A two-level diploid genetic based algorithm for solving the family traveling salesman problem

Petrică Pop
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
petrica.pop@cunbm.utcluj.ro

Oliviu Matei
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
oliviu.matei@holisun.com

Camelia Pintea
Technical University of Cluj-Napoca, North University Center at Baia Mare
Baia Mare, Romania
dr.camelia.pintea@ieee.org

ABSTRACT

In this paper, we consider the Family Traveling Salesman Problem (FTSP), which is a variant of the classical Traveling Salesman Problem (TSP). Given a partition of the nodes into a predefined number of clusters, called families, the aim of the FTSP is to find a minimum cost tour visiting a given number of nodes from each family. We describe a novel solution approach for solving the FTSP obtained by decomposing the problem into two smaller subproblems: a macro-level subproblem and a micro-level subproblem, and solving them separately. The goal of the first subproblem is to provide tours visiting the families using a classical genetic algorithm and a diploid genetic algorithm, while the aim of the second subproblem is to find the minimum-cost tour, corresponding to the above mentioned tours, visiting a given number of nodes from each family. The second subproblem is solved by transforming each global tour into a traveling salesman problem (TSP) which then is optimally computed using the Concorde TSP solver. The preliminary computational results on a usually used set of benchmark instances prove that our solution approach provides competitive solutions in comparison to the existing methods for solving the FTSP.

CCS CONCEPTS

• **Computing methodologies** → **Genetic algorithms**; • **Theory of computation** → *Bio-inspired optimization*; • **Mathematics of computing** → Combinatorial optimization;

KEYWORDS

Combinatorial optimization, traveling salesman problem, generalized traveling salesman problem, family traveling salesman problem, diploid genetic algorithms

ACM Reference Format:
Petrică Pop, Oliviu Matei, and Camelia Pintea. 2018. A two-level diploid genetic based algorithm for solving the family traveling salesman problem. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3205455.3205545

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205545

1 INTRODUCTION

1.1 Problem description and related problems

This paper focuses on the family traveling salesman problem (FTSP), which is a variant of the classical traveling salesman problem (TSP), where the nodes are split into a predefined number of clusters, called families. The FTSP tries to find a minimum cost tour visiting a predefined number of nodes from each family and with the additional constraint that the nodes within each family must be visited contiguously.

The considered optimization problem belongs to the class of generalized combinatorial optimization problems, known also as the class of generalized network design problems. This class of problems extends the classical combinatorial optimization problems in a natural way and its main characteristics are the following: the nodes of the underlying graph are split into clusters and the feasibility constraints of the original optimization problem are expressed in terms of the clusters instead of individual nodes. For more information on the class of generalized combinatorial optimization problems we refer to Pop [15], Feremans et al. [4].

Taking into account its definition, the FTSP is closely related to the following combinatorial optimization problems:

- *the clustered traveling salesman problem (CTSP)* introduced by Chisman Chisman [2] and defined on an undirected graph whose set of nodes is partitioned into a predefined number of clusters. The goal of the CTSP is to determine the lowest cost Hamiltonian tour with the additional constraint that the nodes of each cluster are visited consecutively. The FTSP is an extension of the CTSP in the sense that from each family (cluster) we must visit a given number of nodes. For more information on the CTSP we refer to Potvin and Guertin [18].
- *the generalized traveling salesman problem (GTSP)* which was introduced independently by Henry-Labordere Henry-Labordere [7] and Srivastava et al. Srivastava et al. [20] and defined as follows: given a graph whose nodes are grouped into a number of predefined clusters, the GTSP aims at finding the minimum cost tour visiting exactly one node from each cluster. The FTSP extends the GTSP in the sense that from each family (cluster) we must visit a given number of nodes. If in the FTSP we have to visit one node per family, than the problem reduces to the GTSP. For more information on the GTSP we refer to Pintea et al. [10], Pop and Iordache [16], Pop et al. [17].
- *the generalized covering salesman problem (GCSP)* introduced by Golden et al. Golden et al. [6], where each node may cover

a given subset of nodes, and it has in addition a prespecified demand, and we want to determine a minimum cost tour with the property that the subset of nodes visited on the tour covers all the nodes of the graph. GTSP is a special case of the GCSP and therefore the investigated problem FTSP can be seen as a variant of the GCSP. For more information on the GCSP we refer to Shaelaie et al. [19].

