Metaheuristic Algorithms to Solve the Activity Chain Optimization: A review

Bladimir Toaza* Domokos Esztergár-Kiss**

** Department of Transport Technology and Economics, Budapest University of Technology and Economics, Budapest, Hungary, (e-mail: ltoaza93@edu.bme.hu).*
*** Department of Transport Technology and Economics, Budapest University of Technology and Economics, Budapest, Hungary, (e-mail: esztergar@mail.bme.hu)*

Abstract: Metaheuristics embrace a powerful family of optimization methods. These algorithms are created with the intent of mimicking some types of natural phenomena (such as the principles of physics, the theory of evolution, the communal behavior of groups of animals, or human behavior and style) and employing them to tackle difficult problems. Since the first metaheuristic was proposed, significant progress has been made, and countless new algorithms are constantly being proposed on a daily basis. On the other hand, the Activity Chain Optimization Problem is a combinatorial problem based on the Traveling Salesman Problem, which aims to optimize the daily activity schedules of individuals. Due to the complexity of solving these complex problems, metaheuristics are required as primary methods. Thus, this paper investigates the contribution of metaheuristics to solving the Activity Chain Optimization problem. We mapped descriptive and assessment features for 63 metaheuristics based on a metaheuristic classification. The findings are examined to reveal the usage tendencies of the algorithms, identifying the most prevalent and those that have potential for future research. Additionally, we open a discussion regarding a number of unexplored research gaps and prospects in this appealing scientific field.

1. INTRODUCTION

The artificial intelligence revolution has occasioned the development of fresh methods capable of providing near-optimal solutions to difficult and complex real-world optimization problems that would not have been feasible using traditional methods. Consequently, there is a wide range of exact and approximate techniques for solving optimization problems (Ezugwu *et al.*, 2021). Only the last decade has seen an outbreak in the number of natural or man-made processes employed as metaphors for the creation of next-generation approximate algorithms. There is a record number of novel strategies that have successfully given the best solutions for some difficult benchmark problem instances that were earlier believed to be unsolvable. However, the discipline of approximate algorithms has yet to mature in comparison to mathematics, chemistry, or physics (Sörensen, Sevaux and Glover, 2018). Hussain *et al.* (2019) report that after 2005, the approximate algorithms field attracted a greater number of researchers, where metaheuristics algorithms have largely been used and assessed on numerical problems such as discrete and continuous, constrained and unconstrained, and single- and multi-objective optimization.

An illustration of an optimization problem is the Traveling Salesman Problem (TSP), considered a combinatorial problem. TSP is an NP-hard problem, which means it has many potential feasible solutions. The computational complexity of the TSP is $O(n!)$ when a naïve solution approach is used; it means that the running time using this approach increases with the factorial of n nodes. So even for 20 cities, finding the solution becomes unfeasible (Laporte, 1992). Due to its computational complexity, TSP is often utilized as a benchmark problem for analyzing the performance of several algorithms in discrete optimization. To date, TSP has been employed to validate many algorithms in different applications (Halim and Ismail, 2019), such as vehicle routing (Yousefikhoshbakht, 2021), planning of production (Onar *et al.*, 2016), computer networks (Luo *et al.*, 2021), design of hardware devices (Rico-Garcia *et al.*, 2020).

An application of TSP in the transportation sector and routing problems is a novel approach named Activity Chain Optimization (ACO), which aims to optimize the daily activity scheduling of individuals by considering their preferences about the locations of the activities and the times to perform them (Esztergár-Kiss, 2020). As TSP, ACO is an NP-hard problem, even with higher complexity, due to the consideration of the location's flexibility when an individual wants to perform an activity. Hence, most metaheuristic algorithms employed to solve combinatorial optimization problems, might be applied to solve ACO, thanks to their advantages. Nevertheless, the emergence of novel approaches day by day makes it hard to find and select the best algorithm to solve whichever optimization problem, including the TSP and ACO. In order to sort the appearance of the new metaheuristic algorithms, these have been named "classical" if they were developed before the year 2000, and "new generation" if after. Until 2019, the number of citations received for the new generation metaheuristics with respect to the introduction year was starting to approach the same ratio as the classic metaheuristics (Dokeroglu *et al.*, 2019).

