

## NEW VERSION OF THE BBS METHOD AND ITS USAGE FOR DETERMINING AND SCHEDULING VEHICLE ROUTES

Andrzej Wolfenburg

Institute of Logistics and Warehousing, Poznan, Poland  
e-mail: [andrzej.wolfenburg@ilim.poznan.pl](mailto:andrzej.wolfenburg@ilim.poznan.pl)

---

**Abstract:** *The work presents a method of the vehicle routing and scheduling using the modified Branch and Bound Simulation method for vehicles simultaneously carrying loads from multiple orders. Limitations concerning vehicle loads and time windows in pickup and delivery points are taken into consideration. The developed algorithm appoints the shortest and the cheapest routes with a certain known accuracy. A modification of the BBS method is described. A sample of result using Google API is shown.*

**Key words:** *vehicle routing and scheduling, Branch and Bound Simulation, BBS*

---

### 1. Introduction

Institute of Logistics and Warehousing and CallFreedom company have developed a system T-Traco for vehicle tracking and communication with drivers. It is an innovative solution for global monitoring of any means of transport, cargo containers, and communication with the drivers with total bypassing high roaming charges when using mobile terminals and driver smartphones. The basis of the system is a proprietary communication system with its own SIM cards allowing data transmission and the use the connections of leading mobile operators around the world. The system allows a full track & trace with the possibility of real-time control of temperature and humidity means of transport and containers, as well as communication with drivers, track status reports with a system of alerts, archive of routes with all the events. It is possible to monitor the basic parameters of the vehicle (eg. Velocity, length of routes), and documenting the events of transport (image, video recording). Positioning the vehicle on a map is carried out at intervals of 15-20 seconds. T-Traco is intended for use by global logistics operators, small and medium transport companies, manufacturing and distribution companies interested in monitoring transportation means and freight.

The described solution is an efficient tool for monitoring of vehicles and communion with drivers. The next step in the development of applications is complemented it by automatic functions of planning the allocation of orders to vehicles and functions determining and scheduling

vehicle routes. The purpose of this paper is to present the second issue.

### 2. Description of the problem

The problem to be solved is to optimize routes of freight vehicles. The total time of transport carried by all vehicles is minimized. Instead of the total transport time we can minimize the number of kilometers traveled or the cost of transport. Transportation occur between the desired pick-up points and delivery points, and a vehicle after the completion of the last scheduled order is waiting in the last delivery point (or nearby) for an information to the driver about a new order displayed on the mobile device. This information will be complemented by the coordinator with further guidelines for the order.

The vehicles can carry freight from several orders within their permissible load, and some orders may be executed using not a shortest connection, which is caused by the optimization criterion. Some goods cannot be transported at the same time e.g. chemicals and foods. The condition for not exceeding the carrying capacity of vehicles during the transportation planning process is tested due to several parameters. This may be weight, volume, number of pallets and the number of containers with regard to their sizes. For loading and unloading processes there are given time windows for each shipping point that the vehicle must meet. Also, the physical parameters of the vehicle as the weight and size must comply with the acceptable parameters for a given shipping point. Additionally, they can be taken into account limitations of standard working time of drivers.

It was assumed that the plan execution is done by the coordination center in two sessions daily: morning and afternoon. During morning session it is planned the execution of orders received for pick-up taking place on a given day, while the delivery will take place on the desired time. In the afternoon session are planned orders, which will be picked up this day or up to the next day noon, and delivery occurs on the desired time. At the time of planning it is taken into account the expected time and location of the vehicle after the last previously scheduled delivery and the current position of the vehicle to calculate the possible delay of order execution. The orders before passing them to the execution should be approved by the customer for the price of transportation. The coordinator determines the prices based on cost calculated by the system. Orders whose exercise price was not accepted by the customer are excluded from the process and planning process for rest of the orders and vehicles have to be repeated. The accepted orders are forwarded to the execution by sending them to the drivers of vehicles using the T-Traco system.