1.2 Literature review and applications

The existing literature regarding FTSP is rather scarce: the problem was introduced by Moran-Mirabal et al. Morán-Mirabal et al. [9] motivated by a practical application, namely for optimizing the order of picking in warehouses. The same authors provided an integer programming model of the FTSP and solved it with CPLEX 11 for small instances of the problem, and in addition they proposed two randomized heuristic algorithms: a biased random-key genetic algorithm and a GRASP with evolutionary path-relinking. Recently, Bernardino and Paias Bernardino and Paias [1] described several compact and non-compact integer and mixed integer programming formulations of the FTSP and developed an iterated local search algorithm for solving the problem.

Although there is not much literature on the FTSP, due to its complexity and close relation to several combinatorial optimization problems that have various practical applications, we consider that the FTSP is worth to be investigated.

1.3 Overview

The aim of this paper is to describe a novel two-level optimization approach for Family Traveling Salesman Problem. Our approach is obtained by decomposing the problem into two logical and natural subproblems: a macro-level (global) subproblem and a micro-level (local) subproblem. The first subproblem aims at determining the Hamiltonian tours visiting the families, called global tours, using a genetic algorithm, respectively a diploid genetic algorithm applied to the corresponding global graph (see details in Section 3), while the aim of the second subproblem is to determine the visiting order of the predefined number of nodes within the families for the above mentioned Hamiltonian tour. The second subproblem is solved by transforming each global Hamiltonian tour into a classical TSP which then is computed optimally using the Concorde TSP solver. The results of our preliminary computational experiments on the existing benchmark instances from the literature are presented and analyzed.

Our paper is organized as follows. In Section 2, we give some notations and definitions related to the Family Traveling Salesman Problem that will be used throughout the paper. The two-level diploid genetic based algorithm for solving the FTSP is described in Section 3 and preliminary computational experiments and the achieved results are presented and discussed in Section 4. Finally, the conclusions are depicted in Section 5.

2 DEFINITION OF THE FAMILY TRAVELING SALESMAN PROBLEM

In this section we provide an explicit definition of the Family Traveling Salesman Problem as a graph theoretic model.

Let $G = (N, E)$ be an undirected graph, which we assume to be complete, with $N = \{0\} \cup V$ as the set of nodes and E the set of edges.

The nodes belonging to V correspond to the customers and the node 0 corresponds to the depot. The entire set of nodes N is partitioned into $k + 1$ mutually exclusive nonempty subsets, called families and denoted by $V_0, V_1, ..., V_k$, i.e. the following conditions hold:

1. $V = V_0 \cup V_1 \cup ... \cup V_k$
2. $V_p \cap V_q = \emptyset$ for all $p, q \in \{0, 1, ..., k\}$ and $p \neq q$.

and with the additional condition the family V_0 is a singleton, it contains only the vertex 0, which represents the depot. Therefore the remaining nodes from V belong to the families $V_1, ..., V_k$.

The number of members belonging to family V_i is denoted by n_i and the following relation holds: $\sum_{i=1}^{k} n_i = |V|$. We have to visit nv_i nodes from each family V_i and the total number of nodes that are required to be visited is denoted by NV, $NV = \sum_{i=1}^{k} nv_i$.

We define two kind of edges: edges between nodes belonging to the same family, called intra-cluster edges and edges between nodes belonging to different families, called inter-cluster edges. A nonnegative cost is associated with each edge $e \in E$ and the cost of a tour is equal to the sum of the costs of all the edges belonging to that tour.

The *Family Traveling Salesman Problem* consists in finding a minimum cost tour visiting visiting a given number of nodes from each family such that the following constraints hold:

- each tour starts and ends at the depot;
- once a salesman enters a family, it visits the required nodes within the family before leaving it.

An illustration of the investigated family traveling salesman problem and a feasible solution of the problem are presented in the next figure.

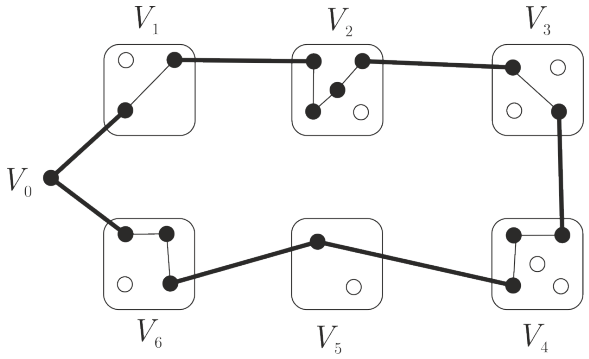

Figure 1: An example of a feasible solution of FTSP consisting of 24 nodes partitioned into 6 families and the depot

We will call such a tour with the property that it visits contiguously a predefined number of nodes from each family a *family tour*.

