CHAPTER

9 Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics

Eneko Osaba[a], Xin-She Yang[b], Javier Del Ser[a,c]
[a]TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Spain
[b]Middlesex University London, School of Science and Technology, London, United Kingdom
[c]University of the Basque Country (UPV/EHU), Bilbao, Spain

CONTENTS

Nature-Inspired Computation and Swarm Intelligence. https://doi.org/10.1016/B978-0-12-819714-1.00020-8

9.1 Introduction

In the current operations research and optimization communities, routing problems are one of the most studied paradigms. Two principal reasons that make this topic a paramount one in the field are (i) their inherent practical nature and social interest, which allow routing problems to be applicable not only in leisure or tourism scenarios, but also in situations related to logistics and business, and (ii) their complexity, making such problems very difficult to optimally solve even for medium-sized datasets. Arguably, the modeling and formulation of this kind of problems draws inspiration from real-world logistic and transportation situations, directly implying a social and/or business benefit in the case of its proper solution. Furthermore, the efficient addressing of these problems usually supposes a tough challenge for the scientific community because of their NP-hard nature. This fact leads related researchers to the adoption of diverse artificial intelligence solvers, aiming at solving them in a computationally affordable fashion. The problem gets even more involved when bearing in mind the rich literature with regard to different formulations of variants. Among this wide variety of problems, the traveling salesman problem (TSP) (Lawler et al., 1985) and the vehicle routing problem (VRP) (Christofides, 1976) are widely recognized as the most studied ones. This study is focused in the first of these problems, the TSP.

In line with this, many optimization approaches have been proposed along the years for dealing with the VRP. The three most studied and well-established schemes are exact methods (Laporte, 1992a,b), heuristics (Vaghela et al., 2018; Pozna et al., 2010), and metaheuristics. The present research is focused on the latter ones, which have demonstrated a remarkable efficiency for properly solving routing problems, especially in the last decade. The most recognized methods in this category could be simulated annealing (SA) (Kirkpatrick et al., 1983) and tabu search (TS) (Glover, 1989) as local search-based solvers, and ant colony optimization (ACO) (Bell and McMullen, 2004; Yu et al., 2009), particle swarm optimization (PSO) (Kennedy et al., 1995; Tang et al., 2015), and the genetic algorithm (GA) (Goldberg, 1989; De Jong, 1975) as population-based methods. In addition to these classical and recognized approaches, the design and implementation of new metaheuristics is a hot topic in the related operations research and optimization communities. As a result of this scientific trend, lots of successful solvers have been proposed in recent years, such as the bat algorithm (BA) (Yang, 2010), the firefly algorithm (FA) (Yang, 2009), the gravitational search algorithm (Rashedi et al., 2009; David et al., 2013), and fireworks algorithm optimization (FAO) (Tan and Zhu, 2010), among many others.

The main contribution of this work can be divided into three different points. First, we devote a comprehensive section for outlining the research carried out in recent years around the TSP problem, focusing our effort on its solving through the use of metaheuristic algorithms. Secondly, we take a step further over the state of the art in the elaboration of a new research direction: the hybridization of novelty search (NS) mechanisms and bio-inspired computation algorithms for solving the TSP. NS (Lehman and Stanley, 2008) was proposed in 2008 as a way to enhance the exploratory ability of population-based algorithmic solvers. After showing its great performance applied to several optimization problems, we hypothesize its promising performance also for the TSP. To this end, we have developed different versions of well-known swarm intelligence methods, namely, PSO, FA, and BA, and we evaluate in this chapter the performance of these metaheuristics embedding the NS mechanism on their basic scheme. We introduce through the chapter the descriptions on how these methods have been modeled for tackling the problem at hand and how the NS has been adapted for this discrete scenario. In order to assess the performance of each implemented solver, outcomes obtained in 15 instances are compared and discussed. Finally, an important additional contribution is our personal envisioned status of this field, which we present in the form of challenges and open opportunities that should be addressed in the near future.

The rest of this chapter is structured as follows. In Section 9.2 the TSP and some of its most important variants are described and mathematically formulated. Section 9.3 elaborates on the first contribution of the chapter by analyzing the recent research done around the TSP. In Section 9.4, the concepts behind NS are introduced, placing emphasis on how we have hybridized metaheuristic solvers with this mechanism. Considered heuristic solvers and their implementation details are described in Section 9.5. The experimental setup is detailed in Section 9.6, along with a discussion on the obtained results. Research opportunities for the area are highlighted in Section 9.7. Finally, Section 9.8 concludes the chapter with a general outlook for the wide audience.

9.2 Problem statement

In this chapter, our experimentation with NS and the chosen bio-inspired optimization methods is done using the basic version of the TSP. As can be read in many scientific works, the canonical TSP can be represented as a complete graph $\mathcal{G} \doteq (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} \doteq \{v_1, v_2, \ldots, v_N\}$ illustrates the vertex group that represents the nodes of the graph, and $\mathcal{A} \doteq \{(v_i, v_j) : v_i, v_j \in \mathcal{V} \times \mathcal{V},\ i, j \in \{1, \ldots, N\} \times \{1, \ldots, N\}, i \neq j\}$ is the group of edges linking every pair of nodes in $\mathcal{V}$. Moreover, each edge (v_i, v_j) has an associated cost $c_{ij} \in \mathbb{R}^+$, denoting the traveling weight of this arc. Because of the symmetric nature of the basic TSP, it is ensured that $c_{ij} = c_{ji}$, meaning that the cost of going from one v_i to another v_j is equal to the reverse trip (v_j, v_i).

Thus, the principal optimization objective of the TSP pivots on the discovery of a route that visits each node once and only once (i.e., a Hamiltonian cycle in graph G), minimizing the total cost of the whole route. This genetic problem can be mathematically formulated as

$$\underset{\mathbf{X}}{\text{minimize}} \quad f(\mathbf{X}) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} c_{ij} x_{ij} \tag{9.1a}$$

$$\text{subject to} \quad \sum_{\substack{j=1 \\ i \neq j}}^{N} x_{ij} = 1, \quad \forall j \in \{1, \ldots, N\}, \tag{9.1b}$$

$$\sum_{\substack{i=1 \\ i \neq j}}^{N} x_{ij} = 1, \quad \forall i \in \{1, \ldots, N\}, \tag{9.1c}$$

$$\sum_{\substack{i \in \mathcal{S} \\ j \in \mathcal{S} \\ i \neq j}} x_{ij} \geq 1, \quad \forall \mathcal{S} \subset \mathcal{V}, \tag{9.1d}$$

where $\mathbf{X} \doteq [x_{ij}]$ is an $N \times N$ binary matrix whose entry $x_{ij} \in \{0, 1\}$ takes value 1 if edge (i, j) is used in the solution. Furthermore, the objective function is represented in Eq. (9.1a) as the sum of costs associated to all the edges in the solution. Moreover, Eqs. (9.1b) and (9.1c) depict that each vertex must be visited once and only once. Lastly, (9.1d) guarantees the absence of subtours and forces that any subset of nodes S has to be abandoned at least one time. This restriction is needed to prevent the existence of subtours on the whole route.

Apart from this canonical formulation of the TSP, many different variants have been modeled over the years, aiming at adapting and addressing different characteristics present in the logistics and transportation world. We list here some of the most famous advanced variants of the TSP:

- *The asymmetric TSP* (ATSP) (Asadpour et al., 2017; Svensson, 2018): The central characteristic of the TSP is that, although there may be arcs where $c_{ij} = c_{ji}$, in general $c_{ij} \neq c_{ji}$.
- *The multiple TSP* (M-TSP) (Kitjacharoenchai et al., 2019; Rostami et al., 2015): In the M-TSP, a set of m exact salesmen are available, which should visit a set of n cities starting and ending at the same city.
- *The TSP with time windows* (TSPTW) (Roberti and Wen, 2016; Fachini and Armentano, 2018): In this variant, the traveling salesman should visit each vertex respecting a time window fixed by each separated node.
- *The time-dependent TSP* (TDTSP) (Arigliano et al., 2019; Furini et al., 2016): The basic idea behind the TDTSP is that the cost of traveling between two different nodes is time-dependent. In other words, travel times could significantly change over the day, for example, during *peak* and *off-peak* hours.
- *The generalized TSP* (GTSP) (Smith and Imeson, 2017; Helsgaun, 2015): In the GTSP, the set of nodes is partitioned into different clusters. The main goal of this formulation is to find a minimum cost tour passing through exactly one node from each cluster.

Other interesting research activity can be detected around the rich TSP (R-TSP), also known as multiattribute TSP (Caceres-Cruz et al., 2015). This type of problems are specific cases of the TSP with complex formulations and multiple restrictions. The principal characteristic of an R-TSP is its complex formulation, which is composed by multiple constraints. This feature directly leads to an increased complexity of resolution, which entails to a major scientific challenge at the same time. These problems are especially important in the current community because they model many real-world problems. Accordingly, the efficient solution of the R-TSP can be useful in many valuable real-world applications. Some remarkable examples can be found in Lahyani et al. (2017), Osaba et al. (2015), or Maity et al. (2019).

As can be seen, the number of TSP variants proposed in the literature is overwhelming, making the listing of all interesting and valuable formulation in this section infeasible. For this reason, we have outlined some of the most commonly used ones, with the intention of settling the idea that there is a vibrant scientific activity behind this problem.

9.3 Recent advances in the traveling salesman problem

Since its formulation, the TSP has become one of the most employed benchmarking problems in performance analysis of discrete optimization algorithms. A plethora of methods have been applied to the TSP and its variants in recent decades. We can highlight classical methods such as GA (Grefenstette et al., 1985; Larrañaga et al., 1999), TS (Fiechter, 1994; Knox, 1994; Gendreau et al., 1998), or SA (Malek et al., 1989; Aarts et al., 1988). Besides these classical algorithms, more recent and effective approaches have been extensively used for solving the TSP, such as ACO (Dorigo and Gambardella, 1997; Jun-man and Yi, 2012), PSO (Clerc, 2004; Shi et al., 2007), or the variable neighborhood search (Carrabs et al., 2007; Burke et al., 2001). In addition to these well-known methods, the TSP and its multiple derivations have been the focus of many benchmarking studies for measuring the quality of many recently proposed nature-inspired methods. Some examples are FA (Kumbharana and Pandey, 2013), CS (Ouaarab et al., 2014), ICA (Yousefikhoshbakht and Sedighpour, 2013), the well-reputed artificial bee colony (ABC) (Karaboga and Gorkemli, 2011), and honey bee mating optimization (Marinakis et al., 2011).