### 3. The BBS method

The BBS (Branch and Bound Simulation) method was developed by the author for solving optimally the dispatching problem for trains in the railway network [14] minimizing the total weighted delays of trains at the end of the time horizon. It was created by combining the discrete simulation with the B&B method. In the simulation, there is one discrete event list present, in the B&B method there are multiple lists of events, each of which is contained in a vertex of the Branch & Bound method graph. Each list of events is a variant of partial solutions.

On the list of events of the particular variant there are train events ordered by decreasing values of their earliest event time. As in method B & B the tree is developed until reaching the optimum solution. The selection criterion for the vertex to be further developed is the smallest value of the lower bound General Measure of Effectiveness (LBGME) which is assigned to the vertex. The GME is a weighted sum of delays of all the trains within particular variant calculated for the moment of the end of a prediction horizon (1-2 hours). The weights depends on categories of trains. The

selected node becomes the parent of such number of new vertices as many acceptable controls exist for a first train from the event list of the variant. The optimal solution is achieved if the simulation time for a train event of a concerned variant exceeds the time of prediction horizon and LBGME value for this variant is not greater than LBGME for each other variants [14] previously created. The BBS method was also used to make optimal scheduling for a workshop [15].

The BBS method has the following features, which together distinguish it from the ordinary B & B method.

- The vertices of the BBS method are simultaneously heads of train event lists (variants of train movements) of the discrete simulation.
- An event for each train occurs only once on the event list.
- Each arc leaving particular vertex represents one value of an admissible control.
- It is assumed a limited horizon of prediction.
- For each vertex (variant) the lower bound of General Measure of Effectiveness is calculated for the end of the assumed horizon of prediction.
- Lower bound of GME (LBGME) is a sum of weighted train delays calculated for a particular moment of simulation time and the lowest possible change of this delay since the particular moment to the end of prediction horizon.
- At each new step of the algorithm, there is selected a vertex which has the lowest value of LBGME among the vertices not developed further yet and from its event list it is selected a the nearest event.
- The system has a feature that if simulation (model) time tends to the end of the prediction horizon, then the LBGME tends to the optimal value of GME calculated for this moment.
- The important issue is a determination the time of the first event for a variant created with a stopping train. The time is so called "propitious moment", which is characteristic for the BBS method.

In Fig. 1 there is presented a sample of the computer memory condition during variant generation for the BBS method. The vertices represent variants of the partial solutions. The vertices with numbers 3,5,7 represent the current variants which can be further developed. Each of them has a list of events for trains. For each train

there is only one event on the list and vice versa. There are four events (trains) for each variant positioned on the list according to not increasing value of the event times. The vertices with number 1,2 represent previous variants developed already. Their list of events has been removed. Instead of that they contain records of controls which caused their creation. Each vertex has in a small rectangle nearby a value of LBGME which is assigned to it. The straight arcs which connect vertices show the sequence of their creation, which will be needed when an optimal solution is obtained. The vertices 3,5,7 are located on the list of variants in the sequence of not increasing value of their LBGME. The head of the list is the vertex with a black circle.

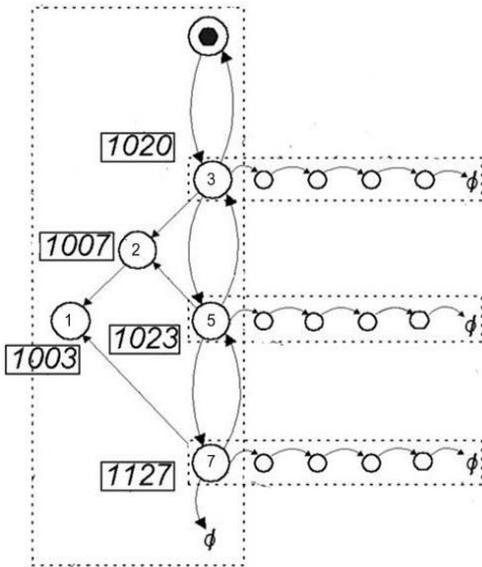


Fig. 1. The condition of the computer memory before variant generation for the BBS method used for dispatching problem solution

The selected variant for a further development is the variant with the number 3, because this variant is located on the first position of the variant list and therefore has the lower value of LBGME among all other variant. The event for the train which is located on the first position of the event list is selected for this variant for their execution time is the lowest among other events. This event is executed and would create new variants which

would be included to the variant list. The current variant 3 in such a case will be removed from the variant list, its events will be also removed and it will be located in such a position, that the straight arc still will connected this variant 3 and variant 2. If three new variants are created, then three new straight arcs will connect these variants and variant 2. Each of these variants will be included to the list of variants in the position related to the value of its LBGME.