The FTSP is NP-hard optimization problem because it includes the classical TSP as a special case when all the families contain only one member.

3 THE TWO-LEVEL DIPLOID GENETIC BASED ALGORITHM FOR SOLVING FTSP

In this section we present our novel approach for solving the FTSP obtained by decomposing the problem into two logical and natural smaller subproblems:

1. a macro-level (global) subproblem which provides global tours visiting the families using a classical GA and a diploid genetic algorithm, denoted by (2GA),
2. a micro-level (local) subproblem whose aim is to find for the above mentioned global tours the visiting order of the required nodes within the families.

Our two-level approach takes advantage of the special structure of the FTSP, i.e. the nodes of the graph are partitioned into a given number of families, and offers computational advantages by using efficient methods for solving the subproblems and by combining the achieved results in order to provide a solution for the FTSP without using any post-processing procedure.

It should be mentioned that one of the main advantages of our method concerns the micro-level subproblem, namely our approach provides not only an optimal way of visiting the required numbers of nodes (i.e. intra-cluster tours) within the families, but also the optimal way of visiting the families.

The two-level solution approach proved its efficiency by solving various complex generalized network design problems, see for example Expósito-Izquierdo et al. [3], Pop et al. [11, 13].

3.1 The macro-level subproblem

In order to define the macro-level subproblem, as in Pop [14, 15], we denote by G' the graph obtained from G after replacing all the nodes of a family V_i with a supernode representing V_i, $\forall i \in \{1, ..., k\}$, the cluster V_0 (depot) consists already of one node. We will call the graph G' the *global graph*. For convenience, we identify V_i with the supernode representing it. The edges of the graph G' are defined between each pair of the graph vertices $V_0, V_1, \ldots, V_k$.

We use a classical genetic algorithm and a diploid genetic algorithm applied to the corresponding global graph in order to provide a Hamiltonian tour visiting the families. We will call such a tour a *global Hamiltonian tour*. One of the main advantages of using this approach is the considerable reduction of the solution space of the original problem.

In the next figure we represent a global Hamiltonian tour corresponding to the example presented in Figure 1.

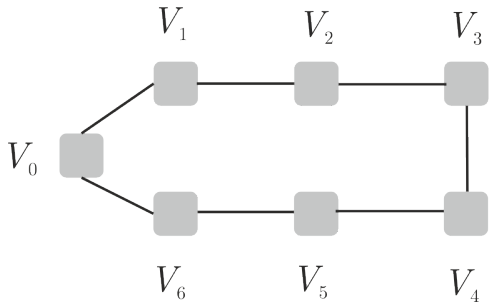

Figure 2: A feasible solution in the global graph

There are several family tours corresponding to a global Hamiltonian tour visiting the families in a given order. Between these family tours there exists one called the best family tour (w.r.t. cost minimization) that will be determined using an efficient transformation of the global Hamiltonian tour into a classical TSP which then is optimally computed using the Concorde TSP solver.

3.2 The genetic algorithm

In this subsection we describe the genetic algorithm that provides us with the global Hamiltonian tours visiting the families.

3.2.1 Representation. We use an efficient representation at the level of the global graph, in which the chromosome for each candidate solution is represented as an array of $k + 1$ integer numbers, each number corresponding to the index of a certain family. The depot is represented by 0. Each global Hamiltonian route is ordered in conformity with the order in which the families are visited. Therefore, in general a chromosome is represented as follows:

$$(0 \quad i_1 \quad i_2 \quad \ldots \quad i_{k-1} \quad i_k).$$

For example, for the solution presented in Figure 2, the corresponding chromosome is represented by the following array:

$$(0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6).$$

Each chromosome representing the order in which the families are visited can be represented at the level of the families as follows:

$$\left(1, \underbrace{a_1, a_2, \ldots, a_{n_{i_1}}}_{V_{i_1}}, \underbrace{a_{n_{i_1}+1}, \ldots, a_{n_{i_1}+n_{i_2}}}_{V_{i_2}}, \ldots, \underbrace{a_{\sum_{l=1}^{k-1} n_{i_l}+1}, \ldots, a_{\sum_{l=1}^{k} n_{i_l}}}_{V_{i_k}} \right)$$

where $a_i \in \{0, 1\}, \forall i \in \left\{1, \ldots, \sum_{l=1}^{k} n_{i_l}\right\}$ and $\{i \in V_{i_l} | a_i = 1\} = nv_{i_l}$, and can be expanded to the corresponding best family tour using the transformation that is going to be described in Subsection 3.3.

