



A matheuristic approach for the family traveling salesman problem

Abtin Nourmohammadzadeh¹ · Malek Sarhani² · Stefan Voß¹

Received: 12 April 2023 / Revised: 11 August 2023 / Accepted: 16 August 2023 /

Published online: 22 September 2023

© The Author(s) 2023

Abstract

In the family traveling salesman problem (FTSP), there is a set of cities which are divided into a number of clusters called families. The salesman has to find a shortest possible tour visiting a specific number of cities from each of the families without any restriction of visiting one family before starting the visit of another one. In this work, the general concept of the Partial Optimization Metaheuristic Under Special Intensification Conditions is linked with the exact optimization by a classical solver using a mathematical programming formulation for the FTSP to develop a matheuristic. Moreover, a genetic and a simulated annealing algorithm are used as metaheuristics embedded in the approach. The method is examined on a set of benchmark instances and its performance is favorably compared with a state-of-the-art approach from literature. Moreover, a careful analysis of the specific components of the approach is undertaken to provide insights into the impact of their interplay.

Keywords Matheuristic · Partial optimization metaheuristic under special intensification conditions · Genetic algorithm · Family traveling salesman problem

✉ Abtin Nourmohammadzadeh
abtin.nourmohammadzadeh@uni-hamburg.de
<https://www.bwl.uni-hamburg.de/en/iwi.html>

✉ Stefan Voß
stefan.voss@uni-hamburg.de
<https://www.bwl.uni-hamburg.de/en/iwi.html>

Malek Sarhani
m.sarhani@aui.ma

¹ Institute of Information Systems (IWI), University of Hamburg, Von-Melle-Park 5, 20146 Hamburg, Germany

² School of Business Administration, Al Akhawayn University in Ifrane, Avenue Hassan II, P.O. Box 104, 53000 Ifrane, Morocco

1 Introduction

The traveling salesman problem (TSP) is one of the most investigated problems in operations research. The classical TSP consists of N cities and a salesman tries to find the shortest possible tour, which includes all the nodes and each one exactly once. Matai et al. (2010) present many real-world applications for the TSP; vehicle routing and communication networks are only two examples of them. The TSP is known to be NP-hard based on the seminal exposition of Karp (1972) and, therefore, this also holds for its generalizations.

The TSP has many variants and generalizations each representing a range of real-world problems. In this paper, we focus on a challenging and emerging TSP variant, which is the family TSP (FTSP). In this variant, we have some predetermined clusters of cities called families. In the FTSP, it is sought to visit a predefined number of nodes in each family and there is no obligation to visit the cities of each cluster contiguously. A main application of the FTSP arises in the order picking problem in warehouses where the goods of the same product type exist in different warehouses or in different storage places within the same warehouse (Morán-Mirabal et al. 2014). In this case, it is assumed that each product is a family and the number of members required to be visited from the family is the demand of that product.

Considering the advantages as well as drawbacks of mathematical programming and metaheuristics, matheuristics are developed to cover the strengths, take advantage of both approaches and boost the optimization ability (Maniezzo et al. 2021). A special characteristic of matheuristics is that the exploitation is done by benefiting from the features of mathematical models; accordingly the name “model-based heuristics” is sometimes used for such algorithms (Maniezzo et al. 2010). In this research, we aim at developing a matheuristic for solving a specific generalization of the TSP. Our method uses a mathematical programming approach for local optimization processes in different parts of the solutions together with a genetic algorithm (GA) as its metaheuristic component. For details on the GA concept, interested readers can be referred to Haupt and Haupt (2003). For local exact optimization, the mathematical model of the problem is programmed in PYTHON and solved by the GUROBI solver (Gurobi Optimization, LLC 2021). For more details on the use and the implementation of metaheuristics for the TSP, we refer to Taillard (2023).

To be more specific, our matheuristic works by using a GA to provide some appropriate candidate solutions, and then, neighborhood searches in a selection of solutions based on simulated annealing (SA) (Kirkpatrick et al. 1983) are performed. Subsequently, the solutions are improved by a procedure devised according to the concept known as Partial OPTimization Metaheuristic Under Special Intensification Conditions (POPMUSIC) (Taillard and Voss 2002). This improvement is made by working each time on a part of the solution through applying the mathematical programming model and using a standard solver like the Gurobi optimizer (Gurobi Optimization, LLC 2021). Beyond finding an appropriate solution approach, our goal is also to investigate the specific components within the solution approach to be able to provide component-based insights into matheuristics and metaheuristics development as it is demanded by, e.g., de Armas et al. (2022), Swan et al. (2022).

The outline of the remainder of this paper is as follows. Related works are introduced in Sect. 2. Section 3 is dedicated to explaining the corresponding mathematical model. The solution methodology is presented in Sect. 4. Then, Sect. 5 discusses the experiments and their results. Finally, the conclusions of this research are drawn in Sect. 6.

2 Literature review

In this section, we review different methods adopted in the literature in order to propose a suitable approach for the FTSP. Firstly, we start with adopted solution approaches to deal with the FTSP. Then, we focus on matheuristics and particularly POPMUSIC. Finally, we add a few pointers regarding the parametrization of GAs.

2.1 The family traveling salesman problem

The FTSP is a variant of the TSP which was introduced in Morán-Mirabal et al. (2014). In that paper, the authors propose an integer programming formulation for the problem, and it is shown that the CPLEX solver can only solve small instances of the problem in reasonable time. The authors develop two randomized heuristics for obtaining solutions to this problem. Bernardino and Paias (2018) introduce several compact and non-compact models for the FTSP. One of the non-compact models is able to solve instances with 127 nodes in less than 70 s and one of the instances with 280 nodes in 3615 s. Moreover, Bernardino and Paias (2021) present two metaheuristics and a hybrid algorithm, which combines a branch-and-cut algorithm with a local search scheme. This method can already be classified as a matheuristic concept. The results show that their proposed methods improve the previously best-known upper bounds. Reyes Vega (2014) reviews the FTSP with capacitated agents, including the different methods adopted for it.

A special case of the FTSP is the equality generalized TSP (GTSP), where the number of cities which must be visited in each family is one. The equality GTSP is investigated, e.g., in Cacchiani et al. (2010). Bernardino and Paias (2022) also address the FTSP with incompatibility constraints, which is a special case of the problem where it is not allowed to visit cities from incompatible families on the same route.

In Table 1, we summarize the works done in the FTSP field. Among others, we can conclude that the GA is a propitious algorithm that shows success in solving the FTSP (e.g. when combined with a local search; furthermore, branching techniques are also successfully applied by Bernardino and Paias 2021). The combination of the two is then an approach worth studying which has not yet been investigated for the FTSP (Bernardino and Paias 2022; Bernardino et al. 2022).

2.2 Matheuristics

Matheuristics have been used for more than a decade as efficient alternatives for pure exact and metaheuristic approaches to solve complex optimization problems. In

Table 1 Studies on the FTSP

Author(s)	Focus	Algorithm(s)
Morán-Mirabal et al. (2014)	Introduce the FTSP; describe an integer programming formulation for the problem	GRASP; biased random-key genetic algorithm
Bernardino and Paias (2018)	Propose several formulations for the FTSP, divided into compact formulations and non-compact ones	Iterated local search (ILS) and exact approaches
Bernardino and Paias (2022)	Introduce a new variant of the FTSP, which takes into account incompatibilities between families; propose two mathematical models	Ant colony optimization; iterated local search; branch-and-cut
Bernardino and Paias (2021)	Propose two metaheuristics, a population-based and a local search method, and a hybrid algorithm. Develop an instance generator to create additional test instances with different visiting patterns	GA with local search; iterated local search hybrid; branch-and-cut algorithm
Bernardino et al. (2022)	Study a family TSP variant present multiple mixed integer linear programming formulations	Branch and cut-based algorithms

fact, they have found applications in many real-life problems, including the inventory routing problem (Vadseth et al. 2021; Su et al. 2020), various types of (vehicle) routing problems (Bosco et al. 2014; Leggieri and Haouari 2018), the multi-depot ring star problem (Hill and Voß 2016), the redundancy allocation problem (Caserta and Voß 2014), the airline crew rostering problem (Doi et al. 2018), the dynamic berth allocation problem (Nishi et al. 2017), and generalized versions of the knapsack problem (Galli et al. 2023), to name a few. For a comprehensive and detailed overview of the advances in matheuristics, the interested reader is referred to Maniezzo et al. (2010), Maniezzo et al. (2021).

