of Computational Intelligence and Applications* 1, 1, 71–82.
- Ozer, A.B., 2010. CIDE: Chaotically initialized differential evolution. *Expert Systems with Applications* 37, 6, 4632–4641.
- Park, S.Y., Lee, J.J., 2016. Stochastic opposition-based learning using a beta distribution in differential evolution. *IEEE Transactions on Cybernetics* 46, 10, 2184–2194.
- Park, T., Ryu, K.R., 2010. A dual-population genetic algorithm for adaptive diversity control. *IEEE Transactions on Evolutionary Computation* 14, 6, 865–884.
- Paul, P.V., Moganarangan, N., Kumar, S.S., Raju, R., Vengattaraman, T., Dhavachelvan, P., 2015. Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: an empirical study based on traveling salesman problems. *Applied Soft Computing* 32, 383–402.
- Poikolainen, I., Neri, F., Caraffini, F., 2015. Cluster-based population initialization for differential evolution frameworks. *Information Sciences* 297, 216–235.
- Prugel-Bennett, A., Tayarani-Najaran, M.H., 2012. Maximum satisfiability: anatomy of the fitness landscape for a hard combinatorial optimization problem. *IEEE Transactions on Evolutionary Computation* 16, 3, 319–338.
- Przewoźniczek, M.W., 2020. Subpopulation initialization driven by linkage learning for dealing with the long-way-to-stuck effect. *Information Sciences* 521, 62–80.
- Rahnamayan, S., Tizhoosh, H.R., Salama, M.M., 2007. A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications* 53, 10, 1605–1614.
- Rahnamayan, S., Tizhoosh, H.R., Salama, M.M., 2008a. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation* 12, 1, 64–79.
- Rahnamayan, S., Tizhoosh, H.R., Salama, M.M., 2008b. Opposition versus randomness in soft computing techniques. *Applied Soft Computing* 8, 2, 906–918.
- Rahnamayan, S., Wang, G.G., 2009. Toward effective initialization for large-scale search spaces. *WSEAS Transactions on Systems* 8, 3, 355–367.
- Ramsey, C.L., Grefenstette, J.J., 1993. Case-based initialization of genetic algorithms. Proceedings of the 5th International Conference on Genetic Algorithms. Morgan Kaufmann, Burlington, MA, pp. 84–91.
- Reeves, C.R., 1999. Landscapes, operators and heuristic search. *Annals of Operations Research* 86, 473–490.
- Ren, Z., Pang, B., Wang, M., Feng, Z., Liang, Y., Chen, A., Zhang, Y., 2018. Surrogate model assisted cooperative coevolution for large scale optimization. *Applied Intelligence* 49, 2, 513–531.
- Resende, M.G.C., Ribeiro, C.C., 2018. Greedy randomized adaptive search procedures: advances and extensions. In *Handbook of Metaheuristics*. Springer, Berlin, pp. 169–220.
- Sacco, W., Henderson, N., 2011. Finding all solutions of nonlinear systems using a hybrid metaheuristic with fuzzy clustering means. *Applied Soft Computing* 11, 8, 5424–5432.
- Sacco, W.F., Rios-Coelho, A.C., 2018. On initial populations of differential evolution for practical optimization problems. In *Computational Intelligence, Optimization and Inverse Problems with Applications in Engineering*. Springer International, Berlin, pp. 53–62.
- Samorani, M., Wang, Y., Wang, Y., Lv, Z., Glover, F., 2019. Clustering-driven evolutionary algorithms: an application of path relinking to the quadratic unconstrained binary optimization problem. *Journal of Heuristics* 25, 4–5, 629–642.
- Sapkal, S., Laha, D., 2011. Comparison of initial solutions of heuristics for no-wait flow shop scheduling. International Conference on Computational Intelligence and Information Technology, Springer, Berlin, pp. 294–298.
- Sarhani, M., Voß, S., 2022. Chunking and cooperation in particle swarm optimization for feature selection. *Annals of Mathematics and Artificial Intelligence* 90, 893–913.
- Saroj, K., Kumar, D., Kanika, 2011. A genetic algorithm with entropy based probabilistic initialization and memory for automated rule mining. International Conference on Computer Science and Information Technology, Springer, Berlin, pp. 604–613.

- Shatnawi, M., Nasrudin, M.F., 2011. Starting configuration of cuckoo search algorithm using centroidal Voronoi tessellations. 2011 11th International Conference on Hybrid Intelligent Systems (HIS). IEEE, Piscataway, NJ, pp. 40–45.
- Shi, Y., Eberhart, R.C., 1999. Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Vol. 3. IEEE, Piscataway, NJ, pp. 1945–1950.
- Simon, D., Ergezer, M., Du, D., Rarick, R., 2011. Markov models for biogeography-based optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41, 1, 299–306.
- Simon, D., Omran, M.G., Clerc, M., 2014. Linearized biogeography-based optimization with re-initialization and local search. *Information Sciences* 267, 140–157.
- Skokos, C., 2010. The Lyapunov characteristic exponents and their computation. In *Dynamics of Small Solar System Bodies and Exoplanets*. Springer, Berlin, pp. 63–135.