**4. Modification of the BBS Method**

The task of determining and scheduling vehicle routes is similar to the task of a train dispatching problem. There, existed a variant of partial solution being train event list with assigned value of LBGME. Here, the variant of partial solution with an assigned value of LBGME for the selected vehicle is a list of previously designated shipping points for a selected route. These points are arranged on the list in the order of their visits by the vehicle. However, due to the possibility of simplifying the implementation of the algorithm a table is used instead of the list. In both tasks a variant with the lowest value of LBGME is selected for further development. The task executed for the chosen variant generates so many new variants as many possible controls exist for the train from the first position in the event list of the variant. The task executed for vehicles generate so many new variants as many following shipping points exist from the last designated point for the partial route described by the selected variant. The task related to trains ends with the determination of the optimal solution when the time of the event for a particular train exceeds the assumed time horizon and LBGME for the chosen variant is not greater than LBGME for each of the previously generated variants. The task related to vehicles ends with the determination of the optimal solution, when considered shipping point is the last point of the route, and LBGME for the chosen variant is not greater than LBGME for each of the previously generated variants. The algorithms which determine the optimal solutions for both tasks are very similar.

For the version of the BBS method used for solving the problem of a route determining and scheduling an useful convention of numbering of orders, pickup and delivery points was introduced. Thanks

this convention the optimization algorithms were simplified. On the basis of the number of one of these three, you can instantly (without knowledge of the total number of orders) determine the value of the leaving two.

*The convention of orders, pickup and delivery points numbering*

Z – set of orders  $Z=\{1,2,z,\dots, n\}$

P – set of pickup points  $P=\{1,3,p,\dots, 2n-1\}$

D – set of delivery points  $D=\{2,4,d,\dots, 2n\}$

$p=2z-1$

$d=2z$

$z=(p+1)/2$

$z=(d+1)/2$

For the problem of determining and scheduling vehicle routes using BBS method we can assume similar memory content as described in Fig. 1 for the dispatching problem. Let us simplify this content. First, we remove events from the event list, because we have only one vehicle and therefore one event instead of many events for many trains. Second, we resign from having a variant list because we can select a variant with the lowest value of LBGME by reviewing all the variant and the list is not absolutely necessary. Third, we can replace the development tree in the left part of the picture by separate list with repetition of some nodes. The result is shown on the Fig. 2.

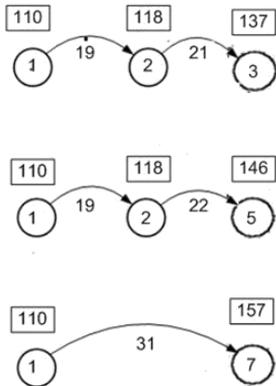


Fig. 2 The condition of the computer memory before variant generation for the BBS method used for the vehicle problem solution.

As before, there are three variants. Now the variants represent possible parts of a route as lists of vertices representing shipping points. The numbering of vertices follows the given numbering convention. Values of LBGME are different as in Fig. 1, because it is a different problem. Here, the values of LBGME are the lowest possible number of kilometers to travel from the current position up to the end of the route. There are numbers displayed by arcs. They are already known numbers of kilometers between pairs of the selected arcs. For generating a new variant we should select a variant associated with the last vertex having the lowest value of LBGME. It is the variant having vertex with number 3. In Fig. 3 there is shown a condition of the computer memory after new variants generation. Assuming that the route contains 8 shipping points (4 orders), the next following shipping points would have numbers: 4,5,7 only. They cannot have numbers 1,2,3 because they were used before.