Even though TSP has been widely evaluated with a considerable number of metaheuristic algorithms, ACO still needs to be explored to reach a fair trade-off between computation time and solution quality. A preliminary application of metaheuristics in ACO is performed by (Esztergár-Kiss, Rózsa and Tettamanti, 2018), where a metaheuristic named genetic algorithms (GA) is used for both single- and multi-objective optimization of the travel time. It is demonstrated here that when GA generates large populations, the computation time increases drastically. Later, Esztergár-Kiss and Remeli (2021) propose a time-based variant of ACO and solve the problem using a dynamic programming exact approach and a greedy heuristic as a baseline. Both solving algorithms are evaluated, showing that the exact approach generates better quality solutions compared to the heuristic, but the computation time increases exponentially (e.g., a problem of size 14 takes 1 hour and 9 minutes) when dynamic programming is used. This comparison strategy might be applied with more algorithms, but to do so, it is first necessary to have a broad knowledge of the metaheuristics available in the literature. Understanding the existing algorithms in the literature is essential because this would provide a different point of view to approach optimization problems and not be limited to the classical techniques. Likewise, obtaining richer knowledge of their main features, such as advantages and disadvantages, principles of working, and parameters, may help avoid unnecessary and expensive experimental simulations and assessments that may last days or weeks. Hence, the current study aims to propose a comprehensive compilation of metaheuristics available in the literature that may be considered potential algorithms to solve ACO.

The rest of this document is organized as follows: previous literature on metaheuristics is stated in Section 2. Section 3 details the data collection techniques of the metaheuristics. Section 4 provides information about the results obtained from the data collection. Section 5 discusses the results and outlines future research. In Section 6, concluding remarks are provided.

2. BACKGROUND ON METAHEURISTIC ALGORITHMS

This section seeks to provide comprehensive information about previous studies for classical and new generation metaheuristic algorithms. We have included surveys that have had a significant impact on the study of combinatorial problems, as well as studies comparing the performance of metaheuristics for solving the TSP or ACO.

2.1 Surveys about Metaheuristics

There have been numerous surveys on metaheuristics submitted over the years, but none have addressed every facet. For example, Ball and Magazine (1981) describe a primary classification and evaluation of heuristics based on their design with a limited number of examples. However, the taxonomies of metaheuristics depend on the researcher perspective. Fister Jr. *et al.* (2013) attempt to classify the metaheuristic algorithms based on the inspiration source into four major categories: swarm intelligence (SI) based, bio-inspired (but not SI-based), physics/chemistry-based, and others. These categories are briefly described, and a comprehensive list of 74 algorithms is presented. Abdel-Basset, Abdel-Fatah and Sangaiah (2018) intend to cover relevant points of metaheuristics to take a general view regarding taxonomies and variants. The authors state the classification of metaheuristics into algorithms that mimic natural phenomena and algorithms that do not base their search method on any natural phenomena. In total, seventeen algorithms are listed. The compilation is based on the number of citations received with respect to the introduction year. Also, Fausto *et al.* (2019) better define the four categories: evolution-based, swarm-based, physics-based, and human-based. A list comprised of 168 algorithms, along with their corresponding abbreviations, authors, and proposal year is documented, and only the most popular in the literature are explained. The compilation of the algorithms is based on the number of citations. Ezugwu *et al.* (2021) provide a comprehensive collection of over 200 metaheuristics from 1960 to 2019 with a new and well-defined taxonomic categorization of both classical and new generation algorithms. This classification emphasizes the design, inspiration source, variants, classification, impacts, and application areas. The bibliometric data are extracted from the Web of Science repository using the keyword metaheuristic to identify the number of citations.

2.2 Metaheuristics in the TSP and ACO

The literature reviews about metaheuristics for solving the TSP are not as abundant as the surveys describing general optimization problems. An early review is proposed by Blum and Roli (2003), who establish a conceptual comparison of six metaheuristics based upon on they implement the two key ideas for directing the search process: intensification and diversification. Bianchi *et al.* (2009) review the governing principles, inspiring notions, and intensification/ diversification techniques of four metaheuristics currently considered popular. Anbuudayasankar, Ganesh and Mohapatra (2014) define two categories for metaheuristics that are TSP-focused: memory-less and memory-based, according to the use of previously examined areas of the solution space. Elshaer and Awad (2020) present a taxonomic review of metaheuristics for solving the vehicle routing problem, which is a generalization of the TSP, grouping them into two categories: single solution-based and population-based. Eight algorithms are listed for the first category and sixteen for the second group. The collection strategy is to analyze articles published between 2009 and 2017 in journals with Impact Factors based on Thomson Reuters 2015, whose solution methodology includes a metaheuristic. The search method of the articles is based on keyword filtering within the title of the documents. Osaba, Yang and Del Ser (2020) perform a literature review about the advances achieved in eleven specific metaheuristic algorithms for solving the TSP.

The authors also present an experimental comparison of hybridized algorithms to find promising algorithms to solve the TSP. Zhang *et al.* (2022) approach the categorization in a differently, setting three historical stages on the construction of metaheuristic algorithms. The first stage, from 1960 to 1970, includes simple heuristic methods. The second stage, from 1970 to 1980, includes mathematical planning-based heuristic methods. The third stage, from 1990 to the present, includes the use of rigorous heuristic methods based on artificial intelligence.

