

THE TRAVELLING SALESMAN PROBLEM WITH MATLAB

Mastering complex algorithms and their programming realization became an important part of modern education on each engineering specialties. Instead of learning particular programming languages, we suggest applying mathematical platforms like Wolfram Mathematica, Maple, MATLAB etc. that provide ready tools for basic mathematical problems. We accept the last one, the MATLAB, as one of the most appropriate for students in engineering. It is to background in this paper that such a difficult problem as Traveling Salesman Problem (TSP) may be mastered by students.

Mathematical formulation of the Traveling Salesman Problem (TSP) sounds in the following way: a “salesman” from the city 0 should visit all the other n cities named as 1, 2, . . . , n only once and come back to 0; among $N=n!$ possible routes the shortest is to be found. Despite the simplicity of the formulation and the significant “age” of the problem, it turned to be one of most complicated in the discrete mathematics and optimization theory up to nowadays.

Direct solution of the TSP lies in the “honest” search of all the possible routes, accurate calculation their length and comparing them until all the variants are examined. With the grows of n , the number of cities, it turns however that the time, required for enumeration of all possible routs, grows drastically so that the solution cannot be achieved during the reasonable time period. That is why this “naive” way was called “brute-force algorithm”, and efforts were paid into invention of more heuristic but realistic algorithms. Several such algorithms were elaborated, that found their recent applications in effective popular automotive GPS navigators. However, the TSP still remains unsolved “in full”. The problem’s state-of-the-art may be found in Wikipedia.

The Traveling Salesman Problem has already been realized in MATLAB. One should simply issue travel in the Command Window. Graphical User Interface (GUI) that appears is shown in the Figure 1A along with solution for $n=50$ cities on American map. We’d like to develop a similar program but oriented to our country Ukraine rather than to USA.

Graphical appearance of our program called SalesMan.m has been shown in the Figure 1,B. It uses the map of Ukraine. Issuing SalesMan in the MATLAB Command Window leads to another GUI shown. The button “Help” provides a short explanation window 2a on the right top. One could choose the number of cities, n , in the narrow List Box window 3. When chosen, the button “Generate cities” is to be pressed, and corresponded number of Ukrainian cities becomes labeled as green points, see Fig. 1B for $n=20$. The button “Optimize distances” 4b starts process of possible routes enumeration, and each route under consideration may be observed by user. At the end, the final route with the least length is demonstrated. As told, we “honestly” enumerate all the possible routes what is called the “Brute-force Method”.

When the final programming product has been highlighted in such a way, we can start explanation how to go to such result. There are few relative simple subprograms among them, but there are also some especially important ones that form an educational and intellectual background of the profession of programming.

First, one needs to create Graphical User Interface (GUI) of the program where all its functionality should be foreseen by means of windows for inputting and outputting information and by buttons that start execution of certain tasks. One of latter is a procedure how to choose particular cities. Secondly, a subprogram should be elaborated that enumerates all possible routes between the cities chosen (the brute-force method). Thirdly, all the routes generated are to be “passed by the program” and the length of them calculated. It would be visual to demonstrate on the map any route currently in consideration. Finally, the route with the least length is to be returned as the final problem solution for the city collection chosen.

Many of algorithms and subprograms required here may devote special attention. Generation of permutations is particularly difficult to students and is based on the recursion what is one of key stone algorithms in Computer Science.

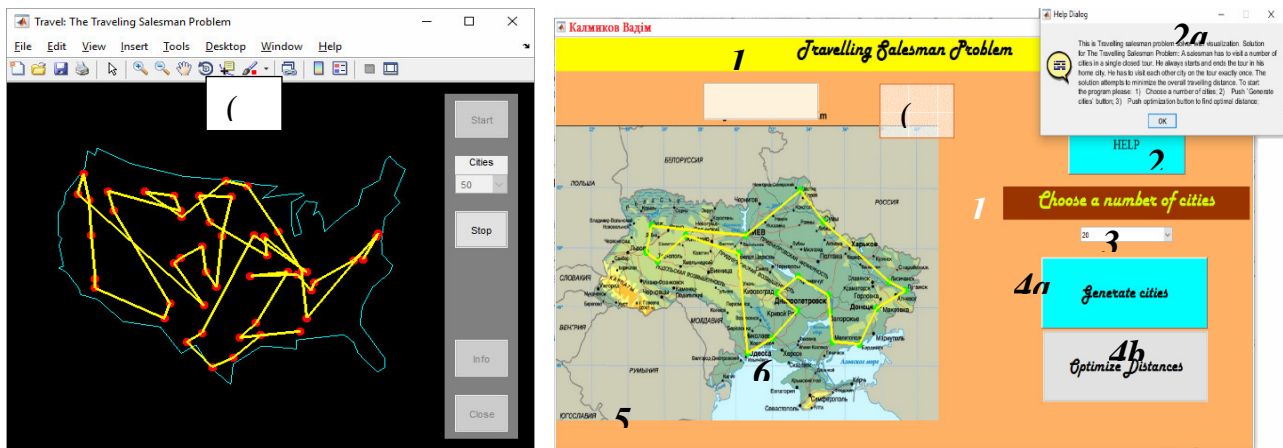


Figure 1. (A) TSP GUI in MATLAB; (B) GUI of our MATLAB-program with the map of Ukraine.

Say, we need to get all permutations of numbers $1, 2, \dots, n$, what means that function $T = \text{Permutations}(n)$ depends on n . In the simplest case $n=1$ the list of permutations T contains only 1. If $n=2$ the Permutation-program returns T as a matrix $[1 \ 2]$. Similar, $\text{Permutation}(3) = [1 \ 2 \ 3; 1 \ 3 \ 2; 3 \ 1 \ 2]$, i.e. new element 3 situates within all previous matrix rows. (We do not account for reverse sequences of numbers as corresponded routes have the same length). Full text of the program will be reported in presentation and published in article.

As the result, students have a pleasant up-to-date graphical program SalesMan that realizes the real logistics problem. It is worth to provide its wide investigation, and particularly the time it consumes. They estimate that the Time is proportional to $n!$, i.e. it grows drastically huge. It is to motivate them to look for other algorithms able to reduce the Time significantly. A number of heuristic approaches were suggested in five last decades but the TSP remains far from complete solution.

So, such a complex problem as TSP is been in strength to students provided the recent mathematical and programming environment MATLAB is applied.