As can be seen, the TSP has been extensively used by operation research and computational intelligence researchers since its formulation for different purposes. The state of the art around this problem is such wide that, in this section, we will focus our attention on highlighting the research and advances conducted in the last few years. Being aware that the related literature is bigger than what is represented in this systematic review, we refer interested readers to surveys such as (Potvin, 1993; Laporte, 1992a; Bellmore and Nemhauser, 1968; Lust and Teghem, 2010; Matai et al., 2010).

9.3.1 TSP and genetic algorithms

GA has been adopted by many authors of the current scientific community for the TSP and its variants. In Dong and Cai (2019), for example, we can find an interesting GA for solving the challenging large-scale colored balanced TSP. In Lo et al. (2018), Lo et al. explored the adaptation of GA to the real-life oriented multiple TSP. Additional common practice related to GA applied to TSP is the formulation of novel operators, as can be seen in Hussain et al. (2019), in which a new crossover operator is proposed. In Roy et al. (2019) a so-called multiparent crossover is formulated, which shares some notions with that proposed in Sakai et al. (2018), called edge assembly crossover. An additional valuable example of this practice can be found in Wang et al. (2016), in which a multioffspring GA is proposed. Authors of that research claim that, in the basic versions of GA, the number of generated offsprings is the same as the number of parents. Thus, they explore the concept that, for the survival and diversity of the species, it should be desirable to generate a greater number of offsprings. In Hussain et al. (2017), Hussain et al. built an effective combination function called modified cycle crossover operator. In Liu and Li (2018), authors present a new method to initialize the population of GA for the TSP, called greedy permuting method. Another initialization strategy is developed in Deng et al. (2015), which rests its inspiration in the well-known k-mean algorithm. As can be easily checked, the literature around TSP and this successful method is abundant nowadays, being the subject of a myriad of works year by year. Interested readers are referred to additional outstanding works such as Bolaños et al. (2015), Groba et al. (2015), and Contreras-Bolton and Parada (2015).

9.3.2 TSP and simulated annealing

Despite being a classic method, SA is still the focus of much research around the TSP and its variants. In Ezugwu et al. (2017), for example, Ezugwu et al. proposed a hybrid metaheuristic for solving the TSP based on SA and the recently proposed symbiotic organisms search method. Zhan et al. (2016) presented the coined list-based SA, the main basis of which rests on a new mechanism for controlling the temperature parameter. The implemented method counts with a list of temperatures, the maximum of which is used by a Metropolis acceptance criterion to decide whether to accept a candidate solution. Furthermore, the temperature list is dynamically adapted according to the solution space of the problem. In the short paper published by Osaba et al. (2016a), an evolutionary SA is developed for the TSP and compared with additional metaheuristics such as the TS. An additional interesting study can be found in Wu and Gao (2017), in which the performance of the basic SA is improved through the use of a greedy search mechanism for properly dealing with the large-scale TSP. A very recent study can be found in Zhou et al. (2019), in which SA is employed for increasing the population diversity of a gene expression programming method, aiming to improve the ability of the search. Additional valuable related research can be found in papers such as Liu and Zhang (2018), Xu et al. (2017), and Makuchowski (2018).

9.3.3 TSP and tabu search

Among the classic methods, TS is probably the one that has suffered most during the course of time. In the past decades, TS was a successful method considered a cornerstone in the combinatorial optimization and TSP scientific community. Over the years, sophisticated methods laid aside the TS, and today it is difficult to find remarkable studies around the figure of the TS. Among these few studies, we can find as one of the most representative ones the research conducted by Lin et al. (2016), who proposed a TS-SA hybrid solver for the symmetric TSP. One of the characteristics of this hybrid scheme is the development of a dynamic neighborhood structure, the principal goal of which is the enhancement of the search efficiency of the method, by means of the randomness reduction of the conventional 2-opt neighborhood. In Osaba et al. (2018d), TS is employed as part of a pool of metaheuristics for solving the open-path asymmetric green TSP. The main objective of this multiattribute TSP variant is to find a route between a fixed origin and destination, visiting a group of intermediate points exactly once, minimizing the CO_2 emitted by the car and the total distance traveled. An additional analysis can be found in Xu et al. (2015). Among other aspects, authors of that study not only explore the efficiency of four different version of TS using different tabu mechanisms, but also the synergy between TS and ACO in a hybrid method.

9.3.4 TSP and ant colony optimization

Conversely to TS, one of the most used methods in recent years in the TSP community is ACO, as can be seen in works such as Ariyasingha and Fernando (2015) and Mahi et al. (2015). In the first of these works, the recently proposed multiobjective ACO is used for solving the multiobjective TSP under different configurations, using two, three, and four objectives, and different numbers of ants and iterations. In the second research, a hybrid approach is presented which employs PSO for optimizing the parameters that affect performance of the ACO algorithm. Additionally, a 3-opt heuristic is endowed to the proposed method for improving local solutions. A similar solver is implemented in Gülcü et al. (2018), called PACO-3Opt. This hybrid parallel and cooperative method, which counts with multiple colonies and a master–slave paradigm, employs also the 3-opt function for avoiding local minima. An additional well-reputed study can be found in Mavrovouniotis et al. (2016). The principal value of this work is its application to the dynamic TSP. For properly dealing with the unstable nature of this instance of the problem, authors endowed the ACO with a local search operator (called unstring and string) which iteratively takes the best solution found by the algorithm and removes/inserts cities in such a way that the solution quality is improved. The same dynamic formulation of the problem is also used in Chowdhury et al. (2018), implementing a variant of ACO in combination with an adaptive large neighborhood search. Moreover, a note-worthy multiobjective version of ACO is presented in Zhang et al. (2016) for solving the biobjective TSP. One of the essential characteristics of the proposed algorithm is the initialization of a pheromone matrix with the prior knowledge of a *Physarum*-inspired mathematical model. Additional interesting works focusing on ACO can be found in Pang et al. (2015), Eskandari et al. (2019), and Zaidi and Gupta (2018). For readers interested in memetic or hybrid approaches, studies such as Sahana et al. (2018), Liao and Liu (2018), and Dahan et al. (2019) are highly recommended.

9.3.5 TSP and particle swarm optimization

Since its introduction in 1995 by Eberhart and Kennedy, PSO has become the most used technique in the swarm intelligence field, and one of its main influential representatives. PSO was developed under the inspiration of the behavior of bird flocks, fish schools, and human communities, and although it was not initially designed to be applied to discrete problems, several modifications have made it possible. Regarding the TSP, lots of papers have been devoted to its application in the last decade; we highlight contributions such as Wang et al. (2003), Clerc (2004), and Pang et al. (2004). Focusing our attention on the research conducted in the most recent years, we can highlight the work introduced in Zhong et al. (2018), in which PSO in combination with a Metropolis acceptance criterion is implemented. The fundamental reason of this merge is to enhance PSO to escape from premature convergence, endowing the method with a sophisticated mechanism to decide whether or not to accept newly produced solutions. A highly interesting research was published by Marinakis et al. (2015), in which a probabilistic TSP was tackled through an adaptive multiswarm PSO. In that adaptive PSO, random values are assigned in the initial phase of the search. After that, these parameters are dynamically optimized simultaneously with the optimization of the objective function of the problem. An additional improved PSO is proposed in Khan et al. (2018) for solving the imprecise cost matrix TSP. Main modifications of the modeled PSO consist on adopting the swap sequence, swap operation, and different velocity update rules. Another interesting example of enhanced PSO can be found in Yu et al. (2015), in which the multiobjective TSP is solved using the so-called set-based comprehensive learning PSO. Further recently published works include Wang and Xu (2017), Chang (2016), and Akhand et al. (2016).

9.3.6 TSP and bat algorithm

Focusing our attention on recently proposed nature-inspired methods, and starting with the well-known BA, we can find an interesting recent study by Al-Sorori et al. (2016), in which a hybrid approach is proposed in combination with the genetic operators crossover and mutation, and using the 2-opt and 3-opt operators as local search mechanisms for improving searching performance and speeding up the convergence. Another interesting approach is proposed in Saji and Riffi (2016), whose principal contribution is the velocity scheme used, represented as the number of permutations needed for a bat to reach the best candidate of the swarm. A similar alternative was presented in Jiang (2016), employing nearest neighbor tour construction heuristics for initializing the population and the 2-opt edge-exchange algorithm for the local search step of the method. Among all these papers, we should highlight the research proposed in Osaba et al. (2016b), which is not only considered as the first adaptation of BA to the TSP and asymmetric TSP problems, but also the most cited one. Several aspects make this study interesting, such as the use of the well-known Hamming distance as distance function or its *inclination* mechanism, allowing the method to modify the solution space scheme along the running.

9.3.7 TSP and firefly algorithm

If we turn our attention to FA, we can highlight the research recently proposed in Mohsen and Al-Sorori (2017). Authors of this study clearly based their work on the previously published study by Al-Sorori et al. (2016), endowing to the adapted FA both crossover and mutation mechanisms, and employing both 3-opt and 2-opt functions for enhancing the convergence and search performance of the method. An additional hybrid scheme was also proposed in Teng and Li (2018), in which FA is combined with GA. Authors of that work redefine the distance of FA by introducing a swap operator and swap sequence to avoid algorithm easily falling into local optima. A more elaborated study is presented in Li et al. (2015), which is focused on the solution of the multiple TSP, this being a generalization of the TSP in which more than one salesman is allowed to be used in the solution. In Chuah et al. (2017), a swap-based FA is developed, which bases its movement strategy on the widely employed swap function. Furthermore, authors of this paper integrate their FA with nearest-neighborhood initialization, a reset strategy, and a fixed-radius near-neighbor 2-opt operator. Two further interesting and valuable studies are presented in Zhou et al. (2015) and Jie et al. (2017), The former one has the particularity of adopting the dynamic mechanism based on a neighborhood search algorithm, while the second one is combined with a *k-opt* algorithm. Additional recent papers focusing on applications of FA can be found in Saraei and Mansouri (2019), Wang et al. (2018b), and Jati et al. (2013).

9.3.8 TSP and cuckoo search

Regarding CS, arguably the most valuable research published recently is the one published by Ouaarab et al. (2014), which is considered by the community as the first application of CS to the TSP. This paper has served as inspiration for subsequent research, such as that described in Ouaarab et al. (2015). In that paper, a random-key CS is proposed, which develops a simplified random-key encoding scheme to pass from a continuous space to a combinatorial space. Especially interesting is the work proposed in Tzy-Luen et al. (2016), in which a subpopulation-based parallel CS on Open Multiprocessing (OpenMP) is implemented for solving the TSP. Lin et al. developed the so-called genotype–phenotype CS in Lin et al. (2017), the essential contribution of which is the representation scheme used for building the solutions. Furthermore, the CS has been present in combination with other methods, such as the studies shown in Hasan (2018) and Kumar et al. (2015), in which the CS is implemented combined with ACO. Another example of this trend can be found in Min et al. (2017), who hybridized CS with the Metropolis acceptance criterion of an SA algorithm, in order to allow accepting inferior solutions with certain probability.

