Hybrid Evolutionary Algorithm for Travelling Thief Problem

NEERAJ PATHAK[1]
RAJEEV KUMAR[2]

[1,2]School of Computer & Systems Sciences
Jawaharlal Nehru University, New Delhi 110 067, India
[1]neerajpathak.jnu@gmail.com
[2]rajeevkumar.cse@gmail.com

Abstract. Travelling Thief Problem (TTP) is a benchmark problem, where two sub-problems, namely Travelling Salesman Problem (TSP), and $0/1$ Knapsack Problem (KP) interact. Since, the TSP and KP both comes under NP-hard, so the complexity to solve TTP will be decided by the investigation of inter-dependency factor. The objective of TTP problem is to suggest a picking plan with a best tour to a thief in such a way that the thief visits all cities exactly once with the collection of items in his knapsack leading to maximum benefits. With increasing size of TSP instance, a well-crafted search technique is required to maintain the inter-dependency. Here, we emphasis on the inter-dependency to design a solution framework which balance the interwoven between individual components. In this article, we suggest a hybrid Evolutionary Algorithm (EA) framework that strengthens the inter-dependency by giving best tour and picking plan concurrently. Here, we use a local search routine to improve the tour as well the objective function. Additionally, one more heuristic about KP component, is used for a better picking plan. On applying the proposed framework we have found that it gives most promising results compared to the classical EA approach.

Keywords: 0/1 Knapsack, Travelling Salesperson, Travelling Thief, Combinatorial Optimization, Constraint Satisfaction, Heuristics, Evolutionary Algorithm

(Received Sep. 24, 2020 / Accepted Nov. 15, 2020)

1 Introduction

Most real-world applications (RWAs) [32, 25, 30, 35, 13, 9] are usually composed of sub-problems with interwoven and inter-dependency. The Travelling Thief Problem (TTP) is one of them. The TTP problem comes under combinatorial optimization problem, where two most popular *NP-hard* problems, namely the Travelling Salesperson Problem (TSP) and the binary $(0/1)$ Knapsack Problem (KP), are interwoven. They are interrelated in such a way that solving individual components *alone* does not assurance to provide an optimal TTP solution.

Since, the TTP encompass two interdependent sub-problems, therefore, the algorithm designing of the multi-component real world problems can be investigated by the algorithm design of TTP that has the same behaviour. The systematic analysis of communication between these components insights the researcher to solve real world problems more satisfactory. There are various other examples of such TTP benchmar,k problem. The supply-chain management process consists of many components like distribution, transportation, loading, etc. that are complex individually but interwoven with each other. Similarly, routing problems with loading constraints, PCB (Printed Circuit Board) in VLSI design, airport ground movement problem etc, are some such sample examples belonging to the class of the TTP.

The TTP as a novel benchmark problem was introduced by Bonyadi et al. [6] in 2013. It is defined as: a thief having a knapsack (capacity W) has to visit all n

cities exactly once with the rented knapsack and return to its starting point provided a set of k items are distributed on these n cities. The speed of thief during the tour between cities become minimized according to the capacity of the knapsack as being filled-in. Therefore, the picking plan and the best tour both are important to contribute to the maximum benefits. In this way, the most important factor is 'inter-dependency' of the subproblems to solve this TTP problem.

In the past, many heuristics and metaheuristics [26, 28, 34, 33, 3, 31, 1] have been proposed to solve TTP. Initially, Polyakovskiy et al. [23] suggested three heuristics, namely Random Local Search (RLS) – a simple constructive heuristic, and (1+1)-Evolutionary Algorithm (EA). These methods solve the TTP problem in two steps, one assigned to TSP component and second one assigned for KP component problem. The same two phase approach was also given by Bonyadi et al. [7], that are: a Density- based Heuristic (DH), and a CoSolver (Coevolution based). Further to improve the results, Mei et al. [19, 18, 20] recommended a cooperative coevolution based method [2, 21, 8, 14, 36, 22, 29, 27] and a memetic algorithm (MATLS) for exploring inter-dependency. Most of the time, the research was mainly aimed at designing a picking plan considering a fixed tour. Similarly, Lin-Kernighan heuristic [15] generated tour, designed by a method given by Faulkner et al. [12] that involves optimizing the picking plan only. The Faulkner's study outperformed the then existing methods by giving a series of heuristics and operators (named as S1-S5 and C1-C6).