Beyond conceptual comparison in surveys, the research trend is to implement experimental comparisons as a way to have a quantitative reference of the algorithms performance. Several authors compare the TSP using exact solution-based algorithms (Ait Bouziaren and Aghezzaf, 2019; Boccia *et al.*, 2021; Roberti and Ruthmair, 2021), classic and new generation metaheuristics (Aziz, 2015; Agrawal and Kaur, 2016; Zhou, Song and Pedrycz, 2018; Wu, 2020; Santos, Madureira and Varela, 2022), or both (Purkayastha *et al.*, 2020; Agung and Christine, 2021; Rbihou and Haddouch, 2021; Fakhravar, 2022). Likewise, day by day, it is possible to see the implementation of new metaheuristic algorithms or the enhancement of existing ones (Kóczy, Földesi and Tüű-Szabó, 2018; Luo *et al.*, 2021; Panwar and Deep, 2021). The performance of the algorithms is usually assessed using efficiency and effectiveness measures and statistical analysis (Halim, Ismail and Das, 2021), for which some guidelines are provided by Rardin and Uzsoy (2001).

To the best of our knowledge, neither conceptual nor experimental comparison has been conducted regarding the ACO problem. Therefore, the main contributions of this paper are: (1) an in-depth view of an attractive research area of metaheuristic design and classification that has been described in the literature to date. (2) a comprehensive collection of metaheuristics suitable for combinatorial problems focusing on the ACO, from 1960 to 2020, with the goal of offering a summarized description of the core concepts to design and apply these algorithms. (3) a method for conducting automated bibliography research in order to obtain data from indexing repositories.

3. METHODOLOGY

In this study, we propose a method to automate the search strategy of articles about solving the TSP with any metaheuristic algorithm. This strategy could also help researchers in other fields. First, we obtained from the literature the relevant definitions, notations, classifications, and categorizations of the metaheuristics to understand the entire context behind the combinatorial problems. Next, we listed the metaheuristics available in the literature up to 2020. Then, we determined features to identify the main characteristics of the listed algorithms. Finally, we mapped the features for each metaheuristic in the list, in order to present a conceptual comparison in a tabular form. The workflow of this study is summarized in Fig. 1, and the steps of the process are detailed as follows.

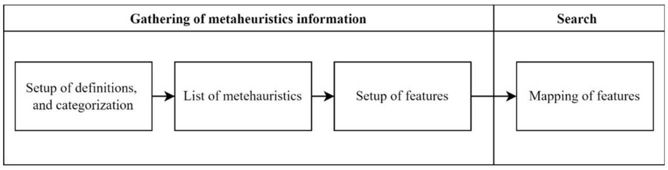

Fig. 1. Workflow of this study

3.1 *Set up of Categorization*

We extracted the definitions and categorizations from Fausto *et al.* (2019), who primarily categorized the algorithms into exact methods and approximate methods. Exact algorithms are assured to find the optimal solution to the problem, while approximate methods do not guarantee optimality but may obtain a near-optimal solution. The problem dimension is the criterion for choosing between these two methods, since the larger the problem, the more complex the solution space. Complex problems may turn exact methods into slower approaches to find a solution, but approximate methods can obtain a fair solution in a shorter processing time. Approximate methods include heuristic algorithms and metaheuristic algorithms. There exists a remarked difference between a heuristic and a metaheuristic algorithm; the first being a problem-dependent strategy, which means that it takes advantage of the particularities of the optimization problem to solve it, so a heuristic can be defined for a specific problem, but not necessarily this same heuristic would be applicable for other problems. The second is a problem-independent strategy and can be applied to a broad range of problems. However, in practice, these terms are used interchangeably, and metaheuristic has absorbed the term heuristic, so the term metaheuristic is now more commonly used to refer to both heuristic and metaheuristic. The metaheuristics are classified in this study according to their source of inspiration into four groups: evolution-based, swarm-based, physics-based, and human-based. Evolution-based algorithms are inspired by the laws of natural evolution. Swarm-based algorithms pretend to simulate the social and collective behavior manifested by groups of animals such as birds, insects, fishes, and others. Physics and Chemistry-based algorithms emulate the laws of physics or chemistry observed within our universe. Human-based algorithms find inspiration from several phenomena associated with human behavior.

3.2 *Metaheuristics Identification*

We obtained the list of algorithms from Ezugwu *et al.* (2021), who compiled 300 metaheuristics that have been developed between 1978 and 2020 and are available in the literature for any optimization problem. However, the metaheuristics have different performances on different optimization problems. For instance, a metaheuristic may perform excellently for continuous optimization problems, while it may perform badly when solving combinatorial optimization problems. Considering this and that the focus of this study is on the ACO, analyzing the complete list could be unnecessary.

