Travelling Holidayman Problem

SEPTEMBER 2021

AUTHOR(S):
Lunara Nurgaliyeva

Nazarbayev University, Kazakhstan

SUPERVISOR(S):
Kristiane Novotny, gluoNNet
Daniel Dobos, gluoNNet & CERN
Cenk Tüysüz, DESY
Karolos Potamianos, University of Oxford, gluoNNet & CERN

CERN openlab

ABSTRACT

The Travelling Salesman Problem is a well-known problem where it is required to figure out the shortest route to visit all the stops. Even though it was formulated almost over a hundred years ago, nowadays it still has relevance and interest. Therefore, the attention throughout this paper was focused on the logistic computing relevant modification of the Travelling Salesman Problem called Travelling Holidayman Problem, which can be applicable in exploring and solving the current challenges of logistics. The problem of Travelling Holidayman is solved based on the greedy algorithm solver of the Travelling Salesman Problem. Even though the greedy algorithm shows the best performance in terms of time, it does not guarantee the optimality of the solution, that is to find the shortest path. Therefore, other approaches such as quantum computing should be considered in solving this problem.

TABLE OF CONTENTS

1. INTRODUCTION

The Travelling Salesman Problem is roughly a hundred years old mathematical as well as information technology problem. The main idea of this problem is that a traveler has to visit all the available cities only once in order to obtain the shortest route. Despite the fact that it sounds like a trivial task, the solution is very complex. The model of the Travelling Salesman Problem has been modified to explore and solve current challenges in the field of logistics. This modification is called the "Travelling Holidayman Problem". The basic idea of this modified model is that the traveler wants to spend the entire year traveling on bank holidays, which means that every time the traveler travels to the country, there is a bank holiday or weekend. The Travelling Holidayman Problem follows almost the same rules of the Travelling Salesman Problem, except in case of the time dependence of the Travelling Holidayman Problem. Since it is impossible to go back in time, travelers can travel each consecutive day. The main objective of this project is to find a suitable algorithm to solve the problem of Travelling Holidayman.

2. DATASET

The dataset used in the current work was scraped from the web (see Figure 1) [1]. It contains bank holidays of 232 countries in the year 2021. Subsequently, based on the weekdays of each country, country dependent weekends were added [2]. The number of total holidays and weekends per date can be seen from the following graph (see Figure 2).

	country	date	name
0	United States	2021-01-01	New Year's Day
1	United States	2021-01-18	Martin Luther King Jr. Day
2	United States	2021-01-20	Inauguration Day (DC, MD*, VA*)
3	United States	2021-02-14	Valentine's Day
4	United States	2021-02-15	Presidents' Day