9.3.9 TSP and artificial bee colony

Since its inception in 2007 by Karaboga and Basturk, the ABC (Karaboga and Basturk, 2007) has also been adapted for solving combinatorial optimization problems such as the TSP. In recent years, specifically, it has been the focus point of some valuable studies within the TSP community. Very recent is the research proposed by one of the designers of the technique, Karaboga, along with Gorkemli in Karaboga and Gorkemli (2019), in which new improved versions of the discrete ABC were introduced for solving the symmetric TSP. Very recent is also the work that can be found in Choong et al. (2019), in which a hyperheuristic method called modified choice function is implemented for properly regulating the choice of the neighborhood search operators used by the onlooker and employed bees. An additional valuable study was presented in Zhong et al. (2017), introducing a hybrid ABC algorithm which adopts the threshold acceptance criterion method as accepting mechanism. Especially valuable is the work introduced by Venkatesh and Singh (2019), in which the challenging generalized covering TSP is tackled. To do that, authors developed an ABC with dynamic degrees of perturbations, where the degree to which a solution is modified for generating new bees is reduced along the execution. Singh also participated in the conduction of the work presented in Pandiri and Singh (2018), in which a hyperheuristic-based ABC was designed for facing a k-interconnected multidepot TSP. Further remarkable studies can be found in Khan and Maiti (2019), Hu et al. (2016), and Meng et al. (2016).

9.3.10 TSP and imperialist competitive algorithm

The imperialist competitive algorithm (ICA) is a multipopulation metaheuristic introduced in 2007 which finds its inspiration in the concept of imperialism, dividing the whole population in independent empires which fight with each other aiming at conquering the weakest colonies of the rest of the empire (Atashpaz-Gargari and Lucas, 2007). The solver has also been prolific in the TSP community, being used in many reputed studies published recently. We can find in Yousefikhoshbakht and Sedighpour (2013) the first adaptation of this sophisticated method, which has served as a main inspiration for many authors and works, such as Xu et al. (2014). In Ardalan et al. (2015), an improved version of ICA was presented for dealing with the generalized TSP. Authors of that work improved the basic version of ICA with some mechanisms such as a novel encoding scheme, an assimilation policy procedure, destruction/construction operators, and imperialist development plans. Furthermore, the Taguchi method is employed for properly configuring some of the most crucial parameters of the algorithm. Chen et al. (2017) proposed a hybrid method combining ICA with a policy learning function. The central idea behind this hybridization is to permit weak colonies to generate increasingly promising offspring by learning the policies of strong individuals. A brief adaptation of ICA can also be found in Osaba et al. (2018e), as part of a pool of metaheuristics for solving the TSP and the ATSP. Interested readers on this specific metaheuristic are referred to Firoozkooh (2011), Haleh and Esmaeili Aliabadi (2015), and Yousefikhoshbakht and Dolatnejad (2016).

9.3.11 TSP and other nature-inspired metaheuristics

Regarding the nature-inspired community, the proposal of PSO and ACO two decades ago decisively influenced the creation of a surfeit of methods, which clearly inherit their essential philosophy. For the design and proposal of these novel approaches, many different inspirational sources have been considered, such as (1) the behavioral patterns of animals such as buffaloes or whales, (2) social and political behaviors as hierarchical societies, and (3) physical processes such as optics systems, electromagnetic theory, or gravitational dynamics.

For this reason, in the current community a countless number of methods of this kind can be found. Along this section, some of the most successful metaheuristic solvers in the TSP community have been reviewed. In any case, we are perfectly aware that the whole community is composed of a plethora of additional methods, usually less often used than the ones outlined here. Furthermore, despite the comprehensive nature of this section, we are also conscious about the difficulty of congregating all the related works published. For this reason, we have only considered these ones that are strictly related with the TSP community, and which have been published in recognized scientific databases.

In any case, a reader may think of certain methods that deserve mention, or even a whole section. Seeking the completeness of this study, in the last part of this section we show a table summarizing additional methods that have been used in recent years for solving the TSP (Table 9.1). In this table we depict the name of the method, its main inspiration, and some related works.

9.4 Novelty search

The main objective of NS is to enhance the diversity capacity of a populations-based metaheuristic. To do that, this mechanism finds novel solutions in the behavioral space instead of the search space. Usually, candidates that comprise a population tend to congregate in the same region of the solution space. Conversely, this tendency does not happen in the behavioral space, which is structured employing the Euclidean distance. In this way, we can measure numerically the novelty of a candidate **x** using the following formula:

$$\rho(\mathbf{c}) = \frac{1}{k}\sum_{i=1}^{k} d(\mathbf{c}, \boldsymbol{\mu}_i), \tag{9.2}$$

where $d(\cdot,\cdot)$ represents the Euclidean distance. Additionally, k is the number of neighbor solutions chosen from the subset of neighbor candidates selected from the subset of neighbors $\mathcal{N} = \{\mu_1, \mu_2, \ldots, \mu_k\} \subseteq \mathcal{P}$ (i.e., the neighborhood size). This last parameter is problem-dependent and should be established empirically. Additionally, the selection of individuals is conducted using the distance metric, which also depends on the problem.

Table 9.1 Summary of additional nature-inspired methods and their application to the TSP.

Algorithms	Main inspiration	Refs.
Flower pollination algorithm (Yang, 2012)	Pollination process of flowers	Zhou et al. (2017); Strange (2017)
Harmony search (Geem et al., 2001)	Mimicking the improvisation of music players	Boryczka and Szwarc (2019b,a)
Fireworks algorithm (Tan and Zhu, 2010)	Fireworks explosions and location of sparks	Luo et al. (2018); Taidi et al. (2017)
African buffalo optimization (Odili et al., 2015)	The organizational ability of African buffaloes	Odili and Mohmad Kahar (2016); Odili et al. (2017)
Brain storm optimization (Shi, 2011)	Human brainstorming process	Xu et al. (2018); Hua et al. (2016)
Golden ball metaheuristic (Osaba et al., 2014b)	Teams and players organization in the soccer world	Osaba et al. (2014a); Sayoti and Riffi (2015)
Penguin search optimization (Gheraibia and Moussaoui, 2013)	Collaborative hunting strategy of penguins	Mzili et al. (2015, 2017)
Honey bee mating optimization (Haddad et al., 2006)	Honey bee mating process	Odili et al. (2016); Marinakis et al. (2011)
Whale optimization algorithm (Mirjalili and Lewis, 2016)	Social behavior of humpback whales	Gupta et al. (2018)
Water cycle algorithm (Eskandar et al., 2012)	Natural surface runoff of water	Osaba et al. (2018e)
Swallow swarm optimization (Neshat et al., 2013)	Reproduce the behavior of swallow swarms	Bouzidi and Riffi (2017)
Black hole algorithm (Hatamlou, 2013)	Black hole phenomenon in the open space	Hatamlou (2018)
Hydrological cycle algorithm (Wedyan et al., 2017)	Movement of water drops in natural cycle	Wedyan et al. (2018)
Dragonfly algorithm (Mirjalili, 2016)	Swarming behavior of dragonflies	Hammouri et al. (2018)
Pigeon-inspired optimization (Duan and Qiao, 2014)	Homing characteristics of pigeons	Zhong et al. (2019)

It is important to highlight that although NS has demonstrated a great efficiency in many works published up to now (Liapis et al., 2015; Gomes et al., 2015; Fister et al., 2019; López-López et al., 2018), the strategy for properly adapting this mechanism to a problem is still weakly defined, and it is subject to the problem at hand (Fister et al., 2018).

In the research that we are presenting in this chapter, NS has been applied in the same way for the three implemented bio-inspired metaheuristics. Something crucial when implementing NS is the modeling of a proper distance metric. In this study, the function selected in the Hamming distance $D_H(\cdot,\cdot)$, which is detailed in the following section. Additionally, a subset $\mathcal{B}$ is considered, in which all the discarded and replaced candidates are inserted every generation. This way, the size of $\mathcal{B}$ is the same as the main population of the solver.

Conceptually, the subset $\mathcal{B}$ is comprised of the solutions which are potentially *novel*, and prone to be reintroduced in the main population. Thus, when an evolved candidate $\mathbf{c}_i$ is better than the individual which it is going to replace, it is directly inserted into the principal population, while the replaced solution is introduced in $\mathcal{B}$. On the other hand, if the trial candidate is not better than its preceding version, the former is inserted into $\mathcal{B}$. Additionally, once the tth generation comes to its end, if r_{NS} (a value drawn from a normal probability distribution) is lower than the parameter $NS_P \in [0.0, 1.0]$, the NS mechanism is conducted. In this research, we have set $NS_P = 0.25$ after a comprehensive empirical analysis.

It is also noteworthy that there is not a specific scientific consensus about the proper number of solutions that should be reintroduced in the main population throughout NS, and how they should replace the existing individuals. In this regard, researchers advocate to adapt these criteria depending on the problem at hand. In this specific work, we have set the number of reinserted candidates to eight. These solutions replace the worst individuals in terms of fitness of the main population. Moreover, these candidates are selected from $\mathcal{B}$ based on their distance regarding the whole swarm. Thus, the eight solutions having a greater diversity with respect to the population are those chosen for reinsertion.

Lastly, the main contribution that we propose in our implemented NS procedure consist on a novel neighborhood changing procedure. Specifically, every time a candidate $\mathbf{c}$ is inserted in $\mathcal{B}$, its movement function $\Psi(\cdot,\cdot)$ is modified. Hence, when a candidate is reintroduced in the principal population, it can explore the solution space using different strategies. This simple mechanism enhances both the diversity of the swarm and the exploratory capacity of the algorithm.

9.5 Proposed bio-inspired methods

We propose in this work the combination of three different bio-inspired metaheuristic methods and the NS mechanism. Before specifying the details of each solver, we introduce here some crucial aspects for properly understanding the research conducted.

These aspects are related to solution representation and the metrics used for measuring the difference between different candidates.

When solving the TSP, the way in which the routes are encoded can follow diverse strategies. In this work, the frequently referenced path encoding has been used. Thus, each individual is represented as a permutation of numbers depicting the sequential order in which the nodes are visited. For instance, in a given 10-node dataset, a possible solution could be encoded as $\mathbf{x} = [8, 9, 1, 4, 3, 5, 2, 6, 7, 0]$, meaning that node 8 is visited first, followed by nodes 9, 1, and so forth. Each candidate adopts this approach. Additionally, the objective function employed is the total cost of a complete path given in Eq. (9.1a).