Consequently, we suggested a reduced list of 110 metaheuristics suitable to solve combinatorial problems.

3.3 Set up of Features

The identification of the features of the metaheuristics could help to perform a conceptual or empirical comparison of the algorithms to identify the potential ones to solve the ACO. We grouped the features in two categories: description features and assessment features. The description features refer to descriptive characteristics of the metaheuristics such as the acronym of the algorithm, the author/developer, the class of the algorithm based on the categorization aforesaid, and the description of the inspiration source. On the other hand, the assessment features refer to the characteristics of the algorithms that might be used to evaluate them empirically, such as the novelty, the impact on the literature of TSP, the complexity, the parameter tuning, the implementation difficulty, the movement in the search space, and the number of candidate solutions. The features are briefly described in Table 1, and detailed information about how the features were obtained is explained next.

Table 1. Features identified

	Feature	Description
Descriptive features	Acronym	It refers to the acronym that simplifies the name of the metaheuristic algorithm for rapid mention in literature.
	Author	It refers to the name or names of the authors who developed the metaheuristic algorithm.
	Class	It refers to the class of the four-group classification, which the metaheuristic belongs to.
	Inspiration	It refers to the specific source for inspiration of the metaheuristic algorithm.
Assessment features	Novelty	It refers to the year in which the metaheuristic algorithm was developed.
	Impact	It refers to the amount of literature in which the metaheuristic was used for solving the TSP and its variants.
	Complexity	It refers to the number of initialized parameters required to develop the metaheuristic algorithm.
	Search space	It refers to the movement of the search agents of the metaheuristic algorithm in the search space, which may be local search, global search, or both.
	Number of candidate solutions	It refers to the population during the search process. It may be single- or population-based.
	Parameter tunning	It refers to the difficulty for tunning the initialized parameter of the metaheuristic algorithm.
	Implementation difficulty	It refers to the overall difficulty to implement the metaheuristic algorithm.

Acronym. The acronym is the abbreviation of the metaheuristic algorithm, usually formed by the initial letters of the words of the name. For instance, genetic algorithms are usually known in the literature as "GA," and iterated local search is known as "ILS." The acronyms are usually decided by the developers of the algorithms when they publish their metaheuristics in a scientific journal. This feature was obtained from the paper where the metaheuristic was published for the first time.

Author. The author or authors of the heuristic algorithm are usually the ones who publish it for the first time in scientific repositories. This feature was obtained from the paper where the metaheuristic was published for the first time.

Class. This feature considers the categorization to contain the metaheuristic in any of the four groups: evolution-based, swarm-based, physics- or chemistry-based, or human-based. For example, genetic algorithms, evolutionary programming, and cultural algorithms are evolution-based algorithms since they follow an evolution-based process. Information about the class of a specific metaheuristic was inferred from the paper where it was mentioned for the first time or another paper that applies it to an optimization problem.

Inspiration. This feature details the source of inspiration for the algorithm**.** Most metaheuristics are nature-inspired; they find inspiration in a natural phenomenon, which may be the social behavior of animals or the evolution process of bacteria, among others. For example, particle swarm optimization (PSO) is inspired by the social behavior of bird flocking or fish schooling, while social cognitive optimization is inspired by the human competition process. Information about this feature was usually described in the paper where the metaheuristic was first reported.

Novelty. The novelty of the metaheuristic is based on its development year, so the later the year of development, the greater the novelty of the metaheuristic. For example, GA were developed for the first time in 1975, while butterfly optimization algorithm (BOA) was developed in 2019. So, BOA has a higher grade of novelty compared to GA. Information about this feature was obtained from the first published paper about the correspondent metaheuristic.

Impact. The impact is reflected in the number of papers published using a specific metaheuristic in any optimization problem. However, as the focus of this work is the ACO, we considered only publications on solving the traveling salesman problem. The number of publications was obtained from a repository.

Complexity. This feature derives from the computational complexity, which is represented with big O notation and characterizes the growth of a function complexity as its input size grows. In this work, the complexity is based on the number of parameters required to run the algorithm; usually, these are called control parameters or initialization parameters. For example, genetic algorithms require generally four parameters: the population size, the maximum number of generations, the crossover operator, and the mutation operator; simulated annealing requires five parameters: the initial temperature, the annealing schedule, the candidate generator, the acceptance probability, and the number of iterations. Because most metaheuristics are run in a finite number of iterations and have predefined population sizes, we did not consider these two parameters when counting the number of parameters of each metaheuristic, so genetic algorithms only have two parameters, simulated annealing only had three, and so on. Information about this feature was obtained by reviewing the paper itself for each metaheuristic when it was reported for the first time.