Figure 1. *Dataset*

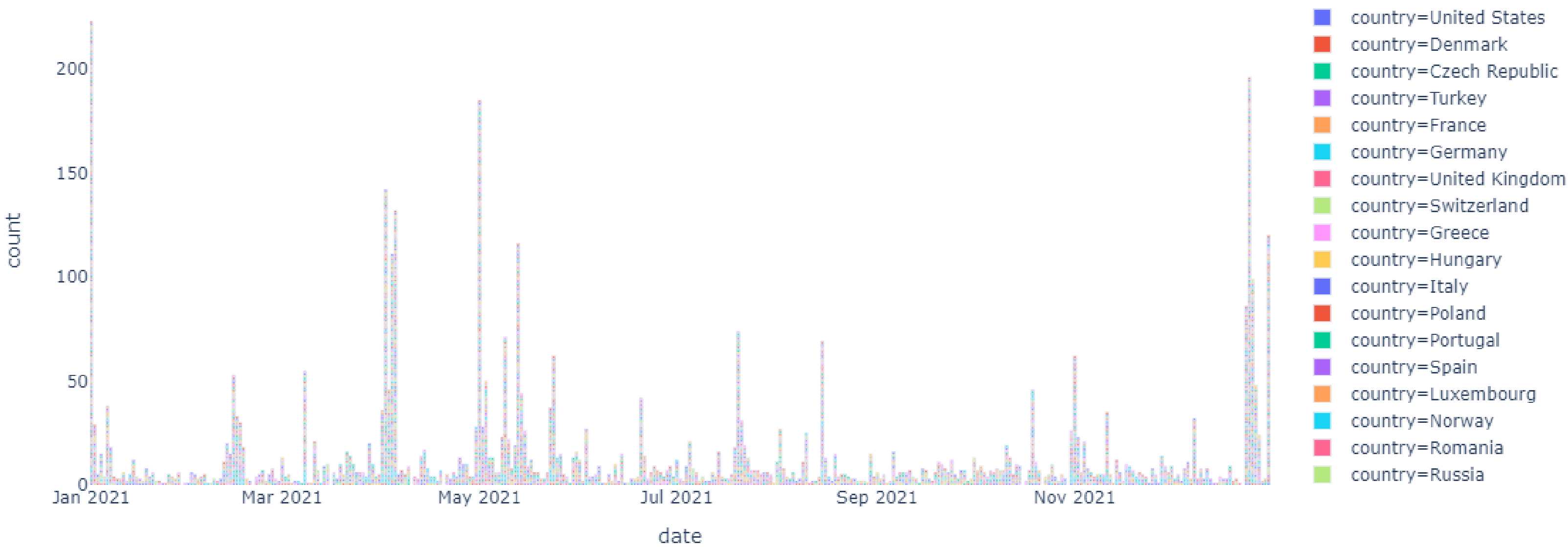

Figure 2. *Stacked bar plot with the number holidays and weekends of each each date*

The figure shows that almost all days are considered a holiday or weekend, except for two days, specifically '2021-03-16' and '2021-04-20'. These are considered as business days in all 232 countries. Geographic locations, names of airports of each country, the information whether the day is a holiday or weekend in a specific country as well as the number of countries per each date have been added (see Figure 3).

	country	date	name	sum of country	holiday	weekend	weekday	Capital Name	name_x	latitude_deg	longitude_deg	name_y	unique_countries	unique_airports
0	Afghanistan	2021-01-01	Weekend	223	False	True	Friday	Kabul	Khost_International_Airport_(U.C.)	33.284605	69.807340	Afghanistan	Afghanistan_2021-01-01	Afghanistan_2021-01-01_Khost_International_Air...
30719	North Macedonia	2021-01-01	New Year's Day	223	True	False	Friday	Skopje	Skopje_International_Airport	41.961601	21.621401	North Macedonia	North Macedonia_2021-01-01	North Macedonia_2021-01-01_Skopje_Internationa...
30590	North Korea	2021-01-01	New Year	223	True	False	Friday	Pyongyang	Haeju_Airfield	38.007390	125.777321	North Korea	North Korea_2021-01-01	North Korea_2021-01-01_Haeju_Airfield
30589	North Korea	2021-01-01	New Year	223	True	False	Friday	Pyongyang	Wonsan_Kalma_International_Airport	39.166801	127.486000	North Korea	North Korea_2021-01-01	North Korea_2021-01-01_Wonsan_Kalma_Internatio...
30350	Nigeria	2021-01-01	New Year's Day	223	True	False	Friday	Abuja	Nnamdi_Azikiwe_International_Airport	9.006790	7.263170	Nigeria	Nigeria_2021-01-01	Nigeria_2021-01-01_Nnamdi_Azikiwe_Internationa...

Figure 3. *Modified Dataset*

3. ALGORITHM

For solving the problem of Travelling Holidayman open-source repository of suboptimal Travelling Salesman Problem (TSP) solver was used, which provides a pure Python code for searching suboptimal solutions to the TSP based on the greedy algorithm (See Figure) [3, 4]. Despite the fact that a greedy algorithm does not guarantee the optimality of the solution, it might output a short path but not the shortest, it takes minimum time for computation. Since the Travelling Salesman Problem is one of the NP-hard problems, several attempts have been made to solve it by different algorithms. According to the 'Comparison of TSP Algorithms" paper, the greedy algorithm showed the best performance in terms of time (See Table 1) [5].