Probably, the most crucial issue when adapting PSO, FA, and BA to a discrete problem such as the TSP is to design the functions resembling how candidates move around the solution space, while guaranteeing their efficient contribution to the search problem under study. For conducting these movements, three well-known movement operators have been used depending on the distance between individuals:

- *Insertion:* This is one of the most frequently used functions for solving combinatorial optimization problems of different nature. Specifically, it selects and extracts one randomly chosen node from the route. Afterwards, this node is reinserted in the route in a randomly selected position.
- *Swapping function*: This well-known function is also widely employed in lots of research studies (Tarantilis, 2005). In this case, two nodes of a solution are selected randomly, and they swap their position.
- *2-opt:* This operator, first proposed in Lin (1965), has been extensively applied in different kinds of routing problems such as the TSP (Tarantilis and Kiranoudis, 2007; Bianchessi and Righini, 2007). The main design principle behind this operator is to randomly eliminate two arcs within the existing route, in order to create two new arcs, avoiding the generation of subtours.

At this point, it is interesting to clarify that *insertion* has been considered as the main operator for all metaheuristic methods; *swapping* and 2-opt, however, compose the pool of functions that NS considers for the reinsertion of candidates.

Finally, for assessing the distance between two different individuals (routes), the well-known Hamming distance $D_H(\cdot,\cdot)$ has been adopted. This function is calculated as the number of noncorresponding elements in the sequence of both individuals; e.g., if the following vectors represent two feasible routes:

$$\mathbf{x}^{p} = [8, 9, 1, 4, 3, 5, 2, 6, 7, 0],$$
$$\mathbf{x}^{p'} = [8, 7, 1, 4, 3, 5, 0, 6, 2, 9],$$

their Hamming distance $D_H(\mathbf{x}^p, \mathbf{x}^{p'})$ would be equal to 4. Once the distance between the two individuals has been computed, the movement is performed. We now introduce the metaheuristic algorithms under consideration.

9.5.1 Bat algorithm

BA was proposed by Yang (2010), and it is based on the echolocation behavior of microbats, which can find their prey and discriminate different kinds of insects even in complete darkness. As can be read in several surveys (Yang and He, 2013a; Chawla and Duhan, 2015), this method has been extensively adapted for dealing with very diverse optimization fields and problems. The fact that many studies can be found in the literature purely focused on BA confirms that it attracts a lot of interest from the community (Saad et al., 2019; Lu and Jiang, 2019; Osaba et al., 2019; Chen et al., 2018).

BA was first proposed for solving continuous optimization problems (Yang, 2010). Thus, a discrete adaptation must be conducted for properly accommodating its scheme to the combinatorial nature of the problem tackled in this study. In the literature, several adaptations of this kind can be found (Osaba et al., 2018b; Cai et al., 2019). First, each bat in the population represents a feasible solution of the TSP. Moreover, both loudness A_i and pulse emissions r_i concepts have been modeled analogously to the naïve BA. In order to simplify the approach, no frequency parameter has been considered. Besides that, velocity v_i has been used adopting the Hamming distance as its similarity function as $v_p^t = \text{rand}[1, D_H(\mathbf{c}_p, \mathbf{c}^{best})]$. In other words, the velocity v_i of the pth bat in the population at generation t is a random number, which follows a discrete uniform distribution between 1 and the Hamming distance between $\mathbf{c}_p$ and the best bat of the swarm $\mathbf{c}^{best}$.

With all this, $\mathbf{c}_p$ moves towards $\mathbf{c}^{best}$ at generation t as

$$\mathbf{c}_p(t+1) = \Psi\left(\mathbf{c}_p(t), \min\left\{V, v_p^t\right\}\right), \tag{9.3}$$

where $\Psi(\mathbf{c}, Z) \in \{$*insertion, swapping, 2-opt*$\}$ is the movement operator, parametrized by the Z times this function is applied to $\mathbf{c}$. After Z trials, the best considered movement is chosen as output.

9.5.2 Firefly algorithm

The first version of FA was developed by Yang (2008, 2009), and it was based on the idealized behavior of the flashing characteristics of fireflies. As we have pointed out for BA, FA has been the focus of many recent comprehensive surveys (Tilahun and Ngnotchouye, 2017; Fister et al., 2013, 2014; Yang and He, 2013b, 2018). Furthermore, it has been recently applied in many different problems and knowledge fields (Osaba et al., 2017; Danraka et al., 2019; Matthopoulos and Sofianopoulou, 2018; Osaba et al., 2018c).

Because the canonical FA was initially designed for dealing with continuous optimization problems, some modifications have also been done for its proper adaptation. Thus, as in the previously described BA, each firefly of the swarm represents a possible solution for the TSP. Moreover, light absorption has been considered, which is an essential concept for adjusting fireflies' attractiveness. For the movement of the fireflies around the solution space the same logic shown in Eq. (9.3) has been followed.

Finally, for measuring the similarity between two individuals, the $D_H(\cdot,\cdot)$ has also been used.

9.5.3 Particle swarm optimization

PSO is one of the most used swarm intelligence metaheuristics, and it has been adapted to both continuous (Precup and David, 2019) and discrete problems (Wu et al., 2019; Qiu and Xiang, 2019) in very recent years. Works such as Zhong et al. (2007) have inspired us for the discrete PSO developed in this research. Also in this adaptation each individual of the population (particles) represents a feasible solution for the faced problem, while the calculation of the velocity $v_i^{(t)}$ and movement functions have been considered as for the previously described solvers. Furthermore, the movement criterion represented in Eq. (9.3) has also been used for driving the movement of particles. Lastly, $D_H(\cdot,\cdot)$ has also been taken as distance function.

9.6 Experimentation and results

The performance of the three developed solvers has been gauged through 15 contrasted TSP datasets, all of them drawn from the famous TSPLIB repository (Reinelt, 1991). The size of the considered datasets is between 30 and 124 nodes. Taking as inspiration the good practices proposed in Osaba et al. (2018a), similar functions and parameters have been considered in all solvers, aiming at obtaining fair and rigorous insights. Additionally, 20 independent runs have been executed for each (*dataset*, *technique*) combination. Thus, we provide statistically reliable findings on the performance of each method. The population size has been established as 50 individuals for each method. In FA, the value of the light absorption coefficient is configured as $\gamma = 0.95$, whereas for BA $\alpha = \beta = 0.98$, $A_i^0 = 1.0$ (*loudness*), and $r_i^0 = 0.1$ (*rate*).

In Table 9.2 the outcomes obtained by each method are shown. For properly understanding the influence of NS in each metaheuristic scheme, we show not only the outcomes of each method using this mechanism (represented with the subscript NS), but also the results of the basic versions. Average (Avg) and standard deviation (SD) are provided for each (*problem*, *technique*) combination. Moreover, we also included in the table the mean generation number t_{conv} for which the best solution was met for every technique and problem instance. We represent this value in hundreds. Furthermore, we depict in bold the best outcome obtained for each metaheuristic, in order to facilitate the visual analysis of the influence of the NS. All the tests conducted in this work have been performed on an Intel Core i7-7600U, and Java has been used as the programming language.

Additionally, being aware that the comparison between the selected schemes PSO, FA, and BA is not the focus of this study, a statistical test has been carried out with the obtained results for the sake of completeness. To do that, and following the guidelines in Derrac et al. (2011), the Friedman nonparametric test for multiple comparison has been conducted, which allows to check if there are significant differences in

Table 9.2 Obtained optimization results using BA, FA, and PSO for the TSP in combination with the NS mechanism.

Instance		PSO			PSO_{NS}			FA			FA_{NS}			BA			BA_{NS}		
Name	**Optima**	**Avg**	**SD**	t_{conv}	**Avg**	**SD**	t_{conv}	**Avg**	**SD**	t_{conv}	**Avg**	**SD**	t_{conv}	**Avg**	**SD**	t_{conv}	**Avg**	**SD**	t_{conv}
Oliver30	420	**420.3**	0.47	0.21	420.4	0.49	0.10	421.0	0.80	0.04	**420.4**	0.58	0.03	421.2	1.69	0.18	**420.2**	0.43	0.11
Eilon50	425	435.4	4.49	0.80	**432.2**	3.89	0.40	**439.3**	2.91	0.14	439.4	2.33	0.11	436.0	5.33	0.71	**432.0**	3.88	0.38
Eil51	426	437.1	4.10	0.78	**434.5**	5.51	0.39	442.5	3.08	0.15	**440.0**	2.22	0.14	437.1	4.85	0.76	**433.5**	2.61	0.35
Berlin52	7542	**7667.3**	89.00	1.07	7699.8	148.69	0.49	7678.1	51.64	0.29	**7593.2**	25.98	0.16	7711.4	118.35	1.05	**7620.5**	100.98	0.49
St70	675	693.6	7.55	1.86	**689.8**	10.09	1.30	702.7	3.92	0.32	**697.3**	3.40	0.26	696.7	8.53	1.97	**688.4**	4.53	1.05
Eilon75	535	565.1	5.04	1.96	**549.4**	7.37	1.25	572.6	2.53	0.38	**569.2**	2.99	0.27	564.3	6.39	2.08	**555.2**	6.85	1.27
Eil76	538	565.2	7.34	2.47	**557.4**	7.87	1.24	572.4	3.27	0.42	**568.7**	2.79	0.23	565.4	7.68	2.28	**557.8**	7.87	1.05
KroA100	21282	22335.0	372.26	8.00	**21907.6**	495.90	5.03	22586.1	77.63	1.14	**22429.9**	77.35	0.47	22528.1	524.92	7.62	**21740.0**	246.23	5.27
KroB100	22140	23457.0	412.72	6.34	**22743.8**	259.96	5.70	23653.9	172.21	1.05	**23346.2**	147.47	0.58	23393.3	374.39	7.64	**22795.2**	343.61	5.04
KroC100	20749	22064.4	430.82	7.50	**21368.8**	347.78	5.74	22197.0	117.20	0.97	**21900.3**	117.55	0.46	22135.8	286.75	7.56	**21347.7**	430.22	5.10
KroD100	21294	22684.9	292.39	6.86	**21866.2**	338.32	5.45	22634.3	104.09	1.19	**22312.3**	85.21	0.48	22561.2	373.45	7.84	**22040.2**	514.55	4.81
KroE100	22068	23362.7	537.66	7.03	**22586.2**	286.88	5.35	23453.4	126.53	1.18	**23248.9**	112.23	0.45	23550.5	384.86	6.50	**22649.8**	393.95	5.76
Eil101	629	673.8	7.47	5.11	**635.0**	6.84	3.97	670.2	4.21	1.00	**662.3**	3.10	0.58	670.5	11.41	5.84	**654.6**	4.99	3.85
Pr107	44303	46592.5	563.92	9.25	**45746.3**	1056.43	5.86	46336.4	224.35	1.35	**45941.3**	90.77	0.42	46727.4	897.94	10.80	**45709.8**	981.18	6.47
Pr124	59030	64150.5	1635.70	14.85	**60387.7**	898.76	11.12	64505.9	332.04	1.43	**62552.8**	204.86	0.84	64436.7	1985.18	14.84	**60554.1**	1179.60	11.39
		FRIEDMAN'S NONPARAMETRIC TEST																	
Rank		4.0333			2.1			4.5333			4			4.7			**1.8333**		

the results obtained by all reported methods. Thus, in the last row of Table 9.2, we have displayed the mean ranking returned by this nonparametric test for each of the compared algorithms and scenarios (the lower the rank, the better the performance). Additionally, the Friedman statistic obtained is 35.914. The confidence interval has been set at 99%, with 9.236 being the critical point in a χ^2 distribution with five degrees of freedom. Since 35.914 > 5.991, it can be concluded that there are significant differences among the results.