In particular, the POPMUSIC algorithm from Taillard and Voss (2002) has shown an outstanding ability in combinatorial optimization problems, including the berth allocation problem (Lalla-Ruiz and Voß 2014; Lalla-Ruiz et al. 2015), the point feature label placement problem (Alvim and Taillard 2009), the capacitated vehicle routing problem (Queiroga et al. 2021) and the multi-depot cumulative capacitated vehicle routing problem (Lalla-Ruiz and Voß 2019). In particular for the TSP, as an extension to Taillard (2017), Taillard and Helsgaun (2019) aim to build a list of good candidate edges with a complexity lower than quadratic in the context of TSP instances given by a general function. Their numerical results show that solutions of excellent quality can be obtained with an empirical complexity lower than quadratic. Moreover, Taillard (2022) proposes an adaptation of the approach. The method is tested on instances with billions of cities. We can then conclude that POPMUSIC is an appropriate approach

to solve very large instances of the TSP. However, to our knowledge, the approach has not yet been applied to the FTSP despite its practical importance. Additionally, POPMUSIC has been successful in solving similar issues involving problem decomposition. In Table 2, we summarize some main related works which use the POPMUSIC approach. More specifically, in that table, we focus on proposing a new hybridization of a branching strategy or a population-based method to be applied to a TSP variant, and papers applying it to a new domain.

As mentioned above, POPMUSIC is all about decomposing a problem and combining a metaheuristic and a mathematical programming approach. Their combination is done in different ways. In particular, the combination of the GA and branching approaches demonstrates potential in tackling diverse combinatorial optimization problems (Nagar et al. 1996; Poojari and Beasley 2009), including the TSP (Choi et al. 2003). However, their combination is not done in a systematic way as in POPMUSIC, which we propose in the following sections.

2.3 Parametrization of genetic algorithms

Most metaheuristics are using one or more parameters. In many cases they can be crucial for the success of a method. One distinction refers to whether a method is parameterized in general or whether the parameters need to be tuned for every instance. The latter way for parametrization, known as per-instance or instance-aware parametrization, is often more effective (Hutter and Hamadi 2005). As an example, consider the case of a GA, where small problem instances might be sufficiently solved by using a small population while large instances need a larger population (see, e.g., Almeida et al. 2013). In this paper, we adopt the response surface methodology (RSM; see, e.g., Box and Draper 2007), which is a widely used per-instance parametrization approach (as indicated in, e.g., Hutter et al. 2014).

The RSM is a statistical and mathematical technique used to optimize and analyze complex systems, processes, or experiments. It is particularly helpful when the relationship between input variables and the response of interest (output) is not easily discernible or is too costly or time-consuming to evaluate through direct experimentation.

RSM involves fitting a mathematical model to experimental data, typically using regression analysis, to approximate the behavior of the system. This model, known as the response surface, allows researchers to explore and identify the optimal input settings that lead to the desired response. By systematically varying the input variables and observing the corresponding responses, RSM helps to efficiently locate the optimal conditions that yield the best outcomes.

The method is widely applied in various fields such as engineering, manufacturing, chemistry, and process optimization, where it aids in understanding the underlying relationships between variables, reducing experimental efforts, and maximizing performance or quality (e.g., de Oliveira et al. 2019).

Table 2 Studies using the POPMUSIC approach

Author(s)	Problem	Contribution or findings
Ostertag et al. (2009)	Vehicle routing problem with time windows	A POPMUSIC combining a population-based metaheuristic with a decomposition strategy
Lalla-Ruiz et al. (2015)	Berth allocation problem under time-dependent limitations	The comparison with the best algorithm in the related literature shows that the POPMUSIC improves its solution quality in almost all instance sets
Taillard and Helsgaun (2019)	TSP	Present how to build a list of good candidate edges with a complexity lower than quadratic
Lalla-Ruiz and Voß (2019)	Multi-depot cumulative capacitated vehicle routing problem	Adopt a POPMUSIC approach that uses reduced versions of the problem instance at hand as a sub-problem to solve the overall problem
Taillard (2022)	TSP	A linearithmic randomized method based on the POPMUSIC for generating good and quick solutions to the TSP
Queiroga et al. (2021)	Capacitated vehicle routing problem	A branch-cut-and-price algorithm is used as a heuristic to solve subproblems POPMUSIC is successfully applied
Yu et al. (2022)	A multi-objective problem in maritime shipping (joint berth allocation and quay crane assignment)	A combination of POPMUSIC with K-means clustering
Xu et al. (2022)	Precedence-Constrained Colored TSP	An augmented variable neighborhood search (VNS) called POPMUSIC-based VNS (adopting a mutation operator)

3 Mathematical models

The following notations and formulations express a general mathematical model for the FTSP, which is based on the model of Morán-Mirabal et al. (2014) but includes some slight modifications:

Input	Explanation
O	The fixed start and end city of the salesman
V	The set of vertices or cities except O
N	The set of all cities $N = V \cup O$
m	The number of families
nv_l	The number of members from family l which must be visited; $l \in \{1, 2, \dots, m\}$
KN	The total number of cities (members) which must be visited; $KN = \sum_{l=1}^m nv_l$
$V_1, V_2, \dots, V_m$	Disjoint clusters of V so that $V_1 \cup V_2 \cup \dots \cup V_m = V$, each regarded as a family
E	The set of roads between any two cities (edges)
c_{ij}	The cost of traversing edge (i, j) ($(i, j) \in E$)
Variable	
x_{ij}	Binary variable = 1 if edge (i, j) ($\forall i, j \in N$) is traversed by the salesman; otherwise = 0
u_i	Auxiliary variable which keeps the number of visited cities until city i in the tour

$$\min z = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (1)$$

$$\sum_{i \in V} x_{Oi} = 1 \quad (2)$$

$$\sum_{j \in V} x_{jO} = 1 \quad (3)$$

$$\sum_{i \in N} x_{ij} \leq 1 \quad \forall j \in V \quad (4)$$

$$\sum_{j \in N} x_{ij} \leq 1 \quad \forall i \in V \quad (5)$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij} = KN + 1 \quad (6)$$

$$\sum_{i \in N} \sum_{j \in V_l} x_{ij} = nv_l \quad l = 1, 2, \dots, m \quad (7)$$

$$\sum_{i \in V_l} \sum_{j \in N} x_{ij} = nv_l \quad l = 1, 2, \dots, m \quad (8)$$

$$\sum_{i \in N} x_{ij} - \sum_{i \in N} x_{ji} = 0 \quad \forall j \in V \quad (9)$$

$$u_i - u_j + (|N| - 1) \times x_{ij} \leq |N| - 2 \quad \forall i, j \in V; i \neq j \quad (10)$$

$$1 \leq u_i \leq |N| - 1 \quad \forall i \in V \quad (11)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (12)$$

The objective function (1) calculates the total cost of visiting the selected family members; it has to be minimized. Constraints (2)–(3) set the start and the end of the tour to be at city O . Constraints (4)–(5) ensure that each city is entered and left at most once. Equation (6) ensures that the required family members are visited. Con-

straints (7)–(8) imply entering and exiting the required number of members from each family. The equal in-degree and out-degree of each node is guaranteed by constraints (9), and constraints (10)–(11) are responsible for avoiding subtours. The method of preventing subtours in this model is different from the one used by the base model presented by Morán-Mirabal et al. (2014). This is based on the Miller–Tucker–Zemlin formulation (Miller et al. 1960) and imposes lower complexity compared to the one of Morán-Mirabal et al. (2014) because with that alternative formulation the number of constraints sharply increases by the number of cities. For more details about subtour elimination constraints for the FTSP see Bernardino and Paias (2018).