The algorithm of the solver had to be modified since the problem of Travelling Holidayman is time relevant. This means, e.g., every time a traveller chooses the next country to travel, he chooses the one which is closest by both date and distance. This algorithm was chosen to generate test and validation datasets.

Algorithm 1 Greedy algorithm

procedure GREEDY($start\_pos, C$)
    $current\_pos \leftarrow start\_pos$
    $V \leftarrow \{\}$
    while $|V| < |C|$ **do**
        $next\_pos \leftarrow \min_c dist(current\_pos, c), \forall c \in C \setminus V$
        $V \leftarrow V \cup next\_pos$
        $current\_pos \leftarrow next\_pos$
    end while
end procedure

Figure 4. *Pseudo code of greedy algorithm* [4]

Algorithm \ City Size	30	100	500
Greedy	0	0	1
Greedy 2-OPT	0	0	2
2-OPT	0	0	14
Greedy 3-OPT	0	11	
3-OPT	0	30	
Simulated Annealing	2	6	27
Genetic Algorithm	6	8	34
Neural Network	1	12	346

Table 1. *Time comparison of different algorithms (in sec)* [5]

4. RESULTS

The final route with the total distance of 542487.81km that was obtained using the solver algorithm can be seen from the following figures (see Figure 5 and Figure 6). Since the traveler is allowed to visit the country more than one time, the number of visits can be larger than one. This number of visits can be seen from the color of each country (see Figure 6). Another possibility to visualise this fact can be seen by the color of the paths that were used by the traveler (see Figure 5).

As was mentioned above, a greedy algorithm does not guarantee that the final route would be the shortest path, since it does not consider all possible combinations. From the following table, it can be observed that the number of possible combinations increases by adding airports for each country. The computation time increases also significantly, which means that the computation of all possible combinations takes even more time (see Table 2). Therefore, the solution for this problem can be considered as an application of Quantum Computing, specifically to a Quantum Graph Neural Network. There is already an existing quantum graph neural network approach for the similar problem of particle tracking [6], which can be used to solve the Travelling Holidayman Problem. For example, the Quantum Edge Network and Quantum Node Network that were used in the Quantum Graph Neural Network approach of the particle tracking problem can be also applied in Travelling Holidayman Problem.

Figure 5. *The plot of the final route with the number of visits per path*

	One airport per country (4.5%)	Two airports per country (8.2%)	Three airports per country (11.3%)
All possible combinations	1.5E+492	3.7E+589	9.4E+642
Time for computation with greedy algorithm	~60s	~4500s	~8100s

Table 2. *Computation time of greedy algorithm for different number of combinations*