Moreover, Yafrani and Ahiod [10] proposed two heuristics, namely MA2B and CS2SA. The first one involves a fast local search heuristic, whereas the second heuristic consists of an adapted simulated annealing with 2-OPT steepest ascent hill climbing approach. For improving and speed up the local search heuristic, complexity reduction strategies are proposed to design an efficient Memetic Algorithm (MA). Some other metaheuristics approaches to solve TTP are: simulated annealing [11], genetic algorithm [5], swarm intelligence [37], tabu search [5], etc. However, due to *harder* nature of the problem, and stochastic nature of the solution strategies, the efficiency of all these approaches cannot be assessed even for smaller TTP instances. Therefore, there is need of a method/framework which may solve large size of TTP instances in reasonable amount of time giving most closer results.

In this article, we propose a hybrid Evolutionary Algorithm (EA) framework that combines the advantages of basic EA and Local Search (LS) strategies. Unlike, the existing methods which involve to design picking plan keeping the tour fixed, the tour varies here for a group of individuals. This way, the picking plan also affects the overall profits of the thief. In this context, we have applied 2-OPT heuristic as a local search to enhance the results. In addition to tour designing, there is also a need of effective picking plan about KP. So, we incorporate a bit-flip operator as a picking plan function.

The rest of the paper is organized as follows. Section 2 describes the problem description of TTP and its relation with two individual components, namely TSP and KP. Section 3 describes the methodology that involves the detailed description of our proposed hybrid EA framework. Section 4 includes the results of the proposed method to assess its efficacy. Finally, conclusion of this work is shown in Section 5.

2 Problem Formulation & Related Work

2.1 Travelling Salesperson Problem

TSP, a NP-hard problem involves to find the *shortest* route for a salesperson, such that every city/node is visited exactly once.

In this problem, we have a distance matrix $D_{ij} = d_{ij}$ for n cities where, d_{ij} is the distance between city i to city j. The salesperson has to visit each city exactly once in such a way that minimizes the time/resources of the complete tour. The tour basically is a permutation of n cities. If the speed of salesman is fixed to a constant (v_c), the objective function can be shown as:

$$f(\bar{x}) = \sum_{i=1}^{n-1} t_{x_i,x_{i+1}} + t_{x_n,x_1}, \bar{x} = (x_1, ..., x_n) \quad (1)$$

To minimize the objective function $f(\bar{x})$ $\bar{x}$,a permutation of city is required. The time travel between two cities can be calculated as:

$$t_{x_i,x_{i+1}} = \frac{d_{x_i,x_{i+1}}}{v_c} \quad (2)$$

2.2 $0/1$ Knapsack Problem

Another component of TTP is a 0/1 knapsack problem which also belongs to a NP-hard problem. In this context, a thief wants to gain maximal profit subject to the constraint of the knapsack's capacity. Given there are m items with their corresponding profits $\{p_i, ..., p_m\}$ and weights $\{w_i, ..., w_m\}$. A thief having a knapsack of capacity W wants to pick a subset of m items to gain maximum profit. It y_i is the decision variable whose value equating to 1 shows item i is chosen, otherwise 0. The problem can be formulated, mathematically, as:

$$\textit{Maximize } g(\bar{y}) = \sum_{i=1}^{m} p_i y_i, \bar{y} = (y_1, ...y_n) \quad (3)$$

subject to

$$w_i y_i \leq W \quad (4)$$

The $g(\bar{y})$ gives profit of the items picked by the thief. In this problem, the objective is find a permutation of items that makes maximal profit bounded by the capacity W of knapsack.