The presented mathematical model is based on an undirected graph, reflecting the bidirectional nature of roads between any two cities, allowing traversal in both directions. However, it is essential to consider that this model might be applied to an asymmetric instance where the costs of traveling an edge in both directions are not necessarily equal. In such cases, the edges of the graph would have distinct associated weights or costs depending on the direction of traversal. Hence, for asymmetric instances, the graph can be considered as a directed one and each direction of an edge can be regarded as a distinct arc. This adjustment accommodates scenarios in which the transportation costs, travel times, or any other relevant factors may differ depending on the direction in which the road is traveled, providing a more accurate representation of the real-world conditions. The flexibility of this mathematical model allows it to be applied to a wide range of transportation systems, both symmetric and asymmetric, optimizing routes and efficiently addressing diverse scenarios.

4 Methodology

We use the concept of the POPMUSIC algorithm proposed in Taillard and Voss (2002) to design an efficient matheuristic for the FTSP.

In our matheuristic, a simple GA is employed, which works iteratively on a population of candidate solutions to improve them by genetic operators such as selection, mutation and crossover. The initial population of this GA is generated randomly. The solutions created by the genetic operators are merged with the original population of the iteration, and then, a proportion of the new population is selected according to the objective value by the roulette wheel method (see Bickel and Thiele 1996) to be improved by SA. The SA optimization method explores neighboring solutions around an incumbent solution to find an improved solution. It evaluates a specified number of neighboring solutions based on their objective value, denoted as NBS , and selects the neighboring solution with a certain probability determined by the formula $e^{\frac{-(f(s)-f(s'))}{T}}$, where $f(s)$ and $f(s')$ represent the objective values of the current and neighboring solutions, respectively. The parameter T represents the current temperature of the system, which controls the likelihood of accepting solutions that may not be better than the current one. By iteratively adjusting the temperature and exploring the solution space, the SA method can escape local optima and ultimately converge towards an optimal solution. In other words, we jump to the neighboring solution if it has a better objective value; otherwise this action is done according to a probability

which is directly proportional to the quality of the new solution and inversely proportional to the temperature. This SA stops whenever it cannot improve the objective value after a maximum number of consecutive iterations called *MCUI SA* (an abbreviation for **M**aximum **C**onsecutive **U**nsuccessful **I**terations of **S**A). The SA solutions are similarly added to the previous population.

Subsequently, a number of the best solutions (*PMB*) regarding the objective value and a number of other solutions (*PMO*) are also improved according to the POPMUSIC framework in each GA iteration. The members of *PMO* are selected again probabilistically using the roulette wheel method.

In the POPMUSIC framework, the solutions are divided into K parts; then, each time, one of these parts is chosen at random and an area including it and the D nearest parts to it called altogether area A is processed by solving the mathematical model of the problem. All variables except for those existing in A are fixed at their values in a given incumbent solution. Consequently, the Gurobi solver is called to optimize the active (non-fixed) variables, which are those existing within A . This procedure continues until no further improvement is found in the solution. It means that if we have consecutively tested all parts of a solution and the exact solving attempts cannot make any improvement, we terminate the POPMUSIC procedure on that solution.

The collaboration of GA, SA and POPMUSIC continues until the number of consecutive unsuccessful GA iterations, i.e., iterations without any improvement in the objective value, exceeds a limit. The pseudocode of this matheuristic algorithm is presented in Algorithm 1.

The parameters of the used GA are the population size ($nPop$), the crossover rate (*Cross_rate*), the mutation rate (*Mutat_rate*), those of SA are the initial temperature (T_0) and cooling rate (*CT*), the number of neighborhood searches (*NBS*) and the maximum number of consecutive stagnant iterations *MCUI SA*. The rate of best solutions and other solutions for being improved by mathematical programming (*PMB_rate* and *PMO_rate*, respectively), the number of solution parts (K), and the number of nearest parts (D) are the parameters belonging to POPMUSIC. Finally, the number of unsuccessful iterations (*MCUI*, an abbreviation for **M**aximum **C**onsecutive **U**nsuccessful **I**terations) is an important parameter for the whole matheuristic.

The classical solution encoding scheme for the TSP is used here for the FTSP. A similar approach is also used in Bernardino and Paías (2021). This is a string containing a permutation of the numbers from one to $|V|$ as shown in the first row of Fig. 2. So each cell of this structure contains the label of one city. In order to translate the structure (chromosome) to an FTSP solution, it is assumed that the salesman visits the cities by their order in the string but whenever it comes to a label belonging to a cluster, whose required number of cities have already been visited, we ignore (eliminate) this city and jump to the next one until it comes to a city that belongs to a family which is still required to be visited.

As a simple example, if it is aimed at converting the first string shown in Fig. 2 to a feasible solution for an FTSP with three families as $\{1,2,5\}$, $\{3,4,9\}$ and $\{6,7,8,10\}$, and the assumption that the salesman must visit one city from the first, two cities from the second and two cities from the third family, we obtain the tour 3–5–9–10–7. The reason is that “3”, which is the first in the string, comes as the first city of the tour and is from the second family. Then, there is “5” in the string, which belongs to the first

Algorithm 1: The proposed matheuristic

Data: Problem inputs
Result: A feasible solution expected to be of good quality

- 1 Set the GA parameters: $nPop$, $Cross_rate$, $Mutat_rate$ and $MCUI$, the SA parameters: T_0 , CT , NBS and $MCUI SA$ as well as the POPMUSIC parameters: PMB_rate , PMO_rate , K and D .
- 2 Generate $nPop$ random candidate solutions.
- 3 Evaluate the solutions of the population based on their objective value.
- 4 $It = 1$.
- 5 Initialize a variable called BO , which keeps the best objective value found so far.
- 6 **while** $It \leq MCUI$ **do**
- 7 Choose the candidates for crossovers by roulette wheel selection.
- 8 Do the crossovers.
- 9 Do the mutations.
- 10 Evaluate the crossover and mutation results based on their objective value, and then merge them with the original population.
- 11 Choose a set PSA of candidates from the population by the roulette wheel selection.
- 12 Improve candidates of PSA by neighborhood searches based on SA.
- 13 Evaluate the SA results based on the objective value, and then, merge them with the previous population.
- 14 Sort the total population based on the objective values.
- 15 Choose PMB from the best solutions regarding the objective value and PMO based on the roulette wheel selection from other solutions.
- 16 **for** Each solution $spm \in (PMB \cup PMO)$ **do**
- 17 Decompose spm in K parts, i.e., the set of parts is $H = \{part_1, \dots, part_K\}$
- 18 Set $P = \emptyset$
- 19 **while** $P \neq \{part_1, \dots, part_K\}$ **do**
- 20 Select a seed part $s_{seed} \in H$ and $s_{seed} \neq P$ at random
- 21 Build a sub-problem R composed of s_{seed} and its nearest D parts (corresponding to area A)
- 22 Optimize R through solving the mathematical formulation by the Gurobi solver
- 23 **if** The objective value of spm has been improved (decreased) **then**
- 24 Update solution S
- 25 $P \leftarrow \emptyset$ (Empty P)
- 26 **else**
- 27 Include s_{seed} in P
- 28 **end**
- 29 **end**
- 30 **end**
- 31 Merge the POPMUSIC results with the population.
- 32 Sort the population based on the objective value and choose the first $nPop$ best of them as the new population.
- 33 Set BO = Objective value of the first solution of the population.
- 34 **if** BO is improved in comparison to its previous value **then**
- 35 $It = 0$
- 36 **else**
- 37 $It = It + 1$
- 38 **end**
- 39 **end**
- 40 Report BO .

family. Now the salesman is finished with the first family because only one city must be visited from it. Therefore, the next node in the string, “2”, is not visited or ignored regarding that it belongs to the first family. “9” is the next and is visited because two cities from the second family must be visited and we still need one. Finally, we need to visit two cities from the third family, “10” and then “7”, that come sooner in this order and are chosen as the last cities of the tour in this string. In this way, the salesman has visited the required number of cities from each family. This decoded solution is shown for some example positions of the ten cities in Fig. 1, where the three families are shown in different colors.