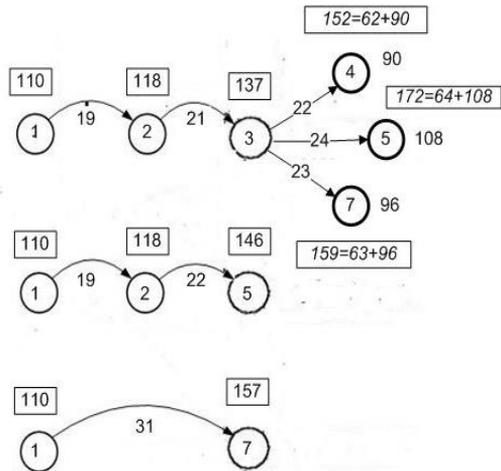


Fig. 3 The condition of the computer memory after variant generation for the BBS method used for the vehicle problem solution

They cannot have numbers 6,8, because these numbers represent delivery points (even numbers, see convention) and on the list there are no associated pickup points (5,7). For each of new shipping points generated it should be associated a value of LBGME. Let us calculate it for the shipping point with number 7. The value of

LBGME for this point us a sum of two addends. The first addend represents distance traveled up to the new shipping point  $63 \text{ km} = 19 \text{ km} + 21 \text{ km} + 23 \text{ km}$ . Second addend 96 represents an admissible heuristic equal to the lower bound of the remaining distance of the route. This value is displayed by the new shipping points and calculated by a special algorithm based e.g. on minimum spanning tree algorithm. The sum of the two addends which is LBGME for the point 7 is 159. The split list will be recorded as new three lists with the same three points 1,2,3 repeated. Instead of lists we can use an array, so was shown in Fig. 4. The next variant selected for further development is variant with shipping point having number 5 because it has the lowest value of LBGME (equal 146) among the shipping points which are last on their partial routes.

1/110	2/118	5/146		
1/110	7/157			
1/110	2/118	3/137	4/152	
1/110	2/118	3/137	5/172	
1/110	2/118	3/137	7/159	

Fig. 4. Lists from Fig. 3 recorded as an array

Fig. 5 is a graph illustrating the process of the variant generation. The abscissa shows the number of shipping points included in the previously generated variants, while the ordinate is the value for the variant LBGME. The broken line shows the sequence of variant generation. The process of generating variants begins in the lower corner of the graph with generating an initial variant and ends in the upper right corner of the graph with generating optimal variant containing the last tenth shipping point of the route.

The line chart breaks down only for integer values of the number of loading points, and the value LBGME in course of new variants generation never decreases. However, increasing and decreasing the number of points included in the routes for subsequent partial variants occurs. It is a feature of the BBS method. To find the optimal solution just

12,346 variants of 907,800 possible there were generated in the example. The optimal solution contains all shipping points of the route, in this case ten points for five orders.

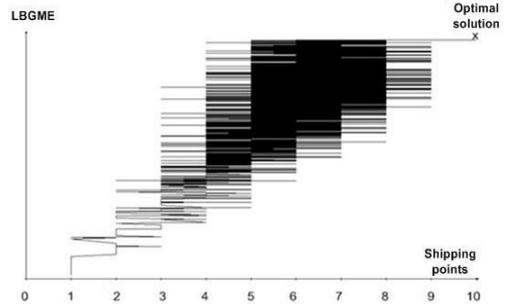


Fig. 5. Graph illustrating the process for obtaining the solution during the process of generating variants

### 5. The allocation of orders and scheduling vehicle routes

The allocation of orders and the scheduling vehicle routes is implemented in the form of a supplementary module to T-Traco system. There is an assumption to plan a simultaneous execution of up to 300 orders with the number of vehicles is 100 – 150. This task is large in relation to those described in the literature. At this size of the task it should not be expected the obtaining the optimal solution, but a solution only close to optimal. This result is achieved by means of two-level methods. On the upper level the task of allocation orders to vehicles is solved. Description of the solution of this task will be the subject of a separate publication. In this paper we confine ourselves to the description of the tasks solved at the lower level. It is the task of determining and scheduling routes for each vehicle. This task is very similar to the task of a Travelling Salesman Problem (TSP). But it contains a number of restrictions on the sequence of visited shipping points, time windows, vehicle capacity, limited working time of drivers and simultaneous carriage of goods of different groups. The diagram of a two-level problem solving method is given in Fig. 6. The solution to the problem begins with an initial allocation of orders  $\underline{z}(k)$  for each vehicle  $k$  on the upper level on the basis of a command from the system T-Traco