2.3 Travelling Thief Problem

Given a following details:

- n cities $N = \{1, ..., n\}$.
- m items distributed on these cities, i.e. $M_i = \{1, ..., m_i\}$ with m_i items at city *i*.
- p_{ik} and w_{ik} is profit and weight respectively, for each item *k* at city *i*.
- v_{min}, and v_{max} is minimum and maximum speed of thief respectively.
- R is a renting rate of thief who travels a distance d_{ij} between city i and city j.
- W is a knapsack capacity of thief.

The objective is to find a permutation of cities $T = \{x_1, ..., x_n\}$ and a bit string representing the picking plan $P = \{y_{21}, ..., y_{nm_n}\}$, such that the objective function $Z(T, P)$ to be maximized as follows:

$$Z(T, P) = \sum_{i=1}^{n} \sum_{k=1}^{m_i} p_{ik} y_{ik} - R \sum_{i=1}^{n} \frac{d_{x_i x_{s(i)}}}{V_i} \quad (5)$$

where, the city i has a successor $s(i)$ in the permutation T. The output Z is the summation of profits of the packed items minus the renting rate times the travel time. The second part of the equation shows travel time of the tour which depends on the distance between city x_i, its successor $x_s(i)$, and the speed V_i. The variable V_i is defined as:

$$V_i = v_{max} - \frac{v_{max} - v_{min}}{W} W_i \quad (6)$$

The speed V_i at city x_i is calculated with the collected weight W_{x_i} from items at city x_i. Without any item, the speed will be v_{max}, and with full capacity it will be v_{min}

$$W_i = \sum_{j=2}^{i} \sum_{k=1}^{m_j} y_{x_{jk}} . W_{x_{jk}} \quad (7)$$

At the last city x_n, the collected weight must be not more than the capacity W. Therefore,

$$W_{x_n} \leq W \quad (8)$$

Figure 1 demonstrates a brief example of the TTP, where each node except the node 1 has a set of items initialized with their corresponding profits and weights.

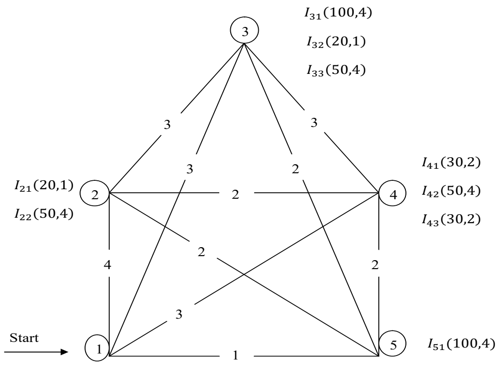

Figure 1: An example for a sample TTP

The capacity of the thief is considered as fixed number, say 5 (W=5), the v_{max} and v_{min} are fixed to 1 and 0.1 respectively with renting rate R=1. Let the optimal tour T={1,3,4,2,5,1} and the corresponding picking plan is P={1,0,0,0,0,0,0,0,1}. Thus, for travelling city 1 to city 2, no item is being picked giving total travel time (8/1)=8. The item I_{21} is picked from city 2. So, from city 2 the value of v_c=0.82 and W_c=1. The resulting travel time from city 2 to 5 is calculated as (2/0.82)=2.439. Similarly, at city 5, item I_{51} is picked so from city 5 v_c=0.1, W_c=5 giving travel time in between (city 5 to city 1) (1/0.1=10). Total travel time from city 1 to return back itself is counted as (10+2.439+8=20.439), and the profit is summed as (100+20=120). Hence, overall objective function $Z(T, P)$ is calculated as (120-20.439=99.561).

3 Methodology

There are various meta-heuristics in the literature to solve the individual components, TSP and KP both. Evolutionary Algorithm (EA), a population based is one of metaheuristics that is expected to give approximate solutions in reasonable amount of time [4]. In most of the cases, there is also a possibility of getting stuck in local optima. To overcome this problem, domain knowledge is required that could also improve the approximate solution of existing approach.

3.1 Proposed Framework

In this study, we apply the domain knowledge to devise an operator that retains the property of inter-dependency at each step. Unlike the implementing a classical EA on individual components, either TSP or KP, the TTP is treated differently. In the context of TTP problem, we emphasise on inter-dependency on each step to devise a better framework. Figure 2 shows the proposed framework, which includes local search heuristic to improve the objective for the thief. The detailed description of individual operators/functions is described in the following sub-sections.