One-point crossovers are performed and the offspring chromosomes are repaired to make their elements unique. For this sake the repeated cities are replaced by the cities which do not exist in the string. The order of adding these required cities is according to the order of the repeated cities, which are removed. In other words, the city with the largest label from the former group comes to the gene containing the largest repeated city label. The second-largest label of the lacking cities is put in the place of the second-largest repeated city label and so on. Figure 2 depicts a simple example for doing the

Fig. 1 Decoding of the first chromosome shown in Fig. 2 to a feasible solution

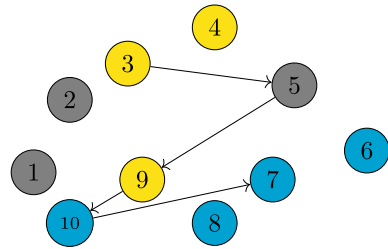
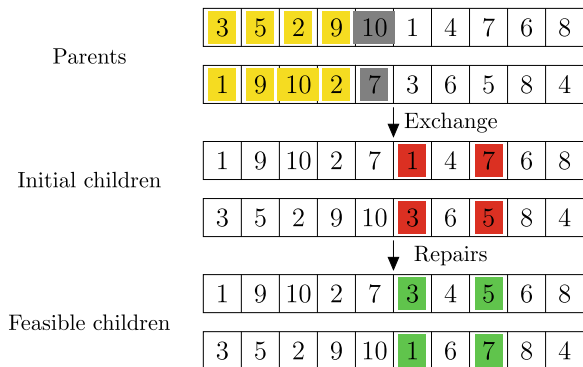


Fig. 2 Crossover and required repairs



crossover, and then, the required repairs to make the offspring results feasible. The grey genes are the crossover points, and after exchanging the corresponding parts, which include the genes from the beginning to the crossover point, “1” and “7” in the first child and “3” and “5” in the second child are repeated. Therefore, their repetitions are replaced by “3” and “5” and “1” and “7” in the first and second child, respectively. The choice to utilize the classical crossover method is grounded in its proven track record of yielding favorable results. On the other hand, some permutation-specific crossover methods may appear more straightforward since they obviate the need for child repair, but its outcomes are noticeably of inferior quality.

For mutation, completely new chromosomes are generated. The reason is that we aim at providing enough diversity in the population. Another method of mutation is to swap the elements in the chromosome. However, this operator is considered in our SA process.

In the SA process, the neighborhood of a chromosome is defined as comprising all chromosomes that differ in the position of either two nodes or edges. In other words, a neighboring chromosome is build by either swapping the contents of two genes or two pairs each including adjacent genes. The probability of doing each of these cases is equal. For the first case the two genes are selected at random, while in the second case two genes are randomly selected from 1 to $|N|-1$ ($|N|$ is the number of nodes or the chromosome length), and then, they and their immediate right genes make our two gene pairs. The first and second neighborhood are shown in Figs. 3 and 4, respectively. The grey genes are chosen randomly.

Each cell (gene) of this chromosome is considered as a solution part to apply POPMUSIC. So K is equal to the number of genes. It means that each time a gene is

Fig. 3 An example for a neighboring chromosome by swapping two nodes

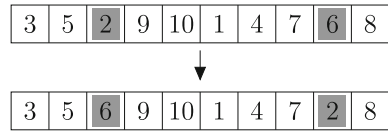


Fig. 4 An example for a neighboring chromosome by swapping two pairs of adjacent genes

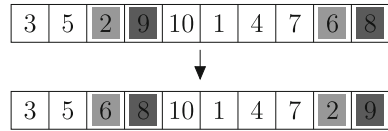
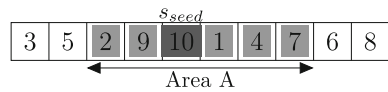


Fig. 5 Applying POPMUSIC to a chromosome



selected as s_{seed} together with the D nearest genes to it, which are located on both of its sides. They are selected one from the right, and then, one from the left, until D genes are picked, and in case that we reach the beginning or end of the chromosome in one side, the rest of the parts are all selected from the other side. These genes comprise the area A and the variables corresponding to pairs of consecutive genes within this area in the mathematical model are reset and optimized by the Gurobi solver, while other variables are fixed at the values that the decoding of the chromosome shows. Figure 5 depicts a simple example for the application of POPMUSIC on a chromosome. s_{seed} is shown in dark grey, D is equal to 5, so the last gene is chosen from the right. The D -nearest genes to s_{seed} are shown in grey. So $x_{2,9}$, $x_{9,10}$, $x_{10,1}$, $x_{1,4}$, $x_{4,7}$ are optimized by Gurobi through solving the corresponding sub-model.

5 Computational experiments

This section is divided into three parts of explaining the problem instances, the parameter setting, and the results. All our computational experiments are run on a computer with a Core(TM) i7 processor, 3.10 GHz CPU and 16 GB of RAM. The used programming language is PYTHON.

5.1 Instances

A set including numerous typical benchmark instances of the FTSP is used. This set consists of 37 specific instances, which are also considered in Morán-Mirabal et al. (2014), Bernardino and Paías (2018) and Bernardino and Paías (2021). These instances are available at <http://familytsp.rd.ciencias.ulisboa.pt/>. Note that except the last instance (rbg443_2), all others are symmetric in a metric space, where the triangle inequality is respected. The characteristics of the investigated instances including their number of cities, families, and known upper bounds (or optimal objective values) are listed in Table 3.

Table 3 The characteristics of the instances of Set 1, * optimal value

Instance	No. of cities	No. of families	No. of visits	Best-known upper bound
burma14_1	14	3	6	13.93*
burma14_2	14	3	10	25.66*
burma14_3	14	3	4	11.89*
bayg29_1	29	4	16	5345.86*
bayg29_2	29	4	17	5791.01*
bayg29_3	29	4	18	5544.33*
att48_1	48	5	34	23,686.00*
att48_2	48	5	25	20,609.10*
att48_3	48	5	15	9024.58*
bier127_1	127	10	62	33,709.70*
bier127_2	127	10	85	88,736.40*
bier127_3	127	10	60	47,726.30*
a280_1	280	20	179	1891.16
a280_2	280	20	155	1688.13
a280_3	280	20	141	1408.14*
gr666_1	666	30	357	1642.35
gr666_2	666	30	328	1293.96
gr666_3	666	30	328	1320.12
pr1002_1	1002	40	486	145,303.34
pr1002_2	1002	40	538	159,933.64
pr1002_3	1002	40	463	137,385.12
pr144_4	144	22	81	49,379
kroA150_3	150	16	117	20,880*
pr152_3	152	23	116	64,425
rat195_1	195	29	94	1285
rat195_3	195	29	145	1814
rat195_4	195	29	97	1320
kroA200_1	200	29	102	16,441
kroA200_3	200	29	159	24,471
kroB200_3	200	29	159	24,088
gr202_4	202	30	118	28,039
gr229_1	229	24	133	70,741
gil262_1	262	37	150	1526
gil262_3	262	37	212	1988
gil262_4	262	37	167	1671
pr264_2	264	37	93	28,748
rbg443_2	443	64	180	350

Table 4 The intervals for the parameters of our algorithm

Parameter	Interval
$nPop$	[50, 500]
$Cross_rate$	[0.1, 0.7]
$Mutat_rate$	[0.1, 0.7]
T_0	$[10^3, 10^6]$
CT	[1, 10]
NBS	[10, 200]
$MCUIA$	[10, 50]
PMB_rate	[0.1, 0.5]
PMO_rate	[0.1, 0.5]
D	[4, 40]
$MCUI$	[10, 100]
K	$ N $

5.2 Parameter setting

The parameter setting of our solution methodology is done separately for each instance, and the required time is considered in the reported execution time of the algorithm. The RSM (Box and Draper 2007) is applied for the parametrization, in which an empirical wide interval for each parameter is the input to find a good value within it. The parametrization with the RSM is implemented in R language. The considered intervals are shown in Table 4.