obtained through the database. This allocation, together with the matrix transit times (and / or distance) between pairs of points  $T(k)$  is sent to the lower level. At this level, the designation order is made for the vehicle  $k$  traveled through various pick-up and delivery points belonging to the order of  $z(k)$  transferred from the upper level. Also, there are calculated arriving times  $B(k)$  for vehicle  $k$  for individual points (pickup and delivery duration times for calculating the departure time are known) and there is calculated the value  $Q(k)$  of the LBGME. This indicator can provide the total expecting driving and rest time during execution of the route, the expected amount of kilometers traveled, or the expected cost of orders for this route. These parameters are passed to the upper level. On the basis of these parameters obtained for all vehicles it is calculated the new allocation of orders for each of the vehicles by an optimization procedure, which are transmitted to the lower level. Iterations end on the upper level if a solution is obtained for all vehicles. The solution obtained at the upper level is sent to the system T-Traco through the database. The task at the lower level is solved using the method BBS (Branch and Bound Simulation).

where the first dimension represents the maximum possible number of generated variants, the second is equal to the number of shipping points in the considered route, and the third is equal to the number of stored parameters (e.g. LBGME, shipping point identifier, distance, time, cost, driving time, etc.). After calling the procedure a starting variant is created, as basis for further variant development. Let us assume that after some new variants were created. Then we select the variant with the lower value of LBGME among all the previously generated variants. Next, taking the variant we try to expand it many times adding at its end one different shipping point taken from the array  $z()$  and creating each time a new variant with different ending shipping point. The new variants have the same shipping point as the old one except the last ones.

Some shipping points from the array  $z()$  are not acceptable as they already exist in the created variant, or for considered delivery points, related pickup points did not occur in previous positions of the variant. In addition, the vehicle may arrive at the shipping point after the time of its closure or its allowable cargo capacity would be exceeded after loading the goods. Such points are omitted and new variants are not created for them. When the allowable driving time has been exceeded, the driver takes a break in driving required by law. When the vehicle arrives before the opening of the shipping point, it has to wait for its opening (we can also take into account that if the waiting time would be too long and then a new variant will not be created).

The new variant is generated in the way that we occupy new elements in the array  $A$ , increasing the rate of the first index by one, and we copy the contents of the elements that made up the old variant to the new one. Then the elements with a new index are complemented by values related to a new shipping point including a new calculated value of LBGME for this variant. We calculate new values as the estimation of the minimum travel time (or the minimum length or cost, depending on a criterion of optimization ) for the rest of the route containing shipping points which were not included to the previously generated variant. These steps are repeat until the exhaustion of all acceptable points of loading of the entire route.

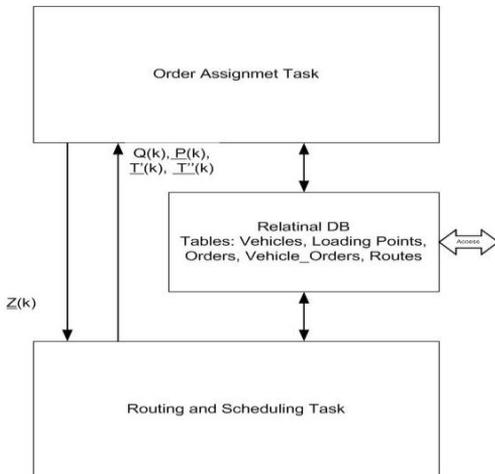


Fig. 6. Two-level problem solving method

**6. Description of the algorithm**

A simplified chart of the procedure is shown in Fig. 7. The recording of variants is done in a three-dimensional array e.g.  $A(0:200000, 0:20, 0:8)$ ,