3.2.2 Fitness evaluation. The fitness function of each individual chromosome in the population is given by the total length of the best corresponding family tour associated to the global Hamiltonian tour specified by the chromosome. This distance also takes into account the order in which the required nodes within the families are visited. The aim of the FTSP is to minimize this total distance.

3.2.3 Initial population. Our initial population was generated randomly from the space of feasible solutions (global Hamiltonian tours) thus providing an unbiased initial population.

3.2.4 Selection. In our algorithm we considered a deterministic selection process. The method used is a $(\mu + \lambda)$ selection which works as follows: the best μ individuals are selected from the pool of μ parents and λ offspring. This method is well known to stick in local optima. However, we overcome this shortcoming by using the diploid representation of the individuals.

3.2.5 Crossover. Our GA uses a custom version of the one cut point crossover. The crossover function takes two parent candidate solutions as input, and outputs two solutions that are derived from the parents. The crossover is applied to the population obtained

from the selection algorithm, we randomly select a cut point between 1 and k, and we create two offspring that preserve the order and indexes of the families in a subsequence of one parent while preserving the relative order of the remaining indexes of the families from the other parent, omitting any symbols that were copied already from the first parent.

3.2.6 Mutation. After obtaining the offspring from the crossover algorithm, a mutation operator is applied with a given probability, set before solving the problem. In our case we use a swapping intra-cluster mutation operator which acts as follows: we randomly select two genes of different values and their values are exchanged. As a consequence, we modify the nodes required to be visited within a family.

3.2.7 Genetic parameters. The parameters included in our GAs are very important for the success of our algorithm. Based on preliminary computational experiments, we set the following parameters: the population size μ has been set at 300, the mutation probability was set at 5% and the maximum number of generations (epochs) in our algorithm was set at 5000.

3.3 The diploid genetic algorithm

In our considered diploid genetic algorithm an individual consists not only of a single solution (chromosome), but in a pair of solution, as it was suggested by Pop et al. Pop et al. [12]. In this way, we mimic the natural diploid individuals Mitchell [8]. The advantage of this representation is that each individual carries twice as much information as in the classical approach (called also "haploid"). This assures a higher diversity of the potential feasible solutions Goldberg and Smith [5], which, in turn, leads to higher chances of avoidance of the local optima by the population. This is because every time a solution is selected for its good fitness, it carries along a worse solution, both belonging to the same individual.

Therefore, unlike the classical individuals, which were synonymous with chromosomes, now we talk about individuals consisting of a pair of chromosomes:

$$I = (C_1, C_2),$$

where C_i are the chromosomes, with $i \in \{1, 2\}$. Actually, both chromosomes are potential feasible solutions belonging to the space of solutions (i.e. global Hamiltonian tours).

Further we define $C_d \in \{C_1, C_2\}$ as the dominant chromosome if it is the best out of the two chromosomes and $C_r \in \{C_1, C_2\}$ as the recessive chromosome if it is the worst out of the two.

3.3.1 Fitness function. Let us denote the fitness of the individual by $f(I)$ and subsequently the fitness of the two chromosomes by $f(C_i)$, with $i \in \{1, 2\}$. Of course, according to the definition provided in this section, we have $f(C_d) \leq f(C_r)$.

According to Pop et al. Pop et al. [12], there are different ways of defining the fitness of an individual:

(1) the fitness of the individual is the fitness of the dominant chromosome, that is: $f(I) = f(C_d)$;
(2) the fitness of the individual is the fitness of the recessive chromosomes: $f(I) = f(C_r)$;
(3) the fitness of the individual is the sum (or average) of the fitness values of the two chromosomes: $f(I) = f(C_1) + f(C_2)$;
(4) the fitness of the individual is a weighted average of the fitness values of the two chromosomes:

$$f(I) = w_1 \cdot f(C_1) + w_2 \cdot f(C_2),$$

where $w_i = \frac{f(C_i)}{f(C_1)+f(C_2)}$ with $i \in \{1, 2\}$.