Regarding our previous experiments on this FTSP and other problems, tuning the parameters once and applying the same parameter values to all instances significantly deteriorates the algorithm's performance. Therefore, it is essential to investigate and fine-tune the parameters specifically for each instance.

Regarding the necessity of doing numerous experiments, we limit the execution time of each RSM experiment to $\frac{1}{5}$ of a normal run time for each instance to avoid long total execution times for the parametrization phase. For this sake, the instance is run once with average values of parameters and the corresponding execution time is measured and regarded as the normal run time.

K is considered to be equal to the number of cities or $|N|$. The reason for this choice is that there are $|V|$ genes or cells in the solution encoding structure and it is an immediate idea to consider each as a solution part.

A parameter setting step is executed here before solving each benchmark instance. The chosen values are shown in Table 5.

5.3 Results

As the comparative approach from literature, the hybrid method presented in Bernardino and Paia (2021) is chosen as it supersedes earlier approaches like the

Table 5 The parameter values set by the RSM and the time of the parameter setting phase for the benchmark instances

Parameter	<i>nPop</i>	<i>Cross_rate</i>	<i>Mutat_rate</i>	T_0	<i>CT</i>	<i>NBS</i>	<i>MCUI</i> <i>SA</i>	<i>PMB_rate</i>	<i>PMO_rate</i>	<i>D</i>	<i>MCUI</i>
<i>Instance</i>											
burma14_1	232	0.60	0.32	10^4	5	45	15	0.30	0.22	15	30
burma14_2	246	0.61	0.34	10^4	5	46	14	0.31	0.22	15	32
burma14_3	402	0.64	0.42	10^4	5	52	17	0.38	0.30	25	45
bayg29_1	412	0.65	0.42	10^4	5	50	20	0.39	0.30	25	46
bayg29_2	420	0.65	0.43	10^4	5	52	18	0.40	0.31	26	46
bayg29_3	472	0.67	0.45	10^5	5	55	20	0.44	0.34	27	47
att48_1	480	0.67	0.46	10^5	5	56	22	0.45	0.34	27	48
att48_2	492	0.68	0.47	10^5	5	54	21	0.46	0.35	27	50
att48_3	215	0.50	0.24	10^4	8	40	20	0.25	0.18	10	25
bier127_1	227	0.52	0.25	10^4	8	44	19	0.26	0.17	12	26
bier127_2	230	0.53	0.26	10^3	5	48	17	0.25	0.19	13	25
bier127_3	244	0.54	0.28	10^5	6	47	18	0.28	0.22	15	28
a280_1	263	0.55	0.29	10^4	8	48	19	0.29	0.23	15	30
a280_2	274	0.55	0.30	10^4	6	50	20	0.30	0.23	16	32
a280_3	270	0.54	0.32	10^4	8	49	19	0.30	0.24	15	28
gr666_1	294	0.55	0.34	10^5	5	51	22	0.33	0.24	15	30
gr666_2	302	0.56	0.35	10^4	7	52	20	0.32	0.24	16	33
gr666_3	315	0.58	0.33	10^4	6	55	22	0.31	0.25	15	34

Table 5 continued

Parameter	$nPop$	Cross_rate	Mutat_rate	T_0	CT	NBS	MCUI SA	PMB_rate	PMO_rate	D	MCUI
pr1002_1	276	0.52	0.35	10^4	5	47	23	0.33	0.26	16	35
pr1002_2	302	0.53	0.36	10^4	5	48	25	0.32	0.28	17	37
pr1002_3	321	0.54	0.37	10^5	4	50	26	0.34	0.28	17	39
pr144_4	215	0.50	0.24	10^4	8	42	17	0.25	0.18	10	25
kroA150_3	227	0.52	0.25	10^4	8	39	18	0.26	0.17	12	26
pr152_3	230	0.53	0.26	10^3	5	37	20	0.25	0.19	13	25
rat195_1	244	0.54	0.28	10^4	5	40	23	0.28	0.22	15	28
rat195_3	263	0.55	0.29	10^4	7	45	25	0.29	0.23	15	30
rat195_4	274	0.55	0.30	10^4	4	44	26	0.30	0.23	16	32
kroA200_1	270	0.54	0.32	10^4	7	46	22	0.30	0.24	15	28
kroA200_3	294	0.55	0.34	10^5	6	48	27	0.33	0.24	15	30
kroB200_3	302	0.56	0.35	10^4	5	50	27	0.32	0.24	16	33
gr202_4	315	0.58	0.33	10^4	8	52	26	0.31	0.25	15	34
gr229_1	276	0.52	0.35	10^4	4	47	22	0.33	0.26	16	35
gil262_1	302	0.53	0.36	10^4	4	50	25	0.32	0.28	17	37
gil262_3	321	0.54	0.37	10^5	5	51	24	0.34	0.28	17	39
gil262_4	334	0.56	0.38	10^5	4	53	26	0.35	0.28	18	42
pr264_2	322	0.52	0.36	10^4	6	50	21	0.35	0.26	17	40
rbg443_2	368	0.59	0.42	10^4	5	55	27	0.37	0.30	24	45

one of Bernardino and Paias (2018).¹ That is, the hybrid method has been shown to be the best solution methodology so far. Each instance is solved five times, and the best value (Best), the average of all runs (Average), the average execution time in seconds (Time (s)) are reported in Table 6 for our matheuristic and the hybrid method. In addition, for our matheuristic, the improvement percentages of best, and then, the average value compared to the hybrid method, and finally, the improvement of the best value over the previous best-known value (separated by a comma) are shown in the column “Improvement (%)”. Bold values indicate instance-based best values within the comparison.

It is observed, that our matheuristic is able to provide solutions with better objective values concerning best and average. In other words, our matheuristic has improved the previously found best solutions in many cases. Moreover, it has quite short execution times, although they are a little longer than those of the hybrid method. The superiority of our matheuristic becomes more evident as we face larger problem instances. The two last rows are dedicated to the average values and the p -values resulting from the execution of the Wilcoxon signed-rank test (see, e.g., Conover 1999) to investigate if there are statistically significant differences between the averages of the results of the two approaches.

The p -value represents the probability of obtaining results as extreme as the observed data, assuming that there is no real difference between the two groups (null hypothesis). A low p -value indicates that the observed difference is unlikely to be due to chance alone, and this leads to rejecting the null hypothesis in favor of the alternative hypothesis, which suggests that there is a significant difference between the two groups.

Typically, a significance level (often denoted as α) is predetermined, commonly set to 0.05. If the calculated p -value is less than the significance level, the results are considered statistically significant, and researchers can conclude that there is evidence of a meaningful difference between the groups. On the other hand, if the p -value is greater than the significance level, there is insufficient evidence to reject the null hypothesis, and the observed difference is deemed not statistically significant. In our case, the low p -values, which lead to the rejection of the hypotheses of the equality at the significance level of 0.05, show that there are statistically significant differences in terms of the solution quality and time.

Overall, it can be deduced that our matheuristic has the capability of providing better quality solutions for the FTSP when compared to the outcomes of the state-of-the-art algorithm. In addition, it is also quite fast.

5.4 Proportion of execution time related to components

In this section, we show how much of the total execution time of the algorithm is dedicated to each part of it. The proportions are shown in Table 7 for all instances separately and the averages are given in the last row. The considered parts are the

¹ It should be noted that the authors used a slightly different machine from ours, i.e., a computer with an Intel Core i7-4790 3.60 GHz processor and 8 GB of RAM.