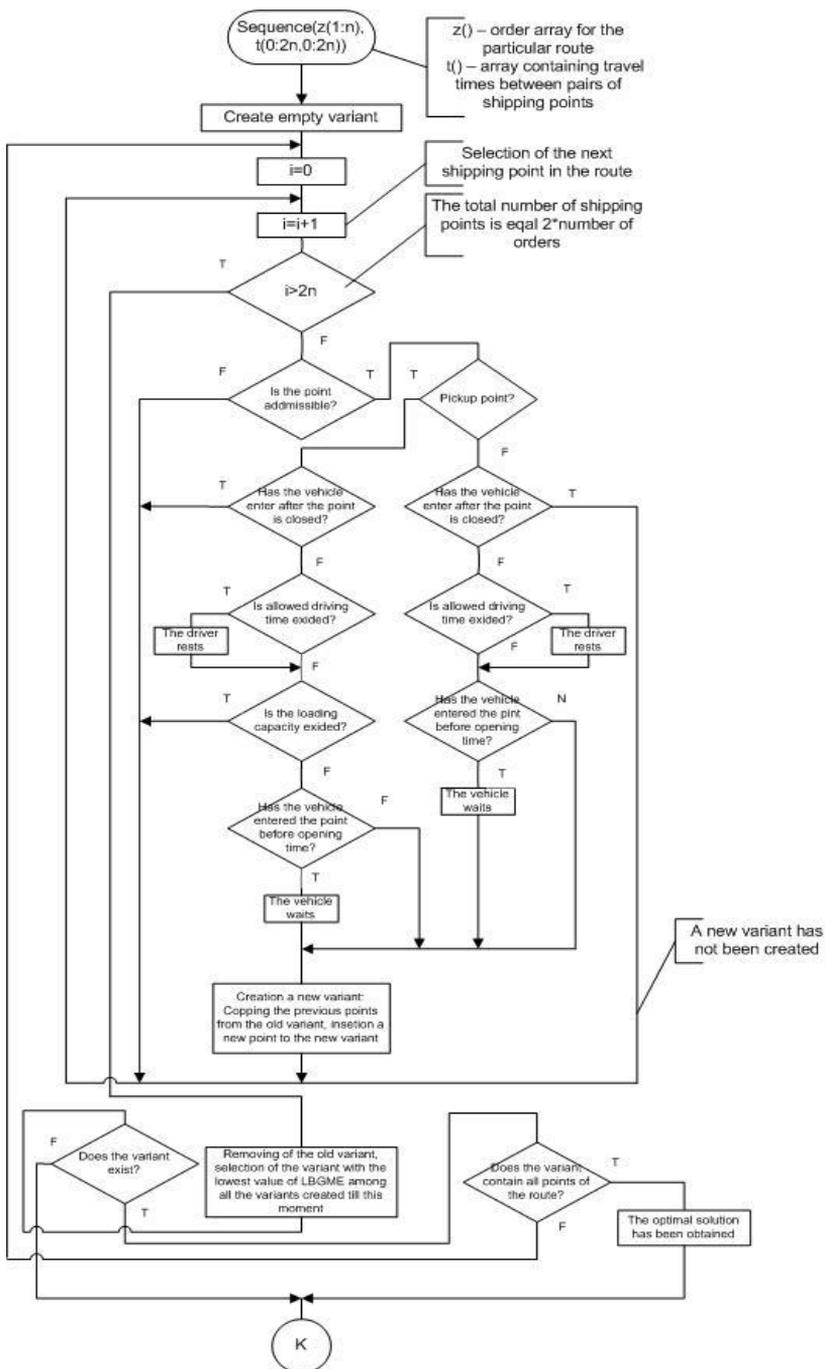


Fig. 7. Simplified flowchart of the procedure

Then we select variant with the smallest value of LBGME from all previously generated variants. If the planned route contains more than 12 shipping point (the value was determined in practice), the time of the determination of the optimal solution would be too long. Then, at the expense of giving up the optimal solution, we can speed up the calculation by limiting review of a set of variants to its subset [16]. There is a danger that we can skip a variant that leads to the optimal solution. However, there is an estimate [16] which determines the maximum distance between the obtained solutions and the optimal solution.