Search space. This feature describes the movement of the metaheuristic search agent or agents in the search space, which can be local or global, or referred to as exploitation and exploration, respectively. Some metaheuristics have strength in exploring the search space, while others exploit the neighborhood of a suboptimal solution. For example, genetic algorithms are an example of global search algorithms since random searching agents are set in the search space and, through the evolution process, the agents start to converge on the optimal solution. The particle collision algorithm is an example of local-search focus; since it initially finds an apparent optimal solution, it exploits the neighborhood looking for a better solution. Other algorithms have both characteristics of local and global search, for example, particle swarm optimization. The information about this feature was obtained by reviewing the paper itself for each metaheuristic when it was reported for the first time.

Candidate solutions. This feature establishes the characteristics of the metaheuristics based on the population size of search agents. Most developed metaheuristics are population-based algorithms such as genetic algorithms, swarm particle optimization, and ant colony optimization, while only a very few of them are single-based solution algorithms such as simulated annealing and tabu search. Because of their strength in exploring the search space, population-based algorithms seem to have an advantage over single-based algorithms. The information about this feature was obtained by reviewing the paper itself for each metaheuristic when it was reported for the first time.

Tuning of parameters. Besides the number of control parameters, how to tune these parameters is important, too. The tuning difficulty of a parameter may influence the usability of a metaheuristic. We categorized the difficulty of tuning as easy, medium, or hard. Easy tuning means that all the parameters of certain metaheuristics are static and set manually in the range predefined by the author. For example, setting 0.9 for the crossover operator in GA. Medium tuning means that at least one of all the parameters uses probability distributions to set the parameters. For example, using Poison distribution when initializing the population in the search space. Hard tuning means that specific rules were created by the author to initialize the parameters. Additionally, the parameters may be derived directly from the population size; they are tuneless in this case. The information about this feature was obtained by reviewing the paper itself for each metaheuristic when it was reported for the first time.

Implementation difficulty. This feature is essentially based on the number of instructions or rules governing the metaheuristic. We categorized the overall implementation difficulty of metaheuristics as low, medium, or high. Low refers to a low difficulty to replicate and implement the metaheuristic for solving the ACO; medium means that the algorithm development process turns a bit complicated since it includes more steps and complicated equations. High difficulty means that the algorithm is quite difficult to implement due to a large number of equations and steps. For example, genetic algorithms have low difficulty because they consist mainly of four stages: evaluating the initial population, crossing over to generate offspring, mutating to generate children, evaluating the new population, and repeating the process, while almost all physics- and chemistry-based algorithms are difficult to implement due to their complex equations and number of steps to solve those equations.

3.4 Features mapping

The bibliometric data were extracted from the Scopus repository using the Scopus API. For that, a script was written in the Python language. The search keywords used were "traveling salesman problem" plus the corresponding algorithm in the list (e.g., "traveling salesman problem" AND "genetic algorithm"). The search range was from 2000 to 2022. The algorithm of the mapping strategy applied in this study is shown in Algorithm 1. For each metaheuristic, we extracted or inferred the descriptive and assessment features. In the case of the descriptive features, these were obtained from the first article published within the range of search. The extraction of the assessment features was extended to articles that include such metaheuristics in any experimental evaluation. Additionally, the impact feature was obtained from the number of publications mentioning or using the metaheuristic.

Algorithm 1. Scopus publications mapping

```
input: N metaheuristics
output: P publications number
initialize: Elsevier API client, publications = 0
for metaheuristic i in N do
    search ← "TITLE-ABS-KEY(traveling salesman
    problem AND '"+ i +"') AND PUBYEAR > 2000
    AND PUBYEAR < 2022"
    results ← execute search
    P(i) ← length of results
end for
return P
```

4. RESULTS

The mapping process revealed that there are between zero and one paper published until the end of 2020 using those metaheuristics developed in the last four years. It means that these metaheuristics have no impact on solving the TSP or its variants, and therefore the ACO. Consequently, these algorithms were filtered from this study, and a final list of 63 metaheuristic algorithms is presented. The list is descendent-sorted by development year and presents the data collected for all the descriptive and assessment features.

Some selected features of the mapped metaheuristics are shown in Fig. 2. A considerable portion of the listed metaheuristics are swarm-based; most of these have a medium difficulty to implement when solving the TSP or its variants. Only a few algorithms are highly difficult to employ, which correspond mostly to Physics-based and Human-based metaheuristics. The second bigger out the four groups is the Evolutionary-based group, whose difficulty is mostly low. Also, generally, the majority of algorithms have a global search strategy in the search space. From this, it is possible to observe that a big portion of global strategy belongs to population-based candidate solutions. Only a small percentage of candidate algorithms are single-based. Additionally, it is observable that the low difficulty of the algorithms is mostly connected to a global search space, while only a few are linked to a high difficulty.