3.2 Hybrid Evolutionary Operators

3.2.1 Initial population generator

Initial population is a group of individuals represented as a tour string associated with its corresponding picking plan P. The individual tour is generated using a well known nearest neighbor heuristic whereas, the picking plan P is generated using a scoring function s_c followed by a bit-flip operator.

Tour generating heuristic: The most common tour generating heuristic is a nearest neighbor (NN) one. It starts from randomly chosen vertex that will be the first vertex in the final tour. Now the distances to all the remaining vertices, are considered and the vertex with the smallest distance is picked and gets added to the tour. The same procedure is continued with a new added vertex. The algorithm is terminated when a tour containing all the vertices is created.

The NN Algorithm is given as follows:

Algorithm 1 Nearest Neighbor (NN)

Input: Distance matrix D_{ij}, vertex set X, start vertex s.
Output: resulting Tour $\bar{\pi}$
node x= s
node y
while $X \neq \phi$ **do**
$\bar{\pi}$.append(x)
X.delete(x)
pick $\exists y \in X$ such that D_{xy} is minimal
x=y
return $\bar{\pi}$
end while

The tour generated by the NN is shown in Figure 3. In this figure, the first row contains four strings of the tour, as output of the NN heuristic. These tours consider all the different starting vertex against the problem specifications, where the first (start) vertex of tour has no item. Therefore, without altering sequence of the tour, we make start vertex on the first position for the computing objective function. The second row of the four strings in Figure 3 shows the tour of the four individuals for computing the fitness value.

picking plan: The picking plan is a dependent factor of the knapsack problem. To find maximal profit by the thief, there is a need of a heuristic which gives some decision parameter(s). The scoring function is used as a heuristic to make decision for ranking of an item. Given, there are k items in city i, which item to be picked depends on the profit-weight ratio heuristic. The most common scoring function by Faulkner et al. [12] is given as:

$$s_{ik} = \frac{p_{ik}}{w_{ik}.d_i}, \tag{9}$$

where, d_i is the distance travelled from city i to the end of the tour. The scoring function s_{ik} incorporates the inter-dependency between the distance travelled with item I_{ik}, its profit and its weight. We have found in the latest study that scoring function could vary depending on inter-dependency To make the best objective value, a few studies have applied the exponent of profit and weight variable. One of the latest scoring function s_{ik} by Maity et al. [16] is shown as:

$$s_{ik} = \frac{p_{ik}^{\beta} t_{1,i}}{w_{ik}^{\beta} t_{i,1}} \tag{10}$$

where;

$$t_{1,i} = \frac{\sum d_{1,i}}{v_{max}}, t_{i,1} = \frac{\sum d_{i,1}}{V_i} \tag{11}$$

The $t_{1,i}$ is the time taken by the thief from starting vertex to the i, and $t_{i,1}$ is the travelling time from vertex i, where item is picked to the end of the tour. In study of Maity et al. [16], the authors have found the best objective value for $\beta = 7.4$ tested experimentally. In this study, we apply the same scoring function s_c to make the maximum profit. After generating score value, it is sorted in a non-decreasing order. This sorted array of the scores makes the decision to pick the items from a city.

Bit-flip operator: The generated picking plan depends on the scoring function (s_c) devised by the solver. There could be the possibility of improving this picking plan further. In order to optimize the picking plan, Faulkner et al. [12] proposed a bit-flip operator. This operator