In our approach, we decide for the third variant, namely the fitness of an individual is the sum of the fitness values of the two chromosomes.

3.3.2 Crossover. Having two individuals $I_1 = (C_1^1, C_2^1)$ and $I_2 = (C_1^2, C_2^2)$, we can define the recombination operator. For a clear distinction between the crossover at the individual level, respectively at the solution (chromosomes) level, we will define them as *individual crossover* or *macro-crossover* at the individual level, respectively *solution crossover*, *chromosomal crossover* or *micro-crossover* at the solution level.

The macro-recombination is defined in a discrete way, only the chromosomes interchange between the two individuals. Having $I_1 = (C_1^1, C_2^1)$ and $I_2 = (C_1^2, C_2^2)$, the offspring are:

$$O_1 = (C_1^{11}, C_2^{11}) \text{ and } O_2 = (C_1^{22}, C_2^{22}),$$

where C_i^{kk} may be any of the parental chromosomes $\{C_1^1, C_2^1, C_1^2, C_2^2\}$.

At the solution level, a micro-crossover is applied in a similar way as it was already mentioned in the case of the classical genetic algorithm and the mutation is applied at the level of the chromosome.

3.4 The micro-level subproblem

In this subsection, we describe an efficient transformation of a global Hamiltonian tour into a classical TSP. The reason for such a transformation is based on the fact that we can optimally solve large instances of the TSP by using the Concorde TSP solver.

The basic idea used in our transformation is to add an artificial cost M to all the inter-cluster edges, this way forcing the salesman to visit all the required nodes within the family before leaving it.

We consider that we have a global Hamiltonian tour visiting families. Then we define the TSP on the subgraph G' associated to G as follows:

1. The set of nodes of G and G' are the same.
2. The entries of the cost matrix G' are defined as follows:
   a) if $v_i, v_j \in V_l$ then $c'(v_i, v_j) = c(v_i, v_j)$
   b) if $v_i \in V_q, v_j \in V_l$ with $q \neq l$ then $c'(v_i, v_j) = c(v_i, v_j) + M$ where $M > \sum_{(v_i, v_j) \in E(G)} c(v_i, v_j)$.

Obviously, there exists a one-to-one correspondence between Hamiltonian cycles in G' and family tours in G. This correspondence is depicted in Figure 3.

Based on the definition of the cost matrix in G' it results straightforward that the cost of the family tour in G is equal to the cost of the Hamiltonian tour in G' less $(k + 1)M$.

As we have already mentioned, an important advantage of our proposed method to solving the lower-level (local) subproblem is the fact that our approach provides not only an optimal way of

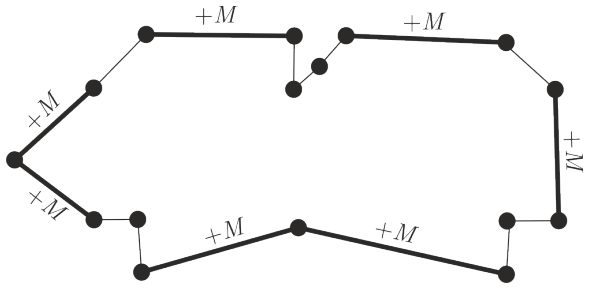

Figure 3: The corresponding Hamiltonian tour in G' to the family tour presented in Figure 1

visiting the required nodes within the families but also the optimal way of visiting the families.

4 PRELIMINARY COMPUTATIONAL RESULTS

In this section we present our preliminary computational results in order to determine the efficiency of our proposed hybrid diploid genetic algorithm for solving the FTSP.

We conducted our computational experiments for solving the FTSP on a set of 12 often used benchmark instances generated by Moran-Mirabal et al. Morán-Mirabal et al. [9]. The instances were generated using instances from TSPLIB as follows: a number k of families was chosen, the number of members n_i for each family V_i was selected uniformly at random respecting the condition: $\sum_{i=1}^{k} n_i = |V|$ and total number of nodes nv_i that are required to be visited from each family was selected at random such that $1 \leq nv_i \leq n_i$ and $NV = \sum_{i=1}^{k} nv_i$. The set of 12 instances is composed by three instances for each combination of the form: $(|V|, k, kN)$ of nodes, families and visits. The number of visits differ from one instance to the next ones within the same class.