Several conclusions can be drawn from the results obtained through this preliminary experimentation. First of all, it can be seen how the employment of the NS mechanism strongly improves the results in all the three used metaheuristic schemes. In the case of PSO, NS improves the average quality of the outcomes in 13 out of 15 instances. For FA, this improvement is observed for all 15 datasets. Finally, BA reaches better solutions in 14 of the 15 cases. These findings support the hypothesis that the NS procedure enhances the exploratory capacity of the three considered bio-inspired metaheuristics thanks to the diversification it injects in the population.

Equally interesting is the phenomenon that can be observed regarding the convergence behavior. In such a case, it can be seen how the introduction of the NS mechanism in the metaheuristic scheme supposes also an improvement on the convergence, reaching the final solution in a lower number of generations, and lower computation effort. Along with the improvement of the results, this feature supposes a huge advantage for the NS procedure.

As final reflection, and being something easily observable through the results obtained by the Friedman nonparametric test, we can highlight that the methods using NS are the ones that reached better outcomes. In this sense, BA_{NS} has emerged as the best alternative, followed by PSO_{NS} and FA_{NS}. Likewise, BA_{NS} is the solver that presents the best convergence behavior among the six implemented techniques. In any case, as mentioned before, the comparison between the different metaheuristics falls outside the scope of this experimentation.

9.7 Research opportunities and open challenges

In light of the literature overview made in Section 9.3 and the novel experimentation conducted exploring the synergies between bio-inspired computation and NS, it is unquestionable that the TSP is a topic that still attracts remarkable attention from the related community, being the scope of abundant research material. The current state of the computation and the multiple resources in the hands of practitioners open the opportunity of facing new challenges in the field. In this context, we foresee promising research directions along diverse axes, among which we pause at the following ones.

- As outlined in Section 9.3, an ample collection of classical and sophisticated solvers have been proposed in both past and recent literature for efficiently solving the TSP and its variants. One of the main challenges that the community should face urgently is to slow down the elaboration of additional novel methods. Despite the existence of a wide variety of well-reputed methods, part of the community continues scrutinizing the natural world seeking to formulate new metaheuristics mimicking some new biological phenomena. Some recent examples con be found in recent studies such as Kaveh and Zolghadr (2016), Arora and Singh (2019), and Wang et al. (2018a). These novel methods do not only offer a step forward for the community, but also augment the skepticism of critical researchers. These practitioners are continuously questioning the need of new methods, which apparently are very similar to previously published ones. In contrast to this trend, the whole community should pull in the same direction, trying to adapt the existing methods to more complex formulations of the TSP, and explore the different synergies that can arise between different approaches or mechanisms.

- Related to the previous challenge, currently, the TSP is still conceived by the community as a benchmarking or an academic problem with a very limited applicability to real-world problems. Trying to deal with this stigma, practitioners in the field should work on the formulation of richer and more complex formulations of the TSP, aiming to adapt the problem to real logistic and transportation problems. This research trend, which is currently receiving some attention from researchers, has led to the coining of the term *rich* or *multiattribute* TSP. As pointed out in Section 9.2, these problems are attracting the interest of the scientific community for their closer match to realistic situations. Despite this growing activity, the research behind these specific formulations is still not remarkable. This is, in part, because part of the community is working in the branch mentioned in the previous challenge. Thus, through this chapter, we call for a profound reflection around not only the formulation of new complex formulations of the TSP, but also the exploration of new ways for their solution, such as hybridized and memetic metaheuristics.
- Finally, we highly encourage involved researchers to consider the tackling of TSP variants of large size. Many of the studies that can be found in the current literature deal with controlled problem datasets of small or medium size (in terms of number of nodes). The experimental part of this study is also an example of this tendency. Notwithstanding, real-world problems are prone to have a higher magnitude, supposing a challenge for both researchers and their proposed solvers. In fact, large-scale variants not only hinder the efficiency of many of the often used methods, but they also suppose a compromise for the convergence of the solvers. In this context, the consideration of new optimization approaches, such as the ones referred to as *large-scale global optimization* techniques, can unchain unprecedented benefits for this field. Some methods that can be considered for being applied are SHADE-ILS (Molina et al., 2018) or multiple offspring sampling (LaTorre et al., 2012). Additional interesting research trends related to computational efficiency can be found in the area of cooperative co-evolutionary algorithms (Ma et al., 2018). Lastly, an additional promising alternative could be the design and implementation of self-adaptive solvers (Kramer, 2008).

9.8 Conclusions

This chapter has focused on the review of the well-known TSP. In the first part of this work, we have briefly introduced this famous problem, along with some of its most valuable variants. After that, we have made a systematic overview of the recent history of this problem, describing some of the most remarkable studies published in recent years. To do that, our attention has gravitated around both classical (SA, TS, GA, etc.) and sophisticated (BA, ICA, FA) metaheuristic solvers. After this literature review, we have presented an experimental study focused on the hybridization of the NS mechanism and three different bio-inspired computation schemes: PSO, FA, and BA. The performance of the implemented solvers has been tested over a benchmark comprised of 15 well-known datasets. The main conclusions drawn from this first study support the hypothesis that NS is a promising mechanism for being considered for the solution of the TSP, proving that this procedure helps metaheuristics to improve the quality of their reached results.

After these preliminary tests, we have concluded our research by sharing our envisioned future of the related community. To do that, we have pinpointed several inspiring opportunities and their related challenges, which should gather most of the research efforts made in the coming years. Among the future lines of research we foresee, we advocate the facing of bigger and more applicable datasets, using alternative methods not yet deeply explored, or synergistic hybridization of solvers proposed by the related experts along the years. Arguably, we foresee an exciting and still prolific future for the TSP community, adding new alluring nodes to visit in this endless path that TSP research is.

Acknowledgment

Eneko Osaba and Javier Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program.

References

Aarts, E.H., Korst, J.H., van Laarhoven, P.J., 1988. A quantitative analysis of the simulated annealing algorithm: a case study for the traveling salesman problem. Journal of Statistical Physics 50 (1–2), 187–206.

Akhand, M., Hossain, S., Akter, S., 2016. A comparative study of prominent particle swarm optimization based methods to solve traveling salesman problem. International Journal of Swarm Intelligence and Evolutionary Computation 5 (139), 2.

Al-Sorori, W., Mohsen, A., et al., 2016. An improved hybrid bat algorithm for traveling salesman problem. In: International Conference on Bio-Inspired Computing: Theories and Applications, pp. 504–511.

Ardalan, Z., Karimi, S., Poursabzi, O., Naderi, B., 2015. A novel imperialist competitive algorithm for generalized traveling salesman problems. Applied Soft Computing 26, 546–555.

Arigliano, A., Ghiani, G., Grieco, A., Guerriero, E., Plana, I., 2019. Time-dependent asymmetric traveling salesman problem with time windows: properties and an exact algorithm. Discrete Applied Mathematics 261, 28–39.

Ariyasingha, I., Fernando, T., 2015. Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem. Swarm and Evolutionary Computation 23, 11–26.

Arora, S., Singh, S., 2019. Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing 23 (3), 715–734.

Asadpour, A., Goemans, M.X., Madry, A., Gharan, S.O., Saberi, A., 2017. An o (log n/log log n)-approximation algorithm for the asymmetric traveling salesman problem. Operations Research 65 (4), 1043–1061.

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

Bell, J.E., McMullen, P.R., 2004. Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics 18 (1), 41–48.

Bellmore, M., Nemhauser, G.L., 1968. The traveling salesman problem: a survey. Operations Research 16 (3), 538–558.

Bianchessi, N., Righini, G., 2007. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research 34 (2), 578–594.

Bolaños, R., Echeverry, M., Escobar, J., 2015. A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the multiple traveling salesman problem. Decision Science Letters 4 (4), 559–568.

Boryczka, U., Szwarc, K., 2019a. An effective hybrid harmony search for the asymmetric travelling salesman problem. Engineering Optimization, 1–17.

Boryczka, U., Szwarc, K., 2019b. The harmony search algorithm with additional improvement of harmony memory for asymmetric traveling salesman problem. Expert Systems with Applications 122, 43–53.

Bouzidi, S., Riffi, M.E., 2017. Discrete swallow swarm optimization algorithm for travelling salesman problem. In: Proceedings of the 2017 International Conference on Smart Digital Environment, pp. 80–84.

Burke, E.K., Cowling, P.I., Keuthen, R., 2001. Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem. In: Applications of Evolutionary Computing. Springer, pp. 203–212.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A.A., 2015. Rich vehicle routing problem: survey. ACM Computing Surveys (CSUR) 47 (2), 32.

Cai, Y., Qi, Y., Cai, H., Huang, H., Chen, H., 2019. Chaotic discrete bat algorithm for capacitated vehicle routing problem. International Journal of Autonomous and Adaptive Communications Systems (IJAACS) 12 (2), 91–108.

Carrabs, F., Cordeau, J.F., Laporte, G., 2007. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading. INFORMS Journal on Computing 19 (4), 618–632.

Chang, J.C., 2016. Modified particle swarm optimization for solving traveling salesman problem based on a Hadoop MapReduce framework. In: 2016 International Conference on Applied System Innovation (ICASI), pp. 1–4.

Chawla, M., Duhan, M., 2015. Bat algorithm: a survey of the state-of-the-art. Applied Artificial Intelligence 29 (6), 617–634.

Chen, M.H., Chen, S.H., Chang, P.C., 2017. Imperial competitive algorithm with policy learning for the traveling salesman problem. Soft Computing 21 (7), 1863–1875.

Chen, S., Peng, G.H., He, X.S., Yang, X.S., 2018. Global convergence analysis of the bat algorithm using a Markovian framework and dynamical system theory. Expert Systems with Applications 114, 173–182.

Choong, S.S., Wong, L.P., Lim, C.P., 2019. An artificial bee colony algorithm with a modified choice function for the traveling salesman problem. Swarm and Evolutionary Computation 44, 622–635.