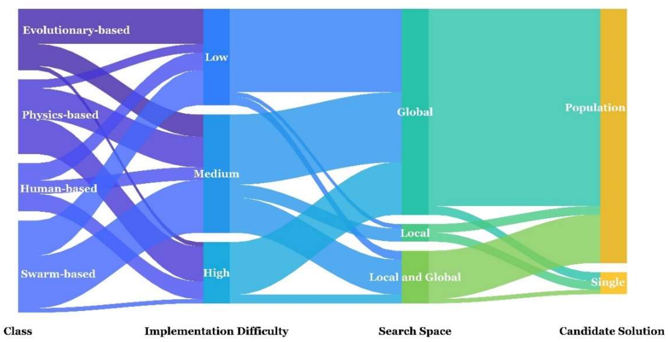

Fig. 2. Coupling of metaheuristics selected features

Fig. 3 shows the fifteen highest valued metaheuristics based on the impact of solving the TSP or its variants between 2000 and 2020. The most employed algorithm is genetic algorithms, which is an Evolutionary-based metaheuristic. The second highest valued metaheuristic is the ant colony optimization algorithm (ACOA), which is a swarm-based algorithm, and whose publications are 30% less compared to genetic algorithms. Both GA and ACOA are population-based algorithms. Simulated annealing is the third highest used, but to a modest extent compared to the first two algorithms. Although this algorithm is single-solution based, it seems to have the potential to be used in the ACO.

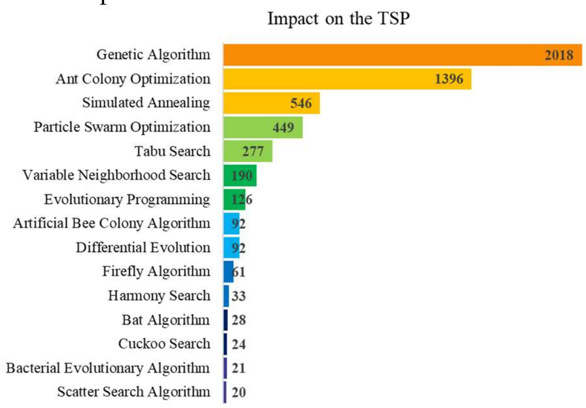

Fig. 3. Impact of the top fifteen metaheuristics on the TSP

5. DISCUSSION AND CONCLUSIONS

In this study, we presented a collection of potential metaheuristics to solve the ACO. From our review strategy, we selected 63 metaheuristics, and we mapped the literature for each metaheuristic to extract descriptive and assessment features. We found that the potential algorithms to solve the ACO are mostly swarm intelligence-based and evolutionary-based. This trend could be caused by the popularity of GA and PSO in the literature to solve any sort of optimization problem. Another factor could be the fact that swarm intelligence-based algorithms find their inspiration in the collective behavior of a species, and the search agents are a population instead of a single individual. This helps to explore the search space globally, making these metaheuristics more suitable for combinatorial problems.

In the literature, there exist a large number of algorithms to solve all sorts of optimization problems. However, not all are suitable for combinatorial problems; this is the reason why, from hundreds of algorithms, only 63 remained for analysis in this study. Additionally, the best approach to determine a potential algorithm is comparing it with others with an experimental method, which means running exhaustive computational simulations. Thus, for future research, an empirical qualitative comparison of all these algorithms based on the features described in this study could be performed, allowing the most highly ranked algorithms to be chosen for an experimental comparison solving the ACO.

REFERENCES

Abdel-Basset, M., Abdel-Fatah, L. and Sangaiah, A.K. (2018) 'Metaheuristic Algorithms: A Comprehensive Review', in A.K. Sangaiah, M. Sheng, and Z. Zhang (eds) *Computational Intelligence for Multimedia Big Data on* *the Cloud with Engineering Applications*. Academic Press, pp. 185–231. Available at: https://doi.org/10.1016/B978-0-12-813314-9.00010-4.

Agrawal, A.P. and Kaur, A. (2016) 'An empirical evaluation of three popular meta-heuristics for solving Travelling Salesman Problem', in *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*. Noida, India: IEEE, pp. 16–21. Available at: https://doi.org/10.1109/CONFLUENCE.2016.7508040.

Agung, C. and Christine, N. (2021) 'Performance analysis of optimization methods FOR SOLVING TRAVELING SALESMAN PROBLEM', *Innovative Technologies and Scientific Solutions for Industries*, 1(15), pp. 69–75. Available at: https://doi.org/10.30837/ITSSI.2021.15.069.

Ait Bouziaren, S. and Aghezzaf, B. (2019) 'An Improved Augmented -Constraint and Branch-and-Cut Method to Solve the TSP With Profits', *IEEE Transactions on Intelligent Transportation Systems*, 20(1), pp. 195–204. Available at: https://doi.org/10.1109/TITS.2018.2808179.

