Radovan SOUŠEK*
Zdeněk DVOŘÁK**
Tomáš FIEDLER***

COMPUTER SUPPORT OF TRANSPORTATION OF HEAVY AND OVERSIZE SHIPMENTS

The paper deals with transportation of heavy and oversize shipments on the part of algorithm development and software solution.

1. PREFACE

Planning of transportation of heavy and oversize shipments is one of the most crucial problems for road transport. Because there is no specialized software which will be able to solve these problems, we have decided to produce such software. Therefore we created program FadRo, which offers computer support for searching for ideal routes. It is designed to support transportation of heavy and oversize shipments, but it can be used also in different areas. In process of programming this software we had to use different algorithm from applied mathematics.

2. THEORETICAL SOLUTION

2.1. ALGORITHM USED FOR SEARCHING THE IDEAL ROUTE IN SOLVING PROBLEMS WHICH FOLLOW A HUGE NUMBER OF CRITERIA.

At the begging we had to focus on some important terms which are important in searching for optimal route in solving problems which follow a huge number of criteria.

In order to solve the problem of optimal route we were given chart with some vertexes and edges. Each edge has given certain value, which can be defined as certain distance, maximal weight and etc. If the solution is limited only by charts operating with negative value of edges, which in

* Dr inż. Radovan SOUŠEK, Uniwersytet Pardubicki, Czechy.
** Dr inż. Zdeněk DVOŘÁK, Uniwersytet w Żylinie, Słowacja.
*** Dr inż. Tomáš FIEDLER, Uniwersytet Pardubicki, Czechy.

fact is not very difficult to solve, we can use very effectively one of the variations of Dijkstr algorithm. These algorithms are well described and they are a part of basic chart theory. Particular ways of their usage and their description can be found in literature which focuses on theory of charts. The problem is how to take more than one criteria in consideration. For evaluating individual options we can use algorithms which follow a huge number of criteria. In final version of software we used dominance, ideal options and function of benefit. For particular solution we used up edited Dijkstr algorithm. Edges of chart do not reflect just one attribute, but a great number of attributes (number of attributes is variable and dependent on number of criteria). Vertexes which play a role of medium of information about shortest route from initial vertex in Dijkstr algorithm now carry data about all not dominant options of route from initial vertex and also information about attributes of all edges between them. Algorithm is trying to find the best possible option until it is sure that there is no better option.

2.2. FORM OF ALGORITHM

Steps in proceeding algorithm:

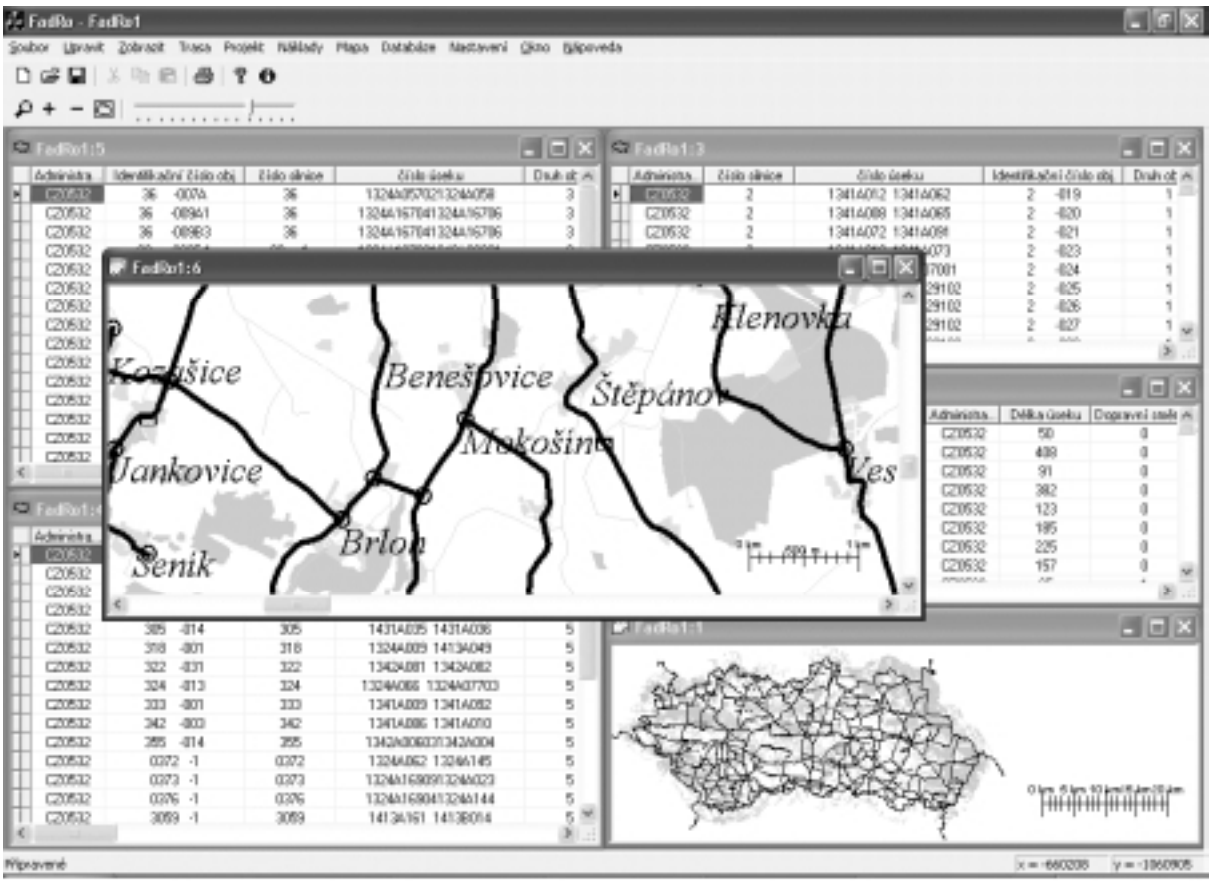
1. The information about optimal option is saved.
2. Value of initial vertex is calculated according to following rule. Current distance shows value 0 and other attributes are laid into axis close to the ideal option.
3. This vertex is added to a set of all vertexes which have to be investigated.
4. We have to consider that this step will be repeated. The algorithm checks whether the set FRONTA is empty, and if yes work algorithm is finished.
5. We take the vertex with the lowest value from the set FRONTA. We label this vertex as current one.
6. We have to find out whether current vertex does not exceed maximal chosen distance. If yes algorithm is finished.
7. Algorithm checks whether current vertex is the final vertex. If yes, their attributes are compared to initially calculate ideal option. If this solution is close enough to the ideal option, the proceeding is finished. If the solution is not good enough, but it is sure that it is final vertex, algorithm goes back to step number 4.
8. Following steps will be processed for each vertex adjacent to current vertex. The thing is to find out if the set of adjacent vertexes is empty. If yes algorithm goes back to step number 4.
9. The adjacent vertex is chosen and the task of algorithm is to find out if vertex is located on a route. If yes, the selection is repeated.
10. New route is added to the vertex. Route is added in a way that attributes are rewritten from current vertex and they are changed according to the attributes of edge between current and adjacent vertexes (the distance is added, number of bridges is added, width is equal to lower value, etc.). The adjacent vertex is considered as the final vertex in route.
11. Algorithm checks whether all criteria are fulfilled. If there is no route which fulfills criteria algorithm goes back to step number 7.
12. New route is compared to existing one in a way that they are considered as pair. If algorithm found out that there is pair of routes in which one route is less effective (its attributes are identical with different with other route or it is even much worse than others), this route is removed from the sets of routes which contribute to this vertex.
13. If the route created in step 9 was not removed, adjacent vertex and information about this route would be added to the set FRONTA and of course to the set of routes. The step number 7 is repeated.

14. The end of algorithm. After finalizing of algorithm three possible situations can take place:
    a) No route was found,
    b) One route was found,
    c) More than one route was found.

In the first case we can say that there is no route between initial and final vertex. In second case there is only one solution and in the third case we have to calculate benefits of each route and than to arrange them according to their values.

3. TECHNICAL DETAILS OF SOLUTION

Fig. 1 Operational environment of software



Sources: author's own elaboration.

3.1. MAP ENVIRONMENT

This software shows everything what is located in database. For instance: bridges, gate-ways, underpasses, forests, buildings, railway tracks, water coverage, contours etc.

One of the basic characteristics of this software is that you can choose from default settings of displaying, edit them or to design your own. This software is able to focus on such details as displaying roads with length higher than 2 km and show their width in map or even to display bridges which were built between years 1980 and 2000.

3.2. DATA PROCESSING

This software works with map basis such as ERSI Shapefile and it is compatible with data from dBASE, FoxPro or Clipper. To add new database you have just to add new database with its shapefile to sources and than it is able to work with it. This software is also compatible with road databank and it has no problem in integrating this data.

You can easily add new criteria for maximal size of shipment and process it according to new maps and data. You can also add your own notes, new photo documentation and of course changes pass ported data. Other very important part of software is interaction between maps and databases. You can find out all the needed information about object just after one click. It is the same in case of certain passage of route. If we consider the outcome of passports, particular obstacles and so on, we have very powerful tool for quick orientation in the route.

3.3. SUPPORT OF OVERSIZED AND HEAVY TRANSPORTS

This software can be divided into two parts, passive and active. Passive is useful for work with data about road network and active is used for finding out the ideal route. In order to find out the ideal route the software uses special algorithm which was mentioned above. For individual user it means possibility how to set many criteria. For instance: exceeding of size of shipment, limiting of certain obstacles, choosing between ways of finding out the optimal route or to give different value to each criteria. Particular settings of criteria can be loaded from previously saved profile. When the ideal route was found it is very easy to label certain parts of route or to list all possible obstacles.

4. EPILOGUE

The main factor, which influences the possibility to use this software, is maximal speed, and pack of information needed to provide route of transportation and also how user is able to define other criteria for transportation according to existing obstacles.

This article was created thanks to support of Institutional research "Theory of transport systems" (MSM 0021627505) University in Pardubice.