Our proposed solution approach for solving the FTSP has been implemented, and in our experiments we performed 10 independent runs for each instance. The testing machine was a Dell Inspiron 15, with Intel i5 2.2GHz and 8GB RAM and the algorithm was developed in Java.

In order to study the performance of our proposed solution approaches, we compared them with the existing heuristic algorithms from the literature.

The obtained computational results are presented in Table 1. The first column of Table 1 gives the name of the instance, followed by four columns that provide the best and average solutions achieved by the biased random-key genetic algorithm (BRKGA) and the greedy randomized adaptive search procedure (GRASP) with evolutionary path-relinking (evPR) developed by Moran-Mirabal et al. Morán-Mirabal et al. [9]. The last four columns contain the best and average solutions obtained by our solution approaches: the classical genetic algorithm (GA) and the diploid genetic algorithm (2GA). The results written in bold represent cases for which the obtained solution is the best existing from the literature.

Analyzing the reported computational results we observe that our two-level diploid genetic based algorithm is competitive in comparison with the existing heuristics for solving the FTSP in terms of the achieved solution quality. We were able to improve the existing solutions 3 out of 12 instances. As well we can observe the superiority of the proposed diploid genetic algorithm in comparison with the classical GA.

Next, we present a statistical analysis using the T-test for dependent samples. The paired sample t-test null hypothesis assumes that the true mean difference is equal to zero. The significance is based on the probability p of observing the test results under the null hypothesis. A low p-value indicates decreased support for the null hypothesis. In the current paper the cutoff value for determining statistical significance is considered 0.1; this corresponds to a 10% (or less) chance of having a result like the one observed if the null hypothesis was true.

Given the best solutions reported on Table 1, we calculate the $t - value$ and probability p as follows:

- BRKGA vs. 2GA: the $t-value$ is -1.489738 and the probability is 0.082198;
- GRASP+evPR vs 2GA: the $t - value$ is -1.669544 and the probability is 0.061595;
- GA vs. 2GA: the $t - value$ is -1.483055 and the probability is 0.083067.

All the above comparisons show that the achieved results are significant for $p \leq 0.10$.

Furthermore, the results of the T-test statistical analysis between the reported average solution results from Table 1.

- BRKGA vs. 2GA: the $t-value$ is -0.679729 and the probability is 0.255364;
- GRASP+evPR vs 2GA: the $t - value$ is -0.670789 and the probability is 0.258096;
- GA vs. 2GA: the $t - value$ is -2.959288 and the probability is 0.006496.

For the first two comparisons the result is not significant for $p \leq 0.10$, while for GA vs. 2GA the result is significant for $p \leq 0.10$.

In the next two figures we present a comparison of the Relative Percentage Differences (RPD) between the achieved best and average solutions. The RPD is computed using the following formula:

$$RPD = \frac{Solution - Best}{Best},$$

where Solution is solution provided by each of considered algorithm: BRKGA, GRASP+evPR and GA and Best is the best solution obtained by 2GA.

Some important features of our proposed solution approaches for solving the FTSP are:

- the use of a two-level approach that decomposes the problem into two logical and natural smaller subproblems: a macro-level (global) subproblem and a micro-level (local) subproblem;
- the use of an efficient GA which delivers the global Hamiltonian tours visiting the clusters;
- the use of a diploid representation that assures a high diversity of the potential feasible solutions;
- the use of an efficient method in order to determine the best corresponding family tour for a given global Hamiltonian tour.