Anbuudayasankar, S.P., Ganesh, K. and Mohapatra, S. (2014) 'Survey of Methodologies for TSP and VRP', in *Models for Practical Routing Problems in Logistics: Design and Practices*. Cham: Springer International Publishing, pp. 11–42. Available at: https://doi.org/10.1007/978-3-319-05035-5_2.

Aziz, Z.A. (2015) 'Ant Colony Hyper-heuristics for Travelling Salesman Problem', *Procedia Computer Science*, 76, pp. 534–538. Available at: https://doi.org/10.1016/J.PROCS.2015.12.333.

Ball, M. and Magazine, M. (1981) 'The design and analysis of heuristics', *Networks*, 11(2), pp. 215–219. Available at: https://doi.org/10.1002/NET.3230110210.

Bianchi, L. *et al.* (2009) 'A survey on metaheuristics for stochastic combinatorial optimization', *Natural Computing*, 8(2), pp. 239–287. Available at: https://doi.org/10.1007/s11047-008-9098-4.

Blum, C. and Roli, A. (2003) 'Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison', *ACM Comput. Surv.*, 35(3), pp. 268–308. Available at: https://doi.org/10.1145/937503.937505.

Boccia, M. *et al.* (2021) 'An Exact Approach for a Variant of the FS-TSP', *Transportation Research Procedia*, 52, pp. 51–58. Available at: https://doi.org/https://doi.org/10.1016/j.trpro.2021.01.008.

Dokeroglu, T. *et al.* (2019) 'A survey on new generation metaheuristic algorithms', *Computers & Industrial Engineering*, 137, p. 106040. Available at: https://doi.org/https://doi.org/10.1016/j.cie.2019.106040.

Elshaer, R. and Awad, H. (2020) 'A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants', *Computers & Industrial Engineering*, 140, p. 106242. Available at: https://doi.org/https://doi.org/10.1016/j.cie.2019.106242.

Esztergár-Kiss, D. (2020) 'Trip Chaining Model with Classification and Optimization Parameters', *Sustainability*, 12(16). Available at: https://doi.org/10.3390/su12166422.

Esztergár-Kiss, D. and Remeli, V. (2021) 'Toward practical algorithms for activity chain optimization', *Transportation Letters*, 13(1), pp. 64–76. Available at: https://doi.org/10.1080/19427867.2019.1702250.

Esztergár-Kiss, D., Rózsa, Z. and Tettamanti, T. (2018) 'Extensions of the Activity Chain Optimization Method', *Journal of Urban Technology*, 25(2), pp. 125–142. Available at: https://doi.org/10.1080/10630732.2017.1407998.

Ezugwu, A.E. *et al.* (2021) 'Metaheuristics: a comprehensive overview and classification along with bibliometric analysis', *Artificial Intelligence Review*, 54(6), pp. 4237–4316. Available at: https://doi.org/10.1007/S10462-020-09952-0.

Fakhravar, H. (2022) 'Combining heuristics and Exact Algorithms: A Review'. arXiv. Available at: https://doi.org/10.48550/arxiv.2202.02799.

Fausto, F. *et al.* (2019) 'From ants to whales: metaheuristics for all tastes', *Artificial Intelligence Review 2019*, 53(1), pp. 753–810. Available at: https://doi.org/10.1007/S10462-018-09676-2.

Fister Jr., I. *et al.* (2013) 'A Brief Review of Nature-Inspired Algorithms for Optimization', *Elektrotehniski Vestnik/Electrotechnical Review*, 80(3), pp. 116–122. Available at: https://arxiv.org/abs/1307.4186v1 (Accessed: 7 October 2021).

Halim, A.H. and Ismail, I. (2019) 'Combinatorial Optimization: Comparison of Heuristic Algorithms in Travelling Salesman Problem', *Archives of Computational Methods in Engineering*, 26(2), pp. 367–380. Available at: https://doi.org/10.1007/s11831-017-9247-y.

Halim, A.H., Ismail, I. and Das, S. (2021) 'Performance assessment of the metaheuristic optimization algorithms: an exhaustive review', *Artificial Intelligence Review 2020 54:3*, 54(3), pp. 2323–2409. Available at: https://doi.org/10.1007/S10462-020-09906-6.

Hussain, K. *et al.* (2019) 'Metaheuristic research: a comprehensive survey', *Artificial Intelligence Review*, 52(4), pp. 2191–2233. Available at: https://doi.org/10.1007/s10462-017-9605-z.

Kóczy, L.T., Földesi, P. and Tüű-Szabó, B. (2018) 'Enhanced discrete bacterial memetic evolutionary algorithm - An efficacious metaheuristic for the traveling salesman optimization', *Information Sciences*, 460–461, pp. 389–400. Available at: https://doi.org/10.1016/J.INS.2017.09.069.