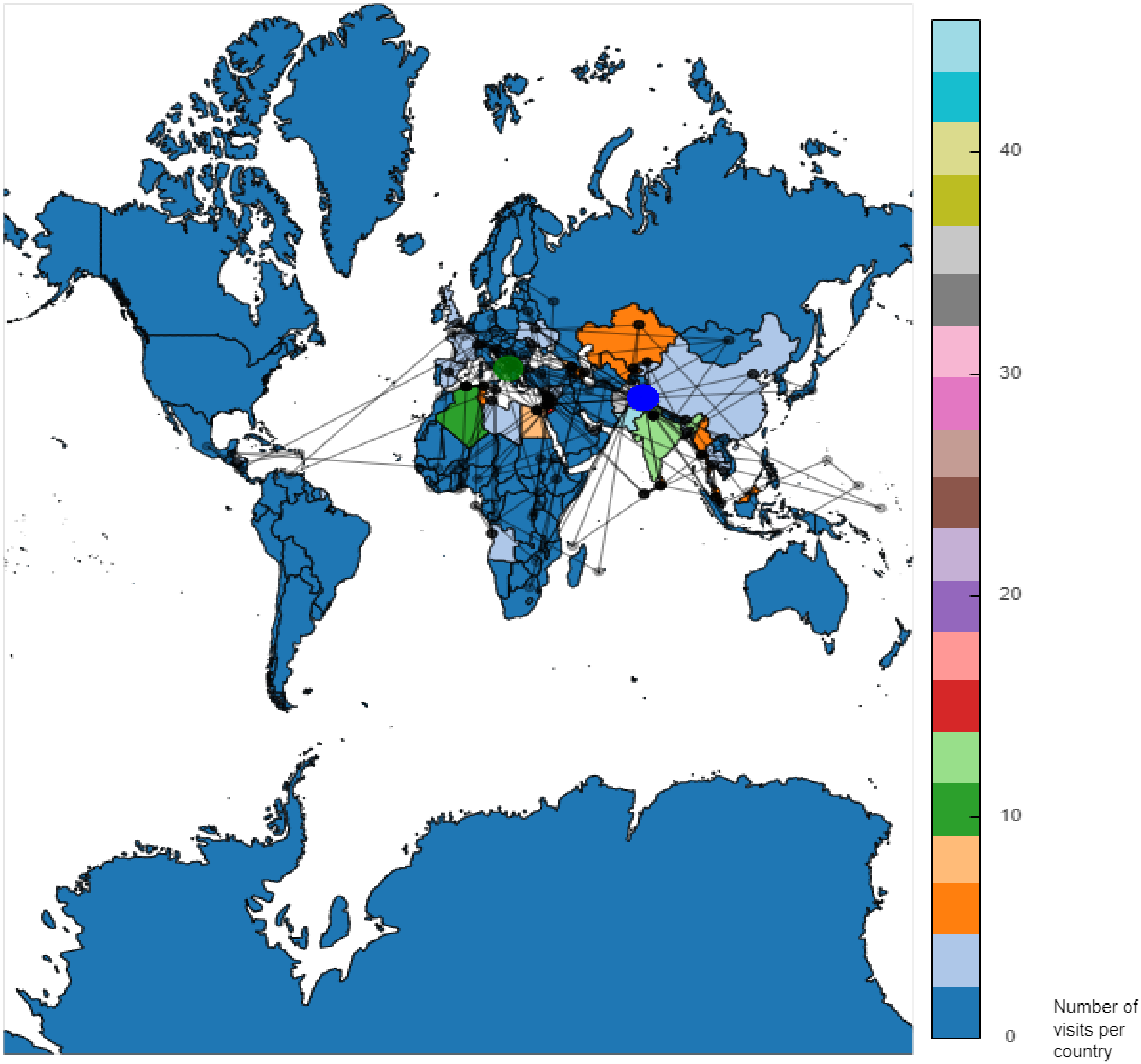

Figure 6. *The plot of the final route with the number of visits per country*

5. CONCLUSION AND FUTURE WORK

Overall, taking all things into consideration it can be concluded that this report presents the holiday relevant modification of the well-known Travelling Salesman Problem which was solved using the greedy algorithm. Since the greedy algorithm does not always output the optimal solution, there is still a lot of room left for progress and plenty of material for future work with the another approach of Quantum Graph Neural Network, starting with the preparing of input (graph) for the Graph Neural Network.

6. REFERENCES

[1]. “Holidays API”. 2021. https://www.timeanddate.com/.

[2]. “Workweek and weekend”. 2021. https://en.wikipedia.org/wiki/Workweek_and_weekend

[3]. Dmitry. 2020. “Suboptimal Travelling Salesman Problem (TSP) solver”.

https://github.com/dmishin/tsp-solver

[4]. Gajduk, A. 2016. “Effectiveness of greedily collecting items in open world games”.

https://github.com/gajduk/greedy-tsp

[5]. Kim, B., Shim J., Zhang, M. 1998. “Comparison of TSP Algorithms”.

[6]. Tüysüz, C. 2020. “Particle Track Reconstruction with Quantum Algorithms”.

https://github.com/cnktysz/HepTrkX-quantum/tree/ff6638fc2883781245f5e6f0d875a5fef1dc90b6/data