Table 1: The comparison of the solvers quality

Instance	BRKGA		GRASP + evPR		GA		2GA	
	Best sol.	Avg. sol.	Best sol.	Avg. sol.	Best sol.	Avg. sol.	Best sol.	Avg. sol.
burma_1	**13.93**	**13.93**	**13.93**	**13.93**	**13.93**	**13.93**	**13.93**	**13.93**
burma_2	**25.66**	**25.66**	**25.66**	**25.66**	**25.66**	26.74	**25.66**	**25.66**
burma_3	**11.89**	**11.89**	**11.89**	**11.89**	**11.89**	**11.89**	**11.89**	**11.89**
bayg_1	**5345.86**	**5345.86**	**5345.86**	**5345.86**	**5345.86**	5594.36	**5345.86**	5421.15
bayg_2	**5791.01**	**5791.01**	**5791.01**	**5791.01**	**5791.01**	5845.88	**5791.01**	5816.34
bayg_3	**5544.33**	**5544.33**	**5544.33**	**5544.33**	**5544.33**	5586.66	**5544.33**	5557.64
att_1	**23686.02**	**23686.02**	**23686.02**	**23686.02**	**23686.02**	24763.45	**23686.02**	24126.79
att_2	**20609.09**	**20609.09**	20635.57	20635.57	**20609.09**	21135.42	**20609.09**	20746.55
att_3	**9024.58**	**9024.58**	**9024.58**	**9024.58**	**9024.58**	9115.60	**9024.58**	**9024.58**
bier_1	36913.74	36950.75	36800.39	36856.17	36778.50	37421.23	**36241.75**	36673.15
bier_2	98216.10	98333.46	97615.41	98370.63	96224.16	97464.85	**95946.62**	96574.68
bier_3	50513.10	50891.36	50715.49	50920.77	50066.42	51264.61	**50024.16**	50954.28

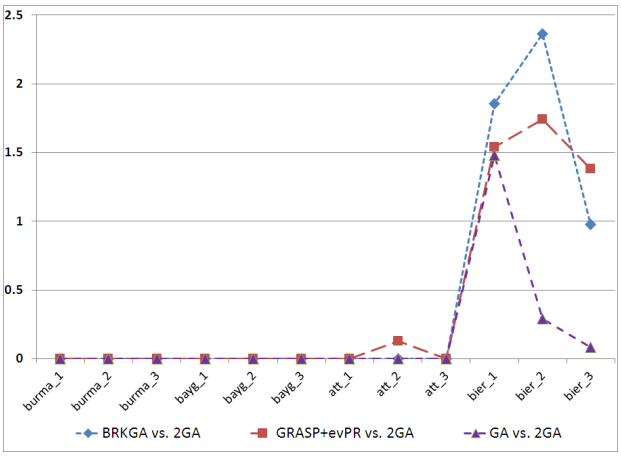

Figure 4: A comparison of the Relative Percentage Differences between the Best solutions achieved by BRKGA, GRASP+evPR and GA and the best solutions obtained by 2GA

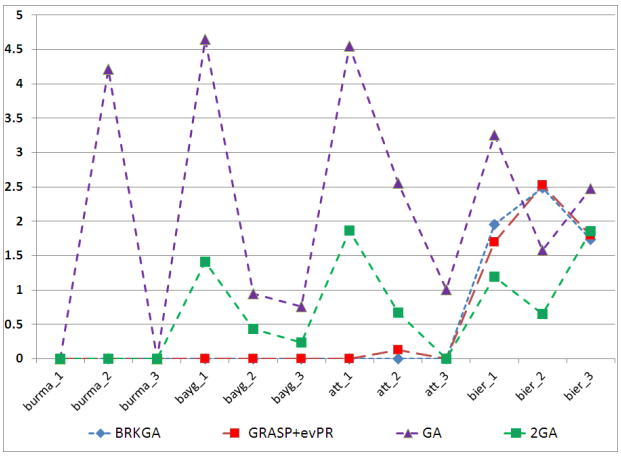

Figure 5: A comparison of the Relative Percentage Differences between the average solutions achieved by BRKGA, GRASP+evPR and GA and the average solutions obtained by 2GA

5 CONCLUSIONS

This paper considers the family traveling salesman problem. For solving this optimization problem we decompose it into two logical and natural smaller subproblems: a macro-level (global) subproblem and a micro-level (local) subproblem. We developed a classical genetic algorithm and hybrid diploid genetic algorithm. The first subproblem provides tours visiting the families using a classical genetic algorithm and a diploid genetic algorithm, while the second subproblem finds the minimum-cost tour, corresponding to the above mentioned tours, visiting a given number of nodes from each family. The second subproblem is solved by transforming each global tour into a traveling salesman problem (TSP) which then is optimally computed using the Concorde TSP solver. One important feature of our approach is the use of a diploid representation of the individuals that assures a higher diversity of the potential feasible solutions. The preliminary computational results show that our hybrid genetic algorithm is robust and compares favorably to existing approaches.