Chowdhury, S., Marufuzzaman, M., Tunc, H., Bian, L., Bullington, W., 2018. A modified ant colony optimization algorithm to solve a dynamic traveling salesman problem: a case study with drones for wildlife surveillance. Journal of Computational Design and Engineering.

Christofides, N., 1976. The vehicle routing problem. RAIRO. Recherche Opérationnelle 10 (V1), 55–70.

Chuah, H.S., Wong, L.P., Hassan, F.H., 2017. Swap-based discrete firefly algorithm for traveling salesman problem. In: International Workshop on Multi-Disciplinary Trends in Artificial Intelligence, pp. 409–425.

Clerc, M., 2004. Discrete particle swarm optimization, illustrated by the traveling salesman problem. In: New Optimization Techniques in Engineering. Springer, pp. 219–239.

Contreras-Bolton, C., Parada, V., 2015. Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. PLoS ONE 10 (9), e0137724.

Dahan, F., El Hindi, K., Mathkour, H., AlSalman, H., 2019. Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem. Sensors 19 (8), 1837.

Danraka, S.S., Yahaya, S.M., Usman, A.D., Umar, A., Abubakar, A.M., 2019. Discrete firefly algorithm based feature selection scheme for improved face recognition. Computing & Information Systems 23 (2).

David, R.C., Precup, R.E., Petriu, E.M., Rădac, M.B., Preitl, S., 2013. Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity. Information Sciences 247, 154–173.

De Jong, K., 1975. Analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis. University of Michigan, Michigan, USA.

Deng, Y., Liu, Y., Zhou, D., 2015. An improved genetic algorithm with initial population strategy for symmetric TSP. Mathematical Problems in Engineering 2015.

Derrac, J., García, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1 (1), 3–18.

Dong, X., Cai, Y., 2019. A novel genetic algorithm for large scale colored balanced traveling salesman problem. Future Generations Computer Systems 95, 727–742.

Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1 (1), 53–66.

Duan, H., Qiao, P., 2014. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. International Journal of Intelligent Computing and Cybernetics 7 (1), 24–37.

Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M., 2012. Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. Applied Soft Computing 110 (111), 151–166.

Eskandari, L., Jafarian, A., Rahimloo, P., Baleanu, D., 2019. A modified and enhanced ant colony optimization algorithm for traveling salesman problem. In: Mathematical Methods in Engineering. Springer, pp. 257–265.

Ezugwu, A.E.S., Adewumi, A.O., Frîncu, M.E., 2017. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. Expert Systems with Applications 77, 189–210.

Fachini, R.F., Armentano, V.A., 2018. Exact and heuristic dynamic programming algorithms for the traveling salesman problem with flexible time windows. Optimization Letters, 1–31.

Fiechter, C.N., 1994. A parallel tabu search algorithm for large traveling salesman problems. Discrete Applied Mathematics 51 (3), 243–267.

Firoozkooh, I., 2011. Using imperial competitive algorithm for solving traveling salesman problem and comparing the efficiency of the proposed algorithm with methods in use. Australian Journal of Basic and Applied Sciences 5, 540–543.

Fister, I., Fister Jr, I., Yang, X.S., Brest, J., 2013. A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation 13, 34–46.

Fister, I., Yang, X.S., Fister, D., 2014. Firefly algorithm: a brief review of the expanding literature. In: Cuckoo Search and Firefly Algorithm. Springer, pp. 347–360.

Fister, I., Iglesias, A., Galvez, A., Del Ser, J., Osaba, E., 2018. Using novelty search in differential evolution. In: International Conference on Practical Applications of Agents and Multi-Agent Systems, pp. 534–542.

Fister, I., Iglesias, A., Galvez, A., Del Ser, J., Osaba, E., Fister Jr, I., Perc, M., Slavinec, M., 2019. Novelty search for global optimization. Applied Mathematics and Computation 347, 865–881.

Furini, F., Persiani, C.A., Toth, P., 2016. The time dependent traveling salesman planning problem in controlled airspace. Transportation Research. Part B: Methodological 90, 38–55.

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

Gendreau, M., Laporte, G., Semet, F., 1998. A tabu search heuristic for the undirected selective travelling salesman problem. European Journal of Operational Research 106 (2), 539–545.

Gheraibia, Y., Moussaoui, A., 2013. Penguins search optimization algorithm (PeSOA). In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 222–231.

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

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional.

Gomes, J., Mariano, P., Christensen, A.L., 2015. Devising effective novelty search algorithms: a comprehensive empirical study. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 943–950.

Grefenstette, J., Gopal, R., Rosmaita, B., Van Gucht, D., 1985. Genetic algorithms for the traveling salesman problem. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 160–168.

Groba, C., Sartal, A., Vázquez, X.H., 2015. Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: an application to fish aggregating devices. Computers & Operations Research 56, 22–32.

Gülcü, Ş., Mahi, M., Baykan, Ö.K., Kodaz, H., 2018. A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem. Soft Computing 22 (5), 1669–1685.

Gupta, R., Shrivastava, N., Jain, M., Singh, V., Rani, A., 2018. Greedy WOA for travelling salesman problem. In: International Conference on Advances in Computing and Data Sciences, pp. 321–330.

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

Haleh, H., Esmaeili Aliabadi, D., 2015. Improvement of imperialist colony algorithm by employment of imperialist learning operator and implementing in travel salesman problem. Journal of Development & Evolution Management 1394 (22), 55–61.

Hammouri, A.I., Samra, E.T.A., Al-Betar, M.A., Khalil, R.M., Alasmer, Z., Kanan, M., 2018. A dragonfly algorithm for solving traveling salesman problem. In: 2018 8th IEEE International Conference on Control System, Computing and Engineering, pp. 136–141.

Hasan, L.S., 2018. Solving traveling salesman problem using cuckoo search and ant colony algorithms. Journal of Al-Qadisiyah for Computer Science and Mathematics 10 (2), 59.

Hatamlou, A., 2013. Black hole: a new heuristic optimization approach for data clustering. Information Sciences 222, 175–184.

Hatamlou, A., 2018. Solving travelling salesman problem using black hole algorithm. Soft Computing 22 (24), 8167–8175.

Helsgaun, K., 2015. Solving the equality generalized traveling salesman problem using the Lin–Kernighan–Helsgaun algorithm. Mathematical Programming Computation 7 (3), 269–287.

Hu, G., Chu, X., Niu, B., Li, L., Lin, D., Liu, Y., 2016. An augmented artificial bee colony with hybrid learning for traveling salesman problem. In: International Conference on Intelligent Computing, pp. 636–643.

Hua, Z., Chen, J., Xie, Y., 2016. Brain storm optimization with discrete particle swarm optimization for TSP. In: 2016 12th International Conference on Computational Intelligence and Security (CIS), pp. 190–193.

Hussain, A., Muhammad, Y.S., Nauman Sajid, M., Hussain, I., Mohamd Shoukry, A., Gani, S., 2017. Genetic algorithm for traveling salesman problem with modified cycle crossover operator. Computational Intelligence and Neuroscience 2017.

Hussain, A., Muhammad, Y.S., Sajid, M.N., 2019. A simulated study of genetic algorithm with a new crossover operator using traveling salesman problem. Journal of Mathematics (ISSN 1016-2526) 51 (5), 61–77.

Jati, G.K., Manurung, R., Suyanto, 2013. Discrete firefly algorithm for traveling salesman problem: a new movement scheme. Swarm Intelligence and Bio-Inspired Computation: Theory and Applications, 295–312.

Jiang, Z., 2016. Discrete bat algorithm for traveling salesman problem. In: 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), pp. 343–347.

Jie, L., Teng, L., Yin, S., 2017. An improved discrete firefly algorithm used for traveling salesman problem. In: International Conference on Swarm Intelligence, pp. 593–600.

Jun-man, K., Yi, Z., 2012. Application of an improved ant colony optimization on generalized traveling salesman problem. Energy Procedia 17, 319–325.

Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39 (3), 459–471.

Karaboga, D., Gorkemli, B., 2011. A combinatorial artificial bee colony algorithm for traveling salesman problem. In: International Symposium on Innovations in Intelligent Systems and Applications, pp. 50–53.

Karaboga, D., Gorkemli, B., 2019. Solving traveling salesman problem by using combinatorial artificial bee colony algorithms. International Journal on Artificial Intelligence Tools 28 (01), 1950004.

Kaveh, A., Zolghadr, A., 2016. A novel meta-heuristic algorithm: tug of war optimization. International Journal of Optimization in Civil Engineering 6 (4), 469–492.

Kennedy, J., Eberhart, R., et al., 1995. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948.

Khan, I., Maiti, M.K., 2019. A swap sequence based artificial bee colony algorithm for traveling salesman problem. Swarm and Evolutionary Computation 44, 428–438.

Khan, I., Pal, S., Maiti, M.K., 2018. A modified particle swarm optimization algorithm for solving traveling salesman problem with imprecise cost matrix. In: 2018 4th International Conference on Recent Advances in Information Technology (RAIT), pp. 1–8.

Kirkpatrick, S., Gellat, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M., Brunese, P.A., 2019. Multiple traveling salesman problem with drones: mathematical model and heuristic approach. Computers & Industrial Engineering 129, 14–30.

Knox, J., 1994. Tabu search performance on the symmetric traveling salesman problem. Computers & Operations Research 21 (8), 867–876.

Kramer, O., 2008. Self-Adaptive Heuristics for Evolutionary Computation, vol. 147. Springer.

Kumar, S., Kurmi, J., Tiwari, S.P., 2015. Hybrid ant colony optimization and cuckoo search algorithm for travelling salesman problem. International Journal of Scientific and Research Publications 5 (6), 1–5.

Kumbharana, S.N., Pandey, G.M., 2013. Solving travelling salesman problem using firefly algorithm. International Journal for Research in Science & Advanced Technologies 2 (2), 53–57.

Lahyani, R., Khemakhem, M., Semet, F., 2017. A unified matheuristic for solving multi-constrained traveling salesman problems with profits. EURO Journal on Computational Optimization 5 (3), 393–422.

Laporte, G., 1992a. The traveling salesman problem: an overview of exact and approximate algorithms. European Journal of Operational Research 59 (2), 231–247.

Laporte, G., 1992b. The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 59 (3), 345–358.

Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S., 1999. Genetic algorithms for the travelling salesman problem: a review of representations and operators. Artificial Intelligence Review 13 (2), 129–170.

LaTorre, A., Muelas, S., Peña, J.M., 2012. Multiple offspring sampling in large scale global optimization. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8.

Lawler, E.L., Lenstra, J.K., Kan, A.R., Shmoys, D.B., 1985. The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization, Vol. 3. Wiley, New York.