Table 6 Comparison between our matheuristic and the best state-of-the-art method

Instance	Hybrid (Bernardino and Paais 2021)		Our matheuristic		Improvements (%)	Time (s)
	Best	Average	Best	Average		
burma14_1	13.93	13.93	13.93	13.93	0, 0, 0	1
burma14_2	25.66	25.66	25.66	25.66	0, 0, 0	1
burma14_3	11.89	11.89	11.89	11.89	0, 0, 0	1
bayg29_1	5345.86	5345.86	5345.86	5345.86	0, 0, 0	2
bayg29_2	5791.01	5791.58	5791.01	5791.01	0, 0.01, 0	2
bayg29_3	5544.33	5544.33	5544.33	5544.33	0, 0, 0	1
att48_1	23,686.02	23,686.02	23,686.02	23,686.02	0, 0, 0.00	2
att48_2	20,609.08	20751.30	20,609.08	20,609.10	0, 0.69, 0	3
att48_3	9024.58	9024.58	9024.58	9024.58	0, 0, 0	61
bier127_1	33,709.70	35833.41	33,709.70	34,216.61	0, 4.51, 0	5
bier127_2	88,736.40	94397.78	88,736.40	89,503.37	0, 5.2, 0	5
bier127_3	47,726.30	48933.78	47,726.30	47,852.65	0, 2.21, 0	5
a280_1	1739.30	1756.32	1734.91	1749.17	0.25, 0.41, 8.26	68
a280_2	1556.08	1562.15	1548.32	1556.38	0.50, 0.37, 8.28	63
a280_3	1408.14	1462.71	1408.14	1421.05	0, 2.85, 0	20
gr666_1	1531.93	1553.61	1523.72	1542.62	0.54, 0.71, 7.22	915
gr666_2	1237.03	1241.68	1228.84	1237.66	0.66, 0.32, 5.03	933
gr666_3	1225.11	1232.10	1218.51	1229.06	0.54, 0.25, 7.70	990
pr1002_1	134625.52	135134.09	133,508.30	134,641.72	0.83, 0.36, 8.12	3010

Table 6 continued

Instance	Hybrid (Bernardino and Paas 2021)		Our matheuristic		Improvements (%)	Time (s)
	Best	Average	Best	Average		
pr1002_2	145326.69	147753.25	144,582.51	146,370.62	0.51, 0.94, 9.60	2592
pr1002_3	126282.93	126736.30	125,589.16	126,121.73	0.55, 0.48, 8.59	3091
pr144_4	49,379	49586.50	49,294	49,410.22	0.17, 0.36, 0.17	20
kroA150_3	21,038	21040.08	21,023	21,015.07	0.07, 0.12, -0.68	16
pr152_3	64,757	64872.95	64,637	64,729.81	0.19, 0.22, -0.33	12
rat195_1	1300	1318.38	1271	1290.62	2.23, 2.11, 1.90	153
rat195_3	1849	1852.62	1841	1849.07	0.43, 0.19, -1.49	19
rat195_4	1330	1449.32	1318	1339.02	0.90, 7.61, 0.15	136
kroA200_1	16,767	16,767	16,749	16,758	0.11, 0.05, -1.87	26
kroA200_3	24,609	24818.74	24,530	24,678.25	0.32, 0.57, -0.24	1822
kroB200_3	24,542	24,542	24,498	24,512	0.18, 0.12, -1.70	185
gr202_4	28,285	28369.11	28,121	28,280.36	0.58, 0.31, -0.29	45
gr229_1	71,612	71696.90	71,570	71,631.53	0.06, 0.09, -1.17	290
gil262_1	1526	1531.81	1506.26	1527.74	1.31, 0.27, 1.29	230
gil262_3	1988	1992.43	1980	1990.61	0.40, 0.09, 0.40	1832
gil262_4	1671	1671	1659	1665	0.72, 0.36, 0.72	1837
pr264_2	28,778	28861.37	28,719	28,752.42	0.21, 0.38, 0.10	73
rbg443_2	350	350	346	349.82	1.15, 0.05, 1.14	175
Average	26890.23	27257.10	26,803.22	25,962.59	0.33, 1.11, 1.62	502.22
p-value			0.25×10^{-4}		0.01×10^{-2}	

Table 7 The percentage (%) of time spent on different parts of our matheuristic

	Parameter setting	GA	SA	POPMUSIC
<i>Instance</i>				
burma14_1	78.25	10.30	5.32	6.13
burma14_2	78.76	10.39	5.48	5.36
burma14_3	79.45	10.47	4.54	5.52
bayg29_1	79.98	10.56	4.87	4.57
bayg29_2	80.78	10.65	3.64	4.91
bayg29_3	81.33	10.74	4.24	3.66
att48_1	81.88	10.85	2.97	4.28
att48_2	82.33	10.91	3.75	2.99
att48_3	75.12	9.63	7.08	8.17
bier127_1	75.82	9.69	7.33	7.14
bier127_2	76.50	9.78	6.33	7.38
bier127_3	76.96	9.85	6.80	6.37
a280_1	77.63	9.92	5.57	6.86
a280_2	78.37	9.97	6.03	5.60
a280_3	79.15	10.04	4.70	6.08
gr666_1	79.89	10.14	5.22	4.74
gr666_2	80.45	10.22	4.05	5.27
gr666_3	81.05	10.29	4.57	4.07
pr1002_1	81.47	10.36	3.56	4.59
pr1002_2	82.25	10.44	3.70	3.59
pr1002_3	82.70	10.51	3.05	3.72
pr144_4	74.71	10.28	6.26	8.75
kroA150_3	75.24	10.33	8.10	6.32
pr152_3	75.63	10.40	5.80	8.15
rat195_1	76.10	10.47	7.57	5.85
rat195_3	76.63	10.52	5.19	7.63
rat195_4	77.25	10.58	6.92	5.24
kroA200_1	77.80	10.66	4.56	6.96
kroA200_3	78.24	10.73	6.42	4.59
kroB200_3	78.89	10.83	3.80	6.46
gr202_4	79.32	10.89	5.95	3.83
gr229_1	73.61	11.84	5.98	8.57
gil262_1	74.03	11.95	7.99	6.01
gil262_3	74.57	12.02	5.36	8.04
gil262_4	75.08	12.08	7.41	5.42
pr264_2	75.71	12.18	4.61	7.48
rbg443_2	76.40	12.25	6.68	4.65
Average	78.09	10.64	5.44	5.80

parameter setting phase, the GA including the evaluation and sorting, SA and POPMUSIC.

It is observed that most of the time (on average 78.09%) is used to set the parameters of the algorithm. This proportion of time is understandable because this initial phase comprises many runs, although their times are limited. Nevertheless, regarding the considerable effect of the parameter values in the performance of the algorithm, investment in the parametrization seems really necessary and beneficial. The required time for the GA is on average 10.64% of the total time. As the GA constitutes the main framework of this algorithm and also contains the objective function calculation of the solutions as well as sorting them accordingly, this share is expected. POPMUSIC consumes on average only 5.80% of the time. This indicates the speed of this approach; the reason is that we only deal with tractable parts of the problem, which are solvable in short times. Finally, the shortest average time (5.44%) belongs to the SA since its aim is to do some fast neighborhood searches around a number of solutions.

By separate analysis of the instances, it can be deduced that more time is needed for parameter tuning of larger or more complex instances and the GA time also slightly increases for them. In terms of POPMUSIC, a trend which shows the increase in time in terms of harder instances is not observed. The reason can lie in the concept of decomposition inherent in POPMUSIC.

5.5 POPMUSIC effects

In the last part, we aim at analyzing the contribution of POPMUSIC and the metaheuristic components in our matheuristic (M). For this sake, we implement it once without the POPMUSIC part (MWP) and once without the genetic operators, i.e. crossover and mutation, and the SA part (MWG). The overall results of all runs of all instances are normalized in the interval $[0, 1]$ based on the best and worst objective value found in the iterations of the matheuristic. So we can show all objective values in the same scale as box plots for the entire matheuristic, MWP, MWG and also the hybrid method of Bernardino and Paías (2021) denoted by H in Fig. 6. It can be observed that eliminating either POPMUSIC or the metaheuristic operators considerably decreases the ability of our approach, even when MWP and MWG are compared to H. It means that the POPMUSIC and the metaheuristic together provide a very good integration and complement each other in searching within the solution space. In other words, the combination of the genetic operators, the SA neighborhood searches, and solving the partial mathematical models in a well-structured framework shows great promise.

Next, we conduct again some pairwise statistical tests to examine the existence of any significant differences between the solution qualities of the analyzed methods. The chosen test is the non-parametric Friedman with Bergman–Hommel post-hoc procedure (Bergmann and Hommel 1988). This is assumed to be one of the best statistical tests when the aim is to compare more than two methods in a pairwise manner (Derrac et al. 2011). The test is implemented in R language. Regarding the low p -values, we reject the null hypotheses, which claim the equality of the average objective values, for all the pairs at the significance level of 0.05.