In the future, we plan to evaluate the generality and scalability of the proposed solution approach by testing it on larger instances and on instances derived from those used in the case of the generalized traveling salesman problem. It would also be promising to investigate the combination of our hybrid diploid genetic algorithm with a local search procedure.

REFERENCES

[1] R. Bernardino and A. Paias. 2017. Solving the family traveling salesman problem. *European Journal of Operational Research* (2017). https://doi.org/10.1016/j.ejor.2017.11.063

[2] J. A. Chisman. 1975. The clustered traveling salesman problem. *Computers & Operations Research* 2, 2 (1975), 115–119.

[3] C. Expósito-Izquierdo, A. Rossi, and M. Sevaux. 2016. A Two-Level solution approach to solve the Clustered Capacitated Vehicle Routing Problem. *Computers & Industrial Engineering* 91 (2016), 274–289.

[4] C. Feremans, M. Labbé, and G. Laporte. 2003. Generalized network design problems. *European Journal of Operational Research* 148, 1 (July 2003), 1–13.

[5] D.E. Goldberg and R.E. Smith. 1987. Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy. In *Proc. of Second International Conference on Genetic Algorithms and their application*. 59–68.

[6] B. Golden, Z. Naji-Azimi, S. Raghavan, M. Salari, and P. Toth. 2012. The Generalized Covering Salesman Problem. *INFORMS Journal on Computing* 24, 4 (2012), 534–553.

[7] A.L. Henry-Labordere. 1969. The record balancing problem: A dynamic programming solution of a generalized travelling salesman problem. *RIRO* (1969).

[8] M. Mitchell. 1998. *An introduction to genetic algorithms*. MIT Press.

[9] L.F. Morán-Mirabal, J.L. González-Velarde, and M.G.C. Resende. 2014. Randomized heuristics for the family traveling salesperson problem. *International Transactions in Operational Research* 21, 1 (2014), 41–57.

[10] Camelia-M. Pintea, Petrică C. Pop, and Camelia Chira. 2017. The generalized traveling salesman problem solved with ant algorithms. *Complex Adaptive Systems Modeling* 5, 1 (07 Aug 2017), 8. https://doi.org/10.1186/s40294-017-0048-9

[11] P.C. Pop, L. Fuksz, A. Horvat-Marc, and C. Sabo. 2018. A novel two-level optimization approach for clustered vehicle routing problem. *Computers & Industrial Engineering* 115 (2018), 304 – 318.

[12] P.C. Pop, O. Matei, and C. Sabo. 2017. A hybrid diploid genetic based algorithm for solving the generalized traveling salesman problem. In *Lecture Notes in Computer Science*, Vol. 10334. 149–160.

[13] P.C. Pop, O. Matei, C. Sabo, and A. Petrovan. 2018. A two-level solution approach for solving the generalized minimum spanning tree problem. *European Journal of Operational Research* 265, 2 (2018), 478–487.

[14] P. C. Pop. 2002. *The Generalized Minimum Spanning Tree Problem*. Twente University Press, the Netherlands.

[15] P. C. Pop. 2012. *Generalized Network Design Problems. Modeling and Optimization*. De Gruyter, Germany.

[16] Petrica C. Pop and Serban Iordache. 2011. A Hybrid Heuristic Approach for Solving the Generalized Traveling Salesman Problem. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*. ACM, New York, NY, USA, 481–488. https://doi.org/10.1145/2001576.2001643

[17] P. C. Pop, O. Matei, and C. Sabo. 2010. A New Approach for Solving the Generalized Traveling Salesman Problem. In *Hybrid Metaheuristics*, María J. Blesa, Christian Blum, Günther Raidl, Andrea Roli, and Michael Sampels (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 62–72.

[18] Jean-Yves Potvin and François Guertin. 1996. *The Clustered Traveling Salesman Problem: A Genetic Approach*. Springer US, Boston, MA, 619–631. https://doi.org/10.1007/978-1-4613-1361-8_37

[19] Mohammad H. Shaelaie, Majid Salari, and Zahra Naji-Azimi. 2014. The generalized covering traveling salesman problem. *Applied Soft Computing* 24 (2014), 867 – 878. https://doi.org/10.1016/j.asoc.2014.08.057

[20] S.S. Srivastava, S. Kumar, R.C. Garg, and P. Sen. 1969. *Generalized travelling salesman problem through n sets of nodes*. CORS.