Lehman, J., Stanley, K.O., 2008. Exploiting open-endedness to solve problems through the search for novelty. In: ALIFE, pp. 329–336.

Li, M., Ma, J., Zhang, Y., Zhou, H., Liu, J., 2015. Firefly algorithm solving multiple traveling salesman problem. Journal of Computational and Theoretical Nanoscience 12 (7), 1277–1281.

Liao, E., Liu, C., 2018. A hierarchical algorithm based on density peaks clustering and ant colony optimization for traveling salesman problem. IEEE Access 6, 38921–38933.

Liapis, A., Yannakakis, G.N., Togelius, J., 2015. Constrained novelty search: a study on game content generation. Evolutionary Computation 23 (1), 101–129.

Lin, S., 1965. Computer solutions of the traveling salesman problem. Bell System Technical Journal 44 (10), 2245–2269.

Lin, Y., Bian, Z., Liu, X., 2016. Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem. Applied Soft Computing 49, 937–952.

Lin, M., Zhong, Y., Liu, B., Lin, X., 2017. Genotype-phenotype cuckoo search algorithm for traveling salesman problem. Computer Engineering and Applications 2017 (24), 28.

Liu, J., Li, W., 2018. Greedy permuting method for genetic algorithm on traveling salesman problem. In: 8th International Conference on Electronics Information and Emergency Communication, pp. 47–51.

Liu, C., Zhang, Y., 2018. Research on MTSP problem based on simulated annealing. In: Proceedings of the 2018 International Conference on Information Science and System, pp. 283–285.

Lo, K.M., Yi, W.Y., Wong, P.K., Leung, K.S., Leung, Y., Mak, S.T., 2018. A genetic algorithm with new local operators for multiple traveling salesman problems. International Journal of Computational Intelligence Systems 11 (1), 692–705.

López-López, V.R., Trujillo, L., Legrand, P., 2018. Novelty search for software improvement of a slam system. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1598–1605.

Lu, Y., Jiang, T., 2019. Bi-population based discrete bat algorithm for the low-carbon job shop scheduling problem. IEEE Access 7, 14513–14522.

Luo, H., Xu, W., Tan, Y., 2018. A discrete fireworks algorithm for solving large-scale travel salesman problem. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8.

Lust, T., Teghem, J., 2010. The multiobjective traveling salesman problem: a survey and a new approach. In: Advances in Multi-Objective Nature Inspired Computing. Springer, pp. 119–141.

Ma, X., Li, X., Zhang, Q., Tang, K., Liang, Z., Xie, W., Zhu, Z., 2018. A survey on cooperative co-evolutionary algorithms. IEEE Transactions on Evolutionary Computation 23 (3), 421–441.

Mahi, M., Baykan, Ö.K., Kodaz, H., 2015. A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. Applied Soft Computing 30, 484–490.

Maity, S., Roy, A., Maiti, M., 2019. A rough multi-objective genetic algorithm for uncertain constrained multi-objective solid travelling salesman problem. Granular Computing 4 (1), 125–142.

Makuchowski, M., 2018. Effective algorithm of simulated annealing for the symmetric traveling salesman problem. In: International Conference on Dependability and Complex Systems, pp. 348–359.

Malek, M., Guruswamy, M., Pandya, M., Owens, H., 1989. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. Annals of Operations Research 21 (1), 59–84.

Marinakis, Y., Marinaki, M., Dounias, G., 2011. Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. Information Sciences 181 (20), 4684–4698.

Marinakis, Y., Marinaki, M., Migdalas, A., 2015. Adaptive tunning of all parameters in a multi-swarm particle swarm optimization algorithm: an application to the probabilistic traveling salesman problem. In: Optimization, Control, and Applications in the Information Age. Springer, pp. 187–207.

Matai, R., Singh, S., Mittal, M.L., 2010. Traveling salesman problem: an overview of applications, formulations, and solution approaches. In: Traveling Salesman Problem, Theory and Applications. IntechOpen.

Matthopoulos, P.P., Sofianopoulou, S., 2018. A firefly algorithm for the heterogeneous fixed fleet VRP. International Journal of Industrial and Systems Engineering.

Mavrovouniotis, M., Müller, F.M., Yang, S., 2016. Ant colony optimization with local search for dynamic traveling salesman problems. IEEE Transactions on Cybernetics 47 (7), 1743–1756.

Meng, L., Yin, S., Hu, X., 2016. A new method used for traveling salesman problem based on discrete artificial bee colony algorithm. Telkomnika 14 (1), 342.
Min, L., Bixiong, L., Xiaoyu, L., 2017. Hybrid discrete cuckoo search algorithm with metropolis criterion for traveling salesman problem. Journal of Nanjing University (Natural Science) 5, 17.
Mirjalili, S., 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Computing & Applications 27 (4), 1053–1073.
Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Advances in Engineering Software 95, 51–67.
Mohsen, A.M., Al-Sorori, W., 2017. A new hybrid discrete firefly algorithm for solving the traveling salesman problem. In: Applied Computing and Information Technology. Springer, pp. 169–180.
Molina, D., LaTorre, A., Herrera, F., 2018. Shade with iterative local search for large-scale global optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8.
Mzili, I., Bouzidi, M., Riffi, M.E., 2015. A novel hybrid penguins search optimization algorithm to solve travelling salesman problem. In: 2015 Third World Conference on Complex Systems (WCCS), pp. 1–5.
Mzili, I., Riffi, M.E., Benzekri, F., 2017. Hybrid penguins search optimization algorithm and genetic algorithm solving traveling salesman problem. In: International Conference on Advanced Information Technology, Services and Systems, pp. 461–473.
Neshat, M., Sepidnam, G., Sargolzaei, M., 2013. Swallow swarm optimization algorithm: a new method to optimization. Neural Computing & Applications 23 (2), 429–454.
Odili, J.B., Mohmad Kahar, M.N., 2016. Solving the traveling salesman's problem using the African buffalo optimization. Computational Intelligence and Neuroscience 2016, 3.
Odili, J.B., Kahar, M.N.M., Anwar, S., 2015. African buffalo optimization: a swarm-intelligence technique. Procedia Computer Science 76, 443–448.
Odili, J.B., Kahar, M.N., Noraziah, A., 2016. Solving traveling salesman's problem using African buffalo optimization, honey bee mating optimization & Lin–Kerninghan algorithms. World Applied Sciences Journal 34 (7), 911–916.
Odili, J., Kahar, M.N.M., Anwar, S., Ali, M., 2017. Tutorials on African buffalo optimization for solving the travelling salesman problem. International Journal of Software Engineering and Computer Systems 3 (3), 120–128.
Osaba, E., Díaz, F., Carballedo, R., Onieva, E., Perallos, A., 2014a. Focusing on the golden ball metaheuristic: an extended study on a wider set of problems. The Scientific World Journal 2014.
Osaba, E., Diaz, F., Onieva, E., 2014b. Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. Applied Intelligence 41 (1), 145–166.
Osaba, E., Onieva, E., Diaz, F., Carballedo, R., Lopez, P., Perallos, A., 2015. An asymmetric multiple traveling salesman problem with backhauls to solve a dial-a-ride problem. In: 2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 151–156.
Osaba, E., Carballedo, R., López-García, P., Diaz, F., 2016a. Comparison between golden ball metaheuristic, evolutionary simulated annealing and tabu search for the traveling salesman problem. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, pp. 1469–1470.
Osaba, E., Yang, X.S., Diaz, F., Lopez-Garcia, P., Carballedo, R., 2016b. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Engineering Applications of Artificial Intelligence 48, 59–71.
Osaba, E., Yang, X.S., Diaz, F., Onieva, E., Masegosa, A.D., Perallos, A., 2017. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. Soft Computing 21 (18), 5295–5308.
Osaba, E., Carballedo, R., Diaz, F., Onieva, E., Masegosa, A., Perallos, A., 2018a. Good practice proposal for the implementation, presentation, and comparison of metaheuristics for solving routing problems. Neurocomputing 271, 2–8.
Osaba, E., Carballedo, R., Yang, X.S., Fister Jr, I., Lopez-Garcia, P., Del Ser, J., 2018b. On efficiently solving the vehicle routing problem with time windows using the bat algorithm with random reinsertion operators. In: Nature-Inspired Algorithms and Applied Optimization. Springer, pp. 69–89.

Osaba, E., Del Ser, J., Camacho, D., Galvez, A., Iglesias, A., Fister, I., 2018c. Community detection in weighted directed networks using nature-inspired heuristics. In: International Conference on Intelligent Data Engineering and Automated Learning, pp. 325–335.

Osaba, E., Del Ser, J., Iglesias, A., Bilbao, M.N., Fister, I., Galvez, A., 2018d. Solving the open-path asymmetric green traveling salesman problem in a realistic urban environment. In: International Symposium on Intelligent and Distributed Computing, pp. 181–191.

Osaba, E., Del Ser, J., Sadollah, A., Bilbao, M.N., Camacho, D., 2018e. A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. Applied Soft Computing 71, 277–290.

Osaba, E., Yang, X.S., Fister Jr, I., Del Ser, J., Lopez-Garcia, P., Vazquez-Pardavila, A.J., 2019. A discrete and improved bat algorithm for solving a medical goods distribution problem with pharmacological waste collection. Swarm and Evolutionary Computation 44, 273–286.

Ouaarab, A., Ahiod, B., Yang, X.S., 2014. Discrete cuckoo search algorithm for the travelling salesman problem. Neural Computing & Applications 24 (7–8), 1659–1669.

Ouaarab, A., Ahiod, B., Yang, X.S., 2015. Random-key cuckoo search for the travelling salesman problem. Soft Computing 19 (4), 1099–1106.

Pandiri, V., Singh, A., 2018. A hyper-heuristic based artificial bee colony algorithm for k-interconnected multi-depot multi-traveling salesman problem. Information Sciences 463, 261–281.

Pang, W., Wang, K.p., Zhou, C.g., Dong, L.j., 2004. Fuzzy discrete particle swarm optimization for solving traveling salesman problem. In: The Fourth International Conference on Computer and Information Technology. CIT'04, 2004, pp. 796–800.

Pang, S., Ma, T., Liu, T., 2015. An improved ant colony optimization with optimal search library for solving the traveling salesman problem. Journal of Computational and Theoretical Nanoscience 12 (7), 1440–1444.

Potvin, J.Y., 1993. State-of-the-art survey—the traveling salesman problem: a neural network perspective. ORSA Journal on Computing 5 (4), 328–348.

Pozna, C., Precup, R.E., Tar, J.K., Škrjanc, I., Preitl, S., 2010. New results in modelling derived from Bayesian filtering. Knowledge-Based Systems 23 (2), 182–194.

Precup, R.E., David, R.C., 2019. Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems. Butterworth-Heinemann.