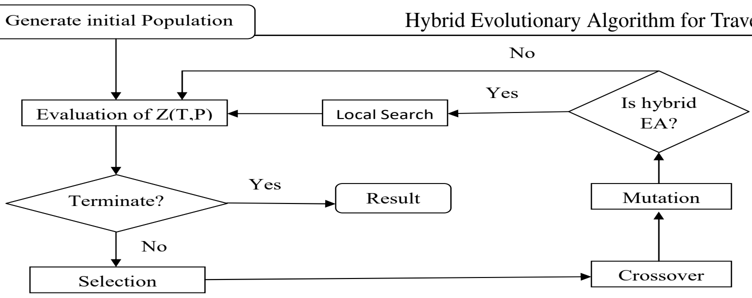

Figure 2: Hybrid EA framework for TTP

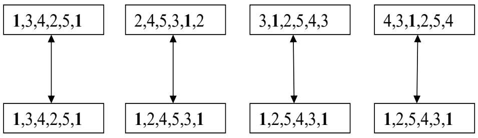

Figure 3: Results of the NN heuristic

involves flipping each bit of the bit strings. If we found improvement then keep the changes, otherwise, make it remains unchanged. The main idea behind this operation is to discard unprofitable items from the picking plan. The single operation of bit-flip operator is considered as attempting all bit-flips once. As discussed in example for Figure 1, the considered picking plan is (1,0,0,0,0,0,0,0,1). Figure 4 shows a binary picking plan P with single bit flip operation.

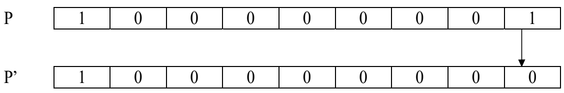

Figure 4: An example of a single bit flip

3.2.2 Evaluation

The evaluation (fitness evaluation) of an individual defines the worth of it comparing to other individuals. This is crucial and problem dependent. In the context of TTP, we have a group of individual tours and a picking plan P (sorted s_c), we can calculate the function $Z(T, P)$. The $Z(T, P)$ is calculated in the same manner as discussed in the example of Figure 1.

3.2.3 Selection

The goal of this operator is to emphasize the good solutions with the population size constant. It determines which individuals participate in reproduction to generate the next population depending on their fitness values of the current population.

Here, we apply the simplest operator (roulette wheel selection) with elitism. Elitism involves restriction of a set fittest individuals to go for the next generation. In the context of TTP, we considered 1% fittest of current population reserved and copied them in the next generation. The rest 99% undergoes through the roulette wheel selection.

3.2.4 Crossover

This operator is used to create new individuals from the existing mating pool after applying the selection operator. Here we apply two well known crossover operators of TSP, namely partially mapped crossover (PMX) and order crossover (OX) over TTP mating pool individuals.

In PMX first we consider partially contents of both the parents to create child sub-content. The C_1 and C_2 will be initialized as $(XXX|168|XX)$ and $(XXX|271|XX)$ respectively. In next step sign X in an individual string is replaced from the parent content considering if the value already there, remains unchanged. The updated tour string C_1 and C_2 are $(34X|168|X5)$ and $(4X5|271|36)$ respectively. We search the position of X in the child tour to its parent tour and follow the sequence in the parent from that position. If number in the following sequence is not in the child tour, put it in that position and repeat the step to replace all X. Finally, the output tour of C_1 and C_2 are $(342|168|75)$ and $(485|271|36)$ respectively.

In operator OX for generating a child C_1, we pick the P_2 sequence from second cut point $(3-7-4-2-5-1-6-8)$ and delete the middle segment (271) of P_1 from these sequence. The output generated sequence is $(3-4-5-6-8)$. Put these sequence in child C_1 from second cut point. The output result of C_1 tour sequence is $(568|271|34)$. Similarly, the output of C_2 is $(427|168|53)$.

3.2.5 Mutation

The mutation involves little bit change in the tour sequence of the individual within itself. It is based on the mutation rate. In this study, we apply two types of mutation, namely reciprocal exchange mutation and inversion mutation.

In reciprocal exchange, we choose a random vertex in the tour sequence and then choose a second random vertex, and then exchange the two. For instance, in an individual tour 5-6-8-2-7-1-3-4. If we end up choosing index 2 and index 6, then the mutated individual is: 5-6-3-2-7-1-8-4. Another operator, in inversion mutation, given a chromosome like: 5-6-8-2-7-1-3-4, we choose a mutation cut like index 2 to index 5 and then invert that sub-tour => 5-6-1-7-2-8-3-4.