Fig. 6 The box plots of normalized objective values based on all runs of all instances

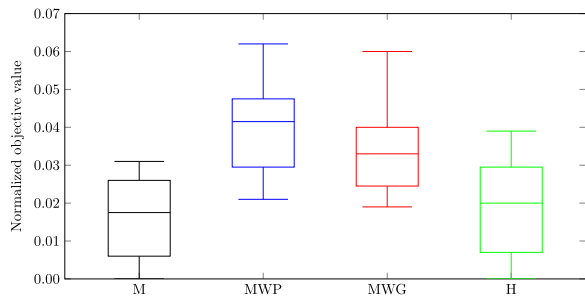


Table 8 *P*-values of the pairwise statistical test based on all results

Pair	<i>P</i> -value
M versus MWP	0.74×10^{-8}
M versus H	0.21×10^{-3}
M versus MWG	0.04×10^{-5}
MWP versus H	0.58×10^{-6}
MWP versus MWG	0.36×10^{-3}
MWG versus H	0.67×10^{-5}

6 Conclusions

This paper has described a new efficient approach to acquire good-quality solutions for a recent generalization of the traveling salesman problem, which has a lot of applications in the real world, namely the family TSP. A matheuristic framework is used in which the searches based on the SA and POPMUSIC concept with mathematical programming are embedded in the iterations of a GA. That is, the advantages of integrating both metaheuristics and an exact optimization procedure are used in our approach, which has enhanced the ability of the search. The proposed method is tested on a large number of benchmark instances and its efficiency is verified by comparing its results with those of the currently best-known state-of-the-art method. The matheuristic has the ability to find optimal solutions for many FTSP instances in short execution times. It owes its favorable performance to the effective interactions between the metaheuristic and the mathematical programming part throughout the progress of the algorithm. We have integrated genetic operators, SA neighborhood searches and mathematical programming of tractable parts of the solutions. They act jointly better than the previously best algorithm from the literature (the hybrid method of Bernardino and Paia [2021](#)), which constructs the solution by providing an optimal solution for only a core part of the problem and then inserting the missing parts. Providing even slightly improved results can be regarded as a considerable success in the research direction. Finally, we proved that the ability of our approach drops sharply if either the POPMUSIC parts or the genetic and SA operators are excluded, providing insights into their interoperability.

In the future, other hybridization schemes for connecting metaheuristics and mathematical programming techniques can be investigated. This may even incorporate specialized solvers rather than a standard solver (like VRP solver for vehicle routing problems; Pessoa et al. 2020). Moreover, matheuristics based on other concepts like the fixed set search algorithm (Jovanovic et al. 2019) can be tried. Since the proper parametrization can positively influence the performance of optimization algorithms, it can be worthwhile to invest much in this part of the developed algorithm.

Acknowledgements This research is funded by dtcc.bw—Digitalization and Technology Research Center of the Bundeswehr. dtcc.bw is funded by the European Union—NextGenerationEU.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability All data used in this work appear in the submitted article or are available from the referenced sites.

Declarations

Conflict of interest None.

Human and animal rights This research did not contain any studies involving animal or human participants, nor did it take place on any private or protected areas. No specific permissions were required for corresponding locations.

Consent to Participate Not applicable.

Consent to publish Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Almeida, F., Gimenez, D., Lopez-Espin, J.J., Perez-Perez, M.: Parameterized schemes of metaheuristics: basic ideas and applications with genetic algorithms, scatter search, and GRASP. *IEEE Trans. Syst. Man Cybern. Syst.* **43**, 570–586 (2013). <https://doi.org/10.1109/tsmca.2012.2217322>
- Alvim, A.C., Taillard, É.D.: POPMUSIC for the point feature label placement problem. *Eur. J. Oper. Res.* **192**, 396–413 (2009). <https://doi.org/10.1016/j.ejor.2007.10.002>
- Bergmann, B., Hommel, G.: Improvements of general multiple test procedures for redundant systems of hypotheses. In: Bauer, P., Hommel, G., Sonnemann, E. (eds.) *Multiple Hypothesenprüfung/Multiple Hypotheses Testing*, pp. 100–115. Springer, Berlin (1988). https://doi.org/10.1007/978-3-642-52307-6_8
- Bernardino, R., Paia, A.: Solving the family traveling salesman problem. *Eur. J. Oper. Res.* **267**, 453–466 (2018). <https://doi.org/10.1016/j.ejor.2017.11.063>

- Bernardino, R., Paias, A.: Heuristic approaches for the family traveling salesman problem. *Int. Trans. Oper. Res.* **28**, 262–295 (2021). <https://doi.org/10.1111/itor.12771>
- Bernardino, R., Paias, A.: The family traveling salesman problem with incompatibility constraints. *Networks* **79**, 47–82 (2022). <https://doi.org/10.1002/net.22036>
- Bernardino, R., Gouveia, L., Paias, A., Santos, D.: The multi-depot family traveling salesman problem and clustered variants: mathematical formulations and branch-&-cut based methods. *Networks* **80**, 502–571 (2022). <https://doi.org/10.1002/net.22125>
- Blickle, T., Thiele, L.: A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.* **4**, 361–394 (1996). <https://doi.org/10.1162/evco.1996.4.4.361>
- Bosco, A., Laganà, D., Musmanno, R., Vucatur, F.: A matheuristic algorithm for the mixed capacitated general routing problem. *Networks* **64**, 262–281 (2014). <https://doi.org/10.1002/net.21574>
- Box, G.E.P., Draper, N.R.: *Response Surfaces, Mixtures, and Ridge Analyses*, 2nd edn. Wiley, Hoboken (2007)
- Cacchiani, V., Muritiba, A.E.F., Negreiros, M., Toth, P.: A multistart heuristic for the equality generalized traveling salesman problem. *Networks* **57**, 231–239 (2010). <https://doi.org/10.1002/net.20421>
- Caserta, M., Voß, S.: A corridor method based hybrid algorithm for redundancy allocation. *J. Heuristics* **22**, 405–429 (2014). <https://doi.org/10.1007/s10732-014-9265-y>
- Choi, I.-C., Kim, S.-I., Kim, H.-S.: A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem. *Comput. Oper. Res.* **30**, 773–786 (2003). [https://doi.org/10.1016/s0305-0548\(02\)00050-3](https://doi.org/10.1016/s0305-0548(02)00050-3)
- Conover, W.: *Practical Nonparametric Statistics*. Wiley, New York (1999)
- de Armas, J., Lalla-Ruiz, E., Tilahun, S.L., Voß, S.: Similarity in metaheuristics: a gentle step towards a comparison methodology. *Nat. Comput.* **21**, 265–287 (2022). <https://doi.org/10.1007/s11047-020-09837-9>
- de Oliveira, L.G., de Paiva, A.P., Balestrassi, P.P., Ferreira, J.R., da Costa, S.C., da Silva Campos, P.H.: Response surface methodology for advanced manufacturing technology optimization: theoretical fundamentals, practical guidelines, and survey literature review. *Int. J. Adv. Manuf. Technol.* **104**, 1785–1837 (2019). <https://doi.org/10.1007/s00170-019-03809-9>
- Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **1**, 3–18 (2011). <https://doi.org/10.1016/j.swevo.2011.02.002>
- Doi, T., Nishi, T., Voß, S.: Two-level decomposition-based matheuristic for airline crew rostering problems with fair working time. *Eur. J. Oper. Res.* **267**, 428–438 (2018). <https://doi.org/10.1016/j.ejor.2017.11.046>
- Galli, L., Martello, S., Rey, C., Toth, P.: Lagrangian matheuristics for the quadratic multiple knapsack problem. *Discret. Appl. Math.* **335**, 36–51 (2023). <https://doi.org/10.1016/j.dam.2022.06.033>
- Gurobi Optimization, LLC (2021). *Gurobi Optimizer Reference Manual*. <https://www.gurobi.com/?s=optimizer+reference+manual>
- Haupt, R.L., Haupt, S.E.: *Practical Genetic Algorithms*. Wiley, New York (2003). <https://doi.org/10.1002/0471671746>
- Hill, A., Voß, S.: An equi-model matheuristic for the multi-depot ring star problem. *Networks* **67**, 222–237 (2016). <https://doi.org/10.1002/net.21674>
- Hutter, F., Hamadi, Y.: Parameter Adjustment Based on Performance Prediction: Towards an Instance-Aware Problem Solver. Technical Report Microsoft Corporation, MSR-TR-2005-125 (2005)
- Hutter, F., Xu, L., Hoos, H.H., Leyton-Brown, K.: Algorithm runtime prediction: methods and evaluation. *Artif. Intell.* **206**, 79–111 (2014). <https://doi.org/10.1016/j.artint.2013.10.003>
- Jovanovic, R., Tuba, M., Voß, S.: Fixed set search applied to the traveling salesman problem. In: Blesa-Aguilera, M.J., Blum, C., Gambini-Santos, H., Pinacho-Davidson, P., del Campo, J.G. (eds.) *Hybrid Metaheuristics*, pp. 63–77. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05983-5_5
- Karp, R. M.: Reducibility among combinatorial problems. In: *Complexity of Computer Computations*, pp. 85–103. Springer US (1972). https://doi.org/10.1007/978-1-4684-2001-2_9
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**, 671–680 (1983). <https://doi.org/10.1126/science.220.4598.671>
- Lalla-Ruiz, E., Voß, S.: POPMUSIC as a matheuristic for the berth allocation problem. *Ann. Math. Artif. Intell.* **76**, 173–189 (2014). <https://doi.org/10.1007/s10472-014-9444-4>
- Lalla-Ruiz, E., Voß, S.: A POPMUSIC approach for the multi-depot cumulative capacitated vehicle routing problem. *Optim. Lett.* **14**, 671–691 (2019). <https://doi.org/10.1007/s11590-018-1376-1>