Qiu, C., Xiang, F., 2019. Feature selection using a set based discrete particle swarm optimization and a novel feature subset evaluation criterion. Intelligent Data Analysis 23 (1), 5–21.

Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Information Sciences 179 (13), 2232–2248.

Reinelt, G., 1991. TSPLIB: a traveling salesman problem library. ORSA Journal on Computing 3 (4), 376–384.

Roberti, R., Wen, M., 2016. The electric traveling salesman problem with time windows. Transportation Research. Part E, Logistics and Transportation Review 89, 32–52.

Rostami, A.S., Mohanna, F., Keshavarz, H., Hosseinabadi, A.A.R., 2015. Solving multiple traveling salesman problem using the gravitational emulation local search algorithm. Applied Mathematics & Information Sciences 9 (2), 1–11.

Roy, A., Manna, A., Maity, S., 2019. A novel memetic genetic algorithm for solving traveling salesman problem based on multi-parent crossover technique. Decision Making: Applications in Management and Engineering.

Saad, A., Dong, Z., Buckham, B., Crawford, C., Younis, A., Karimi, M., 2019. A new Kriging–Bat algorithm for solving computationally expensive black-box global optimization problems. Engineering Optimization 51 (2), 265–285.

Sahana, S.K., et al., 2018. An improved modular hybrid ant colony approach for solving traveling salesman problem. GSTF Journal on Computing (JoC) 1 (2).

Saji, Y., Riffi, M.E., 2016. A novel discrete bat algorithm for solving the travelling salesman problem. Neural Computing & Applications 27 (7), 1853–1866.

Sakai, M., Hanada, Y., Orito, Y., 2018. Edge assembly crossover using multiple parents for traveling salesman problem. In: 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems, pp. 474–477.

Saraei, M., Mansouri, P., 2019. HMFA: a hybrid mutation-base firefly algorithm for travelling salesman problem. In: Fundamental Research in Electrical Engineering. Springer, pp. 413–427.

Sayoti, F., Riffi, M., 2015. Random-keys golden ball algorithm for solving traveling salesman problem. International Review on Modelling and Simulations (IREMOS) 8 (1), 84–89.

Shi, Y., 2011. Brain storm optimization algorithm. In: International Conference in Swarm Intelligence, pp. 303–309.

Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, Q., 2007. Particle swarm optimization-based algorithms for TSP and generalized TSP. Information Processing Letters 103 (5), 169–176.

Smith, S.L., Imeson, F., 2017. GLNS: an effective large neighborhood search heuristic for the generalized traveling salesman problem. Computers & Operations Research 87, 1–19.

Strange, R., 2017. Discrete Flower Pollination Algorithm for Solving the Symmetric Traveling Salesman Problem. Ph.D. thesis.

Svensson, O., 2018. Algorithms for the asymmetric traveling salesman problem. In: 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science.

Taidi, Z., Benameur, L., Chentoufi, J.A., 2017. A fireworks algorithm for solving travelling salesman problem. International Journal of Computational Systems Engineering 3 (3), 157–162.

Tan, Y., Zhu, Y., 2010. Fireworks algorithm for optimization. In: International conference in swarm intelligence, pp. 355–364.

Tang, K., Li, Z., Luo, L., Liu, B., 2015. Multi-strategy adaptive particle swarm optimization for numerical optimization. Engineering Applications of Artificial Intelligence 37, 9–19.

Tarantilis, C.D., 2005. Solving the vehicle routing problem with adaptive memory programming methodology. Computers & Operations Research 32 (9), 2309–2327.

Tarantilis, C., Kiranoudis, C., 2007. A flexible adaptive memory-based algorithm for real-life transportation operations: two case studies from dairy and construction sector. European Journal of Operational Research 179 (3), 806–822.

Teng, L., Li, H., 2018. Modified discrete firefly algorithm combining genetic algorithm for traveling salesman problem. Telkomnika 16 (1), 424–431.

Tilahun, S.L., Ngnotchouye, J.M.T., 2017. Firefly algorithm for discrete optimization problems: a survey. KSCE Journal of Civil Engineering 21 (2), 535–545.

Tzy-Luen, N., Keat, Y.T., Abdullah, R., 2016. Parallel cuckoo search algorithm on OpenMP for traveling salesman problem. In: 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), pp. 380–385.

Vaghela, K.N., Tanna, P.J., Lathigara, A.M., 2018. Job scheduling heuristics and simulation tools in cloud computing environment: a survey. International Journal of Advanced Networking and Applications 10 (2), 3782–3787.

Venkatesh, P., Singh, A., 2019. An artificial bee colony algorithm with variable degree of perturbation for the generalized covering traveling salesman problem. Applied Soft Computing.

Wang, Y., Xu, N., 2017. A hybrid particle swarm optimization method for traveling salesman problem. International Journal of Applied Metaheuristic Computing (IJAMC) 8 (3), 53–65.

Wang, K.P., Huang, L., Zhou, C.G., Pang, W., 2003. Particle swarm optimization for traveling salesman problem. In: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics. IEEE Cat. No. 03EX693, vol. 3, pp. 1583–1585.

Wang, J., Ersoy, O.K., He, M., Wang, F., 2016. Multi-offspring genetic algorithm and its application to the traveling salesman problem. Applied Soft Computing 43, 415–423.

Wang, G.G., Gao, X.Z., Zenger, K., Coelho, L.d.S., 2018a. A novel metaheuristic algorithm inspired by rhino herd behavior. In: Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, vol. 142, pp. 1026–1033.

Wang, Y., Wang, Q.P., Wang, X.F., 2018b. Solving traveling salesman problem based on improved firefly algorithm. Computer Systems & Applications 8, 37.

Wedyan, A., Whalley, J., Narayanan, A., 2017. Hydrological cycle algorithm for continuous optimization problems. Journal of Optimization 2017.

Wedyan, A., Whalley, J., Narayanan, A., 2018. Solving the traveling salesman problem using hydrological cycle algorithm. American Journal of Operations Research 8 (03), 133.

Wu, X., Gao, D., 2017. A study on greedy search to improve simulated annealing for large-scale traveling salesman problem. In: International Conference on Swarm Intelligence, pp. 250–257.
Wu, X., Shen, X., Zhang, L., 2019. Solving the planning and scheduling problem simultaneously in a hospital with a bi-layer discrete particle swarm optimization. Mathematical Biosciences and Engineering 16 (2), 831–861.
Xu, S., Wang, Y., Huang, A., 2014. Application of imperialist competitive algorithm on solving the traveling salesman problem. Algorithms 7 (2), 229–242.
Xu, D., Weise, T., Wu, Y., Lässig, J., Chiong, R., 2015. An investigation of hybrid tabu search for the traveling salesman problem. Bio-Inspired Computing-Theories and Applications, 523–537.
Xu, M., Li, S., Guo, J., 2017. Optimization of multiple traveling salesman problem based on simulated annealing genetic algorithm. In: MATEC Web of Conferences, vol. 100, 02025.
Xu, Y., Wu, Y., Fu, Y., Wang, X., Lu, A., 2018. Discrete brain storm optimization algorithm based on prior knowledge for traveling salesman problems. In: 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 2740–2745.
Yang, X.S., 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press, UK.
Yang, X.S., 2009. Firefly algorithms for multimodal optimization. In: Stochastic Algorithms: Foundations and Applications. Springer, pp. 169–178.
Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Springer, pp. 65–74.
Yang, X.S., 2012. Flower pollination algorithm for global optimization. In: International Conference on Unconventional Computing and Natural Computation, pp. 240–249.
Yang, X.S., He, X., 2013a. Bat algorithm: literature review and applications. International Journal of Bio-Inspired Computation 5 (3), 141–149.
Yang, X.S., He, X., 2013b. Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence (IJSI) 1, 36–50.
Yang, X.S., He, X.S., 2018. Why the firefly algorithm works? In: Nature-Inspired Algorithms and Applied Optimization. Springer, pp. 245–259.
Yousefikhoshbakht, M., Dolatnejad, A., 2016. An efficient combined meta-heuristic algorithm for solving the traveling salesman problem. BRAIN. Broad Research in Artificial Intelligence and Neuroscience 7 (3), 125–138.
Yousefikhoshbakht, M., Sedighpour, M., 2013. New imperialist competitive algorithm to solve the travelling salesman problem. International Journal of Computer Mathematics 90 (7), 1495–1505.
Yu, B., Yang, Z.Z., Yao, B., 2009. An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research 196 (1), 171–176.
Yu, X., Chen, W.N., Hu, X.m., Zhang, J., 2015. A set-based comprehensive learning particle swarm optimization with decomposition for multiobjective traveling salesman problem. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 89–96.
Zaidi, T., Gupta, P., 2018. Traveling salesman problem with ant colony optimization algorithm for cloud computing environment. International Journal of Grid and Distributed Computing 11 (8), 13–22.
Zhan, S.h., Lin, J., Zhang, Z.j., Zhong, Y.w., 2016. List-based simulated annealing algorithm for traveling salesman problem. Computational Intelligence and Neuroscience 2016, 8.
Zhang, Z., Gao, C., Lu, Y., Liu, Y., Liang, M., 2016. Multi-objective ant colony optimization based on the Physarum-inspired mathematical model for bi-objective traveling salesman problems. PLoS ONE 11 (1), e0146709.
Zhong, W.h., Zhang, J., Chen, W.n., 2007. A novel discrete particle swarm optimization to solve traveling salesman problem. In: IEEE Congress on Evolutionary Computation, pp. 3283–3287.
Zhong, Y., Lin, J., Wang, L., Zhang, H., 2017. Hybrid discrete artificial bee colony algorithm with threshold acceptance criterion for traveling salesman problem. Information Sciences 421, 70–84.
Zhong, Y., Lin, J., Wang, L., Zhang, H., 2018. Discrete comprehensive learning particle swarm optimization algorithm with metropolis acceptance criterion for traveling salesman problem. Swarm and Evolutionary Computation 42, 77–88.
Zhong, Y., Wang, L., Lin, M., Zhang, H., 2019. Discrete pigeon-inspired optimization algorithm with metropolis acceptance criterion for large-scale traveling salesman problem. Swarm and Evolutionary Computation 48, 134–144.

Zhou, L., Ding, L., Qiang, X., Luo, Y., 2015. An improved discrete firefly algorithm for the traveling salesman problem. Journal of Computational and Theoretical Nanoscience 12 (7), 1184–1189.

Zhou, Y., Wang, R., Zhao, C., Luo, Q., Metwally, M.A., 2017. Discrete greedy flower pollination algorithm for spherical traveling salesman problem. Neural Computing & Applications, 1–16.

Zhou, A.H., Zhu, L.P., Hu, B., Deng, S., Song, Y., Qiu, H., Pan, S., 2019. Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. Information 10 (1), 7.