3.2.6 Local Search

A local search operation needs a starting solution which gets replaced by the improved solutions found in the search space. In the context of TTP, the tour individual of population is replaced by the new tour individual which gives best fitness value. Here we emphasis on the best tour as well picking plan both to find profitable outcomes. We apply the 2-OPT heuristic as a local search routine to improve the $Z(T,P)$ value associated with an individual in the pool of population. If the value of $Z(T,P)$ increases then we consider the changes otherwise, individual remains unchanged.

3.2.7 Termination

It involves the termination condition of the algorithm execution. The algorithm can be terminated when the objective value $Z(T,P)$ converges or a fixed number of iterations is achieved. In this study, we set the termination condition the same, means when objective value does not improved, then it gets terminated.

4 Experimental Results

The experimental setup of this problem (TTP) is shown in first subsection. In the next subsection comparative results are also analyzed in details.

4.1 Experimental Setup and Data-sets

We used the TTP instance data for assessing the methods that were executed by Polyakovskiy et al. [23]. These are organized based on the instances of TSPLIB by Reinelt et al. [24] and of knapsacks configuration presented by Martello et al. [17]. We considered 10 base problems: eil51, eil76, kroA100, u159, ts225, a280, u574,u724, pr76,pr124. There are three types of KP component in TTP instances.

1. *Uncorrelated*: w_{ik} and p_{ik} are uniformly randomly distributed in the range $[1, 10^3]$ with high knapsack capacity (W=10),

2. *Uncorrelated with similar weights*: similarly, $w_{ik} \in [10^3, 10^3 + 10]$ and $p_{ik} \in [1, 10^3]$ with average knapsack capacity (W=5), and

3. *Bounded strongly correlated*: similarly, $w_{ik} \in [1, 10^3]$ and $p_{ik} = w_{ik} + 100$ with small knapsack capacity (W=1).

Additionally, the number of items per city (item factor) is $F \in \{1, 5, 10\}$. Note that all cities of a single TTP instance have the same number of items, except for the starting city, where no items are available. Here we consider type 1 (uncorrelated) KP component to execute all above mentioned TTP instances.

4.2 Results

We apply the proposed framework over TSPLIB instances with KP (uncorrelated type). The item factor for this execution is selected as 3. The parameters to execute the algorithm are as follows:
Population size: 100; Elite size: 25%; Mutation rate: 2%; Crossover rate: 98%

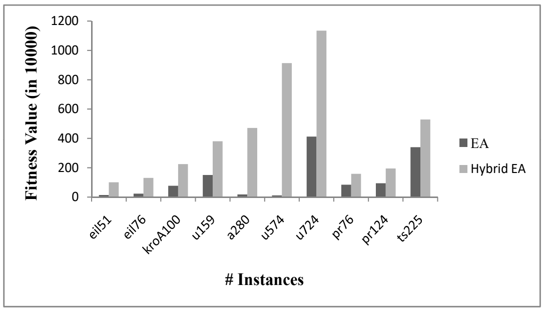

Figure 5: Comparison of EA vs. Hybrid EA

Figure 5 shows the comparison of Fitness value (objective value) applied over the Hybrid EA against the classical EA. It shows that for the instances 'a280' and 'u574', the Hybrid EA has more improvement over the classical EA. The result shows that the proposed framework surpasses the classical EA approach due to its tackling the picking plan and excuting the best tour concurrently.

5 Conclusion

The TTP problem composed of TSP and KP component, as a sample example, is for mapping the real world multi-component problems. The existing techniques of TSP as well KP have been used to design a model to emphasize the inter-dependency factor of the problems.

In this article, we have designed a Hybrid EA framework having the best picking plan associated with the best tour to give more benefits. We applied the framework over small sizes of TTP instances to assess the proof-of-concept, and compared the results with the existing approaches. We have found that the proposed framework gives superior results. Although, it gives better performance, but with the larger size of instances and several variants of the KP components, the result may not show similar trends. This is an area of future work, which we are focusing currently. Thus, the proposed framework has the possibility of solving large number of TTP instances with superior results.