Laporte, G. (1992) 'The traveling salesman problem: An overview of exact and approximate algorithms', *European Journal of Operational Research*, 59(2), pp. 231–247. Available at: https://doi.org/https://doi.org/10.1016/0377-2217(92)90138-Y.

Luo, Y. *et al.* (2021) 'A New Memetic Algorithm to Solve the Stochastic TSP', in *2021 12th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*. Xi'an, China, China: IEEE, pp. 69–75. Available at: https://doi.org/10.1109/PAAP54281.2021.9720453.

Onar, S.Ç. *et al.* (2016) 'A Literature Survey on Metaheuristics in Production Systems', in E.-G. Talbi, F. Yalaoui, and L. Amodeo (eds) *Metaheuristics for Production Systems*.

Cham: Springer International Publishing, pp. 1–24. Available at: https://doi.org/10.1007/978-3-319-23350-5_1.

Osaba, E., Yang, X.-S. and Del Ser, J. (2020) 'Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics', in X.-S. Yang (ed.) *Nature-Inspired Computation and Swarm Intelligence*. London, United Kingdom: Elsevier, pp. 135–164. Available at: https://doi.org/10.1016/B978-0-12-819714-1.00020-8.

Panwar, K. and Deep, K. (2021) 'Discrete Grey Wolf Optimizer for symmetric travelling salesman problem', *Applied Soft Computing*, 105, p. 107298. Available at: https://doi.org/https://doi.org/10.1016/j.asoc.2021.107298.

Purkayastha, R. *et al.* (2020) 'Study and Analysis of Various Heuristic Algorithms for Solving Travelling Salesman Problem—A Survey', in J. Mandal and S. Mukhopadhyay (eds) *Proceedings of the Global AI Congress 2019. Advances in Intelligent Systems and Computing*. Singapore: Springer, pp. 61–70. Available at: https://doi.org/10.1007/978-981-15-2188-1_5.

Rardin, R.L. and Uzsoy, R. (2001) 'Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial', *Journal of Heuristics*, 7(3), pp. 261–304. Available at: https://doi.org/10.1023/A:1011319115230.

Rbihou, S. and Haddouch, K. (2021) 'Comparative study between a neural network, approach metaheuristic and exact method for solving Traveling salesman Problem', in *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*. IEEE, pp. 1–5. Available at: https://doi.org/10.1109/ICDS53782.2021.9626724.

Rico-Garcia, H. *et al.* (2020) 'Parallel implementation of metaheuristics for optimizing tool path computation on CNC machining', *Computers in Industry*, 123, p. 103322. Available at: https://doi.org/https://doi.org/10.1016/j.compind.2020.103322.

Roberti, R. and Ruthmair, M. (2021) 'Exact Methods for the Traveling Salesman Problem with Drone', *TRansportation Science*, 55(2), pp. 315–335. Available at: https://doi.org/10.1287/TRSC.2020.1017.

Santos, A.S., Madureira, A.M. and Varela, L.R. (2022) 'A Self-Parametrization Framework for Meta-Heuristics', *Mathematics*, 10(3), p. 475. Available at: https://doi.org/10.3390/MATH10030475.

Sörensen, K., Sevaux, M. and Glover, F. (2018) 'A History of Metaheuristics', in R. Martí, P.M. Pardalos, and M.G.C. Resende (eds) *Handbook of Heuristics*. Cham: Springer International Publishing, pp. 791–808. Available at: https://doi.org/10.1007/978-3-319-07124-4_4.

Wu, Z. (2020) 'A Comparative Study of solving Traveling Salesman Problem with Genetic Algorithm, Ant Colony Algorithm, and Particle Swarm Optimization', in *2020 2nd International Conference on Robotics Systems and Vehicle Technology*. New York, NY, USA, USA: Association for Computing Machinery, pp. 95–99. Available at: https://doi.org/10.1145/3450292.3450308.

Yousefikhoshbakht, M. (2021) 'Solving the Traveling Salesman Problem: A Modified Metaheuristic Algorithm', *Complexity*. Edited by D. Volchenkov, 2021, p. 6668345. Available at: https://doi.org/10.1155/2021/6668345.

Zhang, H. *et al.* (2022) 'Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms', *Archives of Computational Methods in Engineering*, 29(1), pp. 195–221. Available at: https://doi.org/10.1007/s11831-021-09574-x.

Zhou, H., Song, M. and Pedrycz, W. (2018) 'A comparative study of improved GA and PSO in solving multiple traveling salesmen problem', *Applied Soft Computing*, 64, pp. 564–580. Available at: https://doi.org/10.1016/j.asoc.2017.12.031.