- Lalla-Ruiz, E., Voß, S., Expósito-Izquierdo, C., Melián-Batista, B., Moreno-Vega, J.M.: A POPMUSIC-based approach for the berth allocation problem under time-dependent limitations. *Ann. Oper. Res.* **253**, 871–897 (2015). <https://doi.org/10.1007/s10479-015-2055-6>
- Leggieri, V., Haouari, M.: A matheuristic for the asymmetric capacitated vehicle routing problem. *Discrete Appl. Math.* **234**, 139–150 (2018). <https://doi.org/10.1016/j.dam.2016.03.019>
- Maniezzo, V., Stützle, T., Voß, S. (eds.): *Matheuristics—Hybridizing Metaheuristics and Mathematical Programming*. Springer, New York (2010). <https://doi.org/10.1007/978-1-4419-1306-7>
- Maniezzo, V., Boschetti, M.A., Stützle, T.: *Matheuristics*. Springer, Berlin (2021). <https://doi.org/10.1007/978-3-030-70277-9>
- Matai, R., Singh, S., Lal, M.: Traveling salesman problem: an overview of applications, formulations, and solution approaches. In: Davendra D. (eds) *Traveling Salesman Problem, Theory and Applications*, pp. 1–24. InTech (2010). <https://doi.org/10.5772/12909>
- Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. *J. ACM* **7**, 326–329 (1960). <https://doi.org/10.1145/321043.321046>
- Morán-Mirabal, L., González-Velarde, J., Resende, M.: Randomized heuristics for the family traveling salesperson problem. *Int. Trans. Oper. Res.* **21**, 41–57 (2014). <https://doi.org/10.1111/itor.12026>
- Nagar, A., Heragu, S.S., Haddock, J.: A combined branch-and-bound and genetic algorithm based approach for a flowshop scheduling problem. *Ann. Oper. Res.* **63**, 397–414 (1996). <https://doi.org/10.1007/bf02125405>
- Nishi, T., Okura, T., Lalla-Ruiz, E., Voß, S.: A dynamic programming-based matheuristic for the dynamic berth allocation problem. *Ann. Oper. Res.* **286**, 391–410 (2017). <https://doi.org/10.1007/s10479-017-2715-9>
- Ostertag, A., Doerner, K.F., Hartl, R.F., Taillard, É.D., Waelti, P.: POPMUSIC for a real-world large-scale vehicle routing problem with time windows. *J. Oper. Res. Soc.* **60**, 934–943 (2009). <https://doi.org/10.1057/palgrave.jors.2602633>
- Pessoa, A., Sadykov, R., Uchoa, E., Vanderbeck, F.: A generic exact solver for vehicle routing and related problems. *Math. Program.* **183**, 483–523 (2020). <https://doi.org/10.1007/s10107-020-01523-z>
- Poojari, C., Beasley, J.: Improving benders decomposition using a genetic algorithm. *Eur. J. Oper. Res.* **199**, 89–97 (2009). <https://doi.org/10.1016/j.ejor.2008.10.033>
- Queiroga, E., Sadykov, R., Uchoa, E.: A POPMUSIC matheuristic for the capacitated vehicle routing problem. *Comput. Oper. Res.* **136**, 105475 (2021). <https://doi.org/10.1016/j.cor.2021.105475>
- Reyes Vega, K. A.: Solving the family traveling salesman problem with capacitated agents. Master thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico (2014). <https://hdl.handle.net/11285/640275>
- Su, Z., Lü, Z., Wang, Z., Qi, Y., Benlic, U.: A matheuristic algorithm for the inventory routing problem. *Transp. Sci.* **54**, 330–354 (2020). <https://doi.org/10.1287/trsc.2019.0930>
- Swan, J., Adriaensen, S., Brownlee, A.E., Hammond, K., Johnson, C.G., Kheiri, A., Krawiec, F., Merelo, J., Minku, L.L., Özcan, E., Pappa, G.L., García-Sánchez, P., Sörensen, K., Voß, S., Wagner, M., White, D.R.: Metaheuristics “in the large”. *Eur. J. Oper. Res.* **297**, 393–406 (2022). <https://doi.org/10.1016/j.ejor.2021.05.042>
- Taillard, É. D., Voss, S.: POPMUSIC—partial optimization metaheuristic under special intensification conditions. In: Ribeiro, C., Hansen, P. (eds.), *Essays and Surveys in Metaheuristics*, pp. 613–629. Springer, Boston (2002). https://doi.org/10.1007/978-1-4615-1507-4_27
- Taillard, É.D.: Design of Heuristic Algorithms for Hard Optimization. Springer International Publishing (2023). <https://doi.org/10.1007/978-3-031-13714-3>
- Taillard, E. D.: TSP neighbourhood reduction with POPMUSIC. In: *Metaheuristic International Conference (MIC’17) Proceedings*, pp. 237–240 (2017)
- Taillard, É.D.: A linearithmic heuristic for the travelling salesman problem. *Eur. J. Oper. Res.* **297**, 442–450 (2022). <https://doi.org/10.1016/j.ejor.2021.05.034>
- Taillard, É.D., Helsgaun, K.: POPMUSIC for the travelling salesman problem. *Eur. J. Oper. Res.* **272**, 420–429 (2019). <https://doi.org/10.1016/j.ejor.2018.06.039>
- Vadseth, S.T., Andersson, H., Stålhane, M.: An iterative matheuristic for the inventory routing problem. *Comput. Oper. Res.* **131**, 105262 (2021). <https://doi.org/10.1016/j.cor.2021.105262>

- Xu, X., Li, J., Zhou, M., Yu, X.: Precedence-constrained colored traveling salesman problem: an augmented variable neighborhood search approach. *IEEE Trans. Cybern.* **52**, 9797–9808 (2022). <https://doi.org/10.1109/tcyb.2021.3070143>
- Yu, J., Voß, S., Song, X.: Multi-objective optimization of daily use of shore side electricity integrated with quayside operation. *J. Clean. Prod.* **351**, 131406 (2022). <https://doi.org/10.1016/j.jclepro.2022.131406>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.