References

[1] Affonso, E. T., Nunes, R. D., Rosa, R. L., Pivaro, G. F., and Rodríguez, D. Z. Speech quality assessment in wireless voip communication using deep belief network. *IEEE Access*, 6:77022–77032, 2018.

[2] Affonso, E. T., Rodríguez, D. Z., Rosa, R. L., Andrade, T., and Bressan, G. Voice quality assessment in mobile devices considering different fading models. In *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, pages 21–22. IEEE, 2016.

[3] Affonso, E. T., Rosa, R. L., and Rodríguez, D. Z. Speech quality assessment over lossy transmission channels using deep belief networks. *IEEE Signal Processing Letters*, 25(1):70–74, 2017.

[4] Ahuja, J. and Ratnoo, S. D. Feature selection using multi-objective genetic algorith m: A hybrid approach. *INFOCOMP Journal of Computer Science*, 14(1):26–37, 2015.

[5] Alharbi, S. T. A hybrid genetic algorithm with tabu search for optimization of the traveling thief problem. *Int. Journal Advanced Computer Science & Applications*, 9(11):276–287, 2018.

[6] Bonyadi, M. R., Michalewicz, Z., and Barone, L. The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. In *Proc. IEEE Congress Evolutionary Computation (CEC)*, pages 1037–1044, 2013.

[7] Bonyadi, M. R., Michalewicz, Z., Przybylek, M. R., and Wierzbicki, A. Socially inspired algorithms for the travelling thief problem. In *Proc. Conf. Genetic & Evolutionary Computation (GECCO)*, pages 421–428, 2014.

[8] de Almeida, F. L., Rosa, R. L., and Rodríguez, D. Z. Voice quality assessment in communication services using deep learning. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6. IEEE, 2018.

[9] de Almeida Junior, F., FranÃ§a, R., and Rodríguez, D. Image quality treatment to improve iris biometric systems. *INFOCOMP Journal of Computer Science*, 16:20–28, 2017.

[10] El Yafrani, M. and Ahiod, B. Population-based vs. single-solution heuristics for the travelling thief problem. In *Proc. Genetic & Evolutionary Computation Conf. (GECCO)*, pages 317–324, 2016.

[11] El Yafrani, M. and Ahiod, B. Efficiently solving the traveling thief problem using hill climbing and simulated annealing. *Information Sciences*, 432:231–244, 2018.

[12] Faulkner, H., Polyakovskiy, S., Schultz, T., and Wagner, M. Approximate approaches to the traveling thief problem. In *Proc Conf. Genetic & Evolutionary Computation (GECCO)*, pages 385–392, 2015.

[13] Guimarães, R., Rodríguez, D. Z., Rosa, R. L., and Bressan, G. Recommendation system using sentiment analysis considering the polarity of the adverb. In *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, pages 71–72, 2016.

[14] Jordane da Silva, M., Carrillo Melgarejo, D., Lopes Rosa, R., and Zegarra Rodríguez, D. Speech quality classifier model based on dbn that considers atmospheric phenomena. *Journal of Communications Software and Systems*, 16(1):75–84, 2020.

[15] Lin, S. and Kernighan, B. W. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

[16] Maity, A. and Das, S. Efficient hybrid local search heuristics for solving travelling thief problem. *Applied Soft Computing*, page 106284, 2020.

[17] Martello, S., Pisinger, D., and Toth, P. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.

[18] Mei, Y., Li, X., Salim, F., and Yao, X. Heuristic evolution with genetic programming for traveling thief problem. In *Proc. IEEE Congress Evolutionary Computation (CEC)*, pages 2753–2760. IEEE, 2015.

[19] Mei, Y., Li, X., and Yao, X. Improving efficiency of heuristics for the large scale traveling thief problem. In *Proc. Asia-Pacific Conf. Simulated Evolution & Learning*, pages 631–643. Springer, 2014.

[20] Mei, Y., Li, X., and Yao, X. On investigation of interdependence between sub-problems of the travelling thief problem. *Soft Computing*, 20(1):157–172, 2016.