- Snaselova, P., Zboril, F., 2015. Genetic algorithm using theory of chaos. *Procedia Computer Science* 51, 316–325.
- Sniedovich, M., Voß, S., 2006. The corridor method: a dynamic programming inspired metaheuristic. *Cybernetics* 35, 3, 551–578.
- Sobol', I.M., Asotsky, D., Kreinin, A., Kucherenko, S., 2011. Construction and comparison of high-dimensional Sobol' generators. *Wilmott* 56, 64–79.
- Sondergeld, L., Voß, S., 1996. A star-shaped diversification approach in tabu search. In *Meta-Heuristics*. Springer US, New York, pp. 489–502.
- Sondergeld, L., Voß, S., 1999. Cooperative intelligent search using adaptive memory techniques. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer US, New York, pp. 297–312.
- Sörensen, K., 2015. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research* 22, 1, 3–18.
- Sörensen, K., Glover, F.W., 2013. Metaheuristics. In *Encyclopedia of Operations Research and Management Science*. Springer US, New York, pp. 960–970.
- Stützle, T., Hoos, H.H., 2000. MAX–MIN ant system. *Future Generation Computer Systems* 16, 8, 889–914.
- Taillard, É.D., Gambardella, L.M., Gendreau, M., Potvin, J.Y., 2001. Adaptive memory programming: a unified view of metaheuristics. *European Journal of Operational Research* 135, 1, 1–16.
- Tian, D., 2017. Particle swarm optimization with chaos-based initialization for numerical optimization. *Intelligent Automation & Soft Computing* online available, 1–12. <https://doi.org/10.1080/10798587.2017.1293881>.
- Tian, J., Sun, C., Tan, Y., Zeng, J., 2020. Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization. *Knowledge-Based Systems* 187, 104815.
- Tizhoosh, H., 2005. Opposition-based learning: a new scheme for machine intelligence. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. IEEE, Piscataway, NJ, pp. 695–701.
- Tometzki, T., Engell, S., 2011. Systematic initialization techniques for hybrid evolutionary algorithms for solving two-stage stochastic mixed-integer programs. *IEEE Transactions on Evolutionary Computation* 15, 2, 196–214.
- Traonmilin, Y., Aujol, J.F., 2020. The basins of attraction of the global minimizers of the non-convex sparse spike estimation problem. *Inverse Problems* 36, 4, 045003.
- Trelea, I.C., 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters* 85, 6, 317–325.
- Tseng, L.Y., Chen, C., 2008. Multiple trajectory search for large scale global optimization. IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, Piscataway, NJ, pp. 3052–3059.
- Vargas, M., Fuertes, G., Alfaro, M., Gatica, G., Gutierrez, S., Peralta, M., 2018. The effect of entropy on the performance of modified genetic algorithm using earthquake and wind time series. *Complexity* 2018, Article ID 4392036, 1–13.
- Voß, S., 1993. Tabu search: applications and prospects. In *Network Optimization Problems: Algorithms, Applications and Complexity*. World Scientific, Singapore, pp. 333–353.
- Voß, S., 1995. Solving quadratic assignment problems using the reverse elimination method. In *The Impact of Emerging Technologies on Computer Science and Operations Research*. Springer US, New York, pp. 281–296.
- Voß, S., Fink, A., 2012. Hybridizing reactive tabu search with simulated annealing. *Lecture Notes in Computer Science* 7219, 509–512.
- Voß, S., Fink, A., Duin, C., 2005. Looking ahead with the pilot method. *Annals of Operations Research* 136, 1, 285–302.

- Voß, S., Martello, S., Osman, I.H., Roucairol, C., (eds.) 1999. *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Springer US, New York.
- Voß, S., Woodruff, D.L., 2003. Optimization software class libraries. In *Optimization Software Class Libraries*. Kluwer, Dordrecht, pp. 1–24.
- Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., Ventresca, M., 2011. Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences* 181, 20, 4699–4714.
- Wang, M., Li, B., Zhang, G., Yao, X., 2018. Population evolvability: dynamic fitness landscape analysis for population-based metaheuristic algorithms. *IEEE Transactions on Evolutionary Computation* 22, 4, 550–563.
- Watson, J.P., 2010. An introduction to fitness landscape analysis and cost models for local search. In *Handbook of Meta-heuristics*. Springer US, New York, pp. 599–623.
- Weise, T., Wu, Z., Wagner, M., 2019. An improved generic bet-and-run strategy with performance prediction for stochastic local search. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 2395–2402. <https://doi.org/10.1609/aaai.v33i01.33012395>.
- Weyland, D., 2015. A critical analysis of the harmony search algorithm—how not to solve Sudoku. *Operations Research Perspectives* 2, 97–105.
- Wolpert, D., Macready, W., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 1, 67–82.
- Zhan, Z.H., Zhang, J., Li, Y., Chung, H.H., 2009. Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 6, 1362–1381.
- Zhang, Q., Leung, Y.W., 1999. An orthogonal genetic algorithm for multimedia multicast routing. *IEEE Transactions on Evolutionary Computation* 3, 1, 53–62.
- Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E., 2007. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In *Lecture Notes in Computer Science*, Vol. 4403. Springer, Berlin, pp. 832–846.