[21] Nunes, R. D., Pereira, C. H., Rosa, R. L., and Rodríguez, D. Z. Real-time evaluation of speech quality in mobile communication services. In *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pages 389–390. IEEE, 2016.

[22] Nunes, R. D., Rosa, R. L., and Rodríguez, D. Z. Performance improvement of a non-intrusive voice quality metric in lossy networks. *IET Communications*, 13(20):3401–3408, 2019.

[23] Polyakovskiy, S., Bonyadi, M. R., Wagner, M., Michalewicz, Z., and Neumann, F. A comprehensive benchmark set and heuristics for the traveling thief problem. In *Proc. Conf. on Genetic & Evolutionary Computation (GECCO)*, pages 477–484. ACM, 2014.

[24] Reinhelt, G. {TSPLIB}: a library of sample instances for the tsp (and related problems) from various sources and of various types. *URL: http://comopt. ifi. uniheidelberg. de/software/TSPLIB95*, 2014.

[25] Rodríguez, D. Z., Abrahao, J., Begazo, D. C., Rosa, R. L., and Bressan, G. Quality metric to assess video streaming service over tcp considering temporal location of pauses. *IEEE Transactions on Consumer Electronics*, 58(3):985–992, 2012.

[26] Rodríguez, D. Z., Abrahao, J., Begazo, D. C., Rosa, R. L., and Bressan, G. Quality metric to assess video streaming service over tcp considering temporal location of pauses. *IEEE Transactions on Consumer Electronics*, 58(3):985–992, 2012.

[27] Rodríguez, D. Z. and Junior, L. C. B. Determining a non-intrusive voice quality model using machine learning and signal analysis in time. *INFOCOMP Journal of Computer Science*, 18(2), 2019.

[28] Rodríguez, D. Z., Rosa, R. L., Alfaia, E. C., Abrahão, J. I., and Bressan, G. Video quality metric for streaming service using dash standard. *IEEE Transactions on broadcasting*, 62(3):628–639, 2016.

[29] Rodríguez, D. Z., Rosa, R. L., Almeida, F. L., Mittag, G., and Möller, S. Speech quality assessment in wireless communications with mimo systems using a parametric model. *IEEE Access*, 7:35719–35730, 2019.

[30] Rodríguez, D. Z., Rosa, R. L., and Bressan, G. Video quality assessment in video streaming services considering user preference for video content. In *2014 IEEE International Conference on Consumer Electronics (ICCE)*, pages 570–571, 2014.

[31] Rodríguez, D. Z., Wang, Z., Rosa, R. L., and Bressan, G. The impact of video-quality-level switching on user quality of experience in dynamic adaptive streaming over http. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):1–15, 2014.

[32] Rosa, R. L., Rodríguez, D. Z., and Bressan, G. Music recommendation system based on user's sentiments extracted from social networks. *IEEE Transactions on Consumer Electronics*, 61(3):359–367, 2015.

[33] Rosa, R. L., Rodríguez, D. Z., and Bressan, G. Music recommendation system based on user's sentiments extracted from social networks. *IEEE Transactions on Consumer Electronics*, 61(3):359–367, 2015.

[34] Rosa, R. L., Schwartz, G. M., Ruggiero, W. V., and Rodríguez, D. Z. A knowledge-based recommendation system that includes sentiment analysis and deep learning. *IEEE Transactions on Industrial Informatics*, 15(4):2124–2135, 2018.

[35] Rosa, R. L., Schwartz, G. M., Ruggiero, W. V., and Rodríguez, D. Z. A knowledge-based recommendation system that includes sentiment analysis and deep learning. *IEEE Transactions on Industrial Informatics*, 15(4):2124–2135, 2019.

[36] Vieira, S. T., Rosa, R. L., and Rodríguez, D. Z. A speech quality classifier based on tree-cnn algorithm that considers network degradations. *Journal of Communications Software and Systems*, 16(2):180–187, 2020.

[37] Wagner, M. Stealing items more efficiently with ants: a swarm intelligence approach to the travelling thief problem. In *Int. Conf Swarm Intelligence*, pages 273–281. Springer, 2016.