The selected variant for the development may not exist if we have not generated yet a single acceptable variant. Then we finish the procedure by notifying the program at the higher level of the absence of a solution for a given set of orders. When there is a variant with the lowest value of LBGME, then we check to see if it contains all the shipping points of the route. When this condition is met we obtained the optimal solution and we terminate the procedure. When the condition is not satisfied, we need to expand a partial solution of the following shipping points for the route.

In Fig. 8. a sample of the optimal solution for the vehicle 3 is shown using Google API. The route is the shortest one. It contains starting point A and 12 shipping points. The last delivery point is point M.

Each segment of the route is listed by its number, letters of beginning and ending point on the map, identifiers associated with these points in database, times for entering and leaving the segment by the vehicle, length of the segment in kilometers. By the segment numbers sometimes appear letters with numbers. Their meaning is: W– waiting time for shipping point opening in minutes, P – driver pause in minutes, O – driver daily rest time in minutes. At the end of the list there is a summary: total time: 24:45, total length in kilometers: 599.7.

**7. Completion**

We have been currently working on the integration of the module performing the task of order allocation to vehicles and the task for determining routes and scheduling with the application T - Traco. At the same time algorithms and procedures for performing these tasks are tested, and improved. In particular, different algorithms are tested for calculation of LBGME for the task of the determination of optima routs, since they are critical to the execution time of the program. After the final integration of the module with the application T-Traco studies will be carried in attempting to determine the accuracy and efficiency of the designed application in relation to the similar already operating.



Fig. 8 Sample of an optimal route

**References**

- [1] Cordeau J.F. A branch-and-cut algorithm for the dial-a-ride problem, *Oper Res* 54, str 573–586, *Inform*, 2006
- [2] Cudziło M., Zarządzanie zleceniami dostaw w rozproszonej strukturze magazynowej, *Gospodarka Materialowa i Logistyka*, 2011 nr 2, pp. 12-17, 2011
- [3] Fabri A., Recht P, On dynamic pickup and delivery vehicle routing with several time windows and waiting times. *Transport Res B-Meth* 40, pp. 335–350, 2006
- [4] Fagerholt K., Christiansen M, A travelling salesman problem with allocation, time window and precedence constraints – an application to ship scheduling. *Int Trans Oper Res* 7, pp. 231–244, 2006
- [5] Ghiani G, Guerriero F., Laporte G., Musmanno R., Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *Eur J Oper Res* 151, pp. 1–11, 2003
- [6] Hajdul M., Komodalność, czyli efektywna organizacja procesów transportowych, *Logistyka*, 2009 nr 2, pp. 26- 27, 2009
- [7] Hajdul M., Cudziło M., Metody i narzędzia wspierające wielkoobszarową integrację procesów logistycznych z wykorzystaniem nowoczesnych technologii, *Nowoczesne rozwiązania technologiczne w logistyce*, pod red.: Golińska P., Stajniak M., Wydawnictwo Politechniki Poznańskiej, pp. 69-81, 2010
- [8] Hernandez-Perez H., Salazar-Gonzalez J.J., A branch-and cut algorithm for the traveling salesman problem with pickup and delivery. *Discrete Appl Math* 145, pp.126–139, 2004
- [9] Lau H.C., Liang Z, Pickup and delivery with time windows : Algorithms and test case generation. *Int J Artif Intell Tools* 11, pp. 455–472, 2004
- [10] Lu Q., Dessouky M.M., An exact algorithm for the multiple vehicle pickup and delivery problem. *Transport Sci* 38, pp. 503–514, 2004
- [11] Parragh, S.N., Doerner, K.F., Hartl, R.F., A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. *Journal fuer Betriebswirtschaft* 58, pp. 81-117, 2008, Wien
- [12] Ropke S., Cordeau J.F., Laporte G. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks* 49, pp. 258–272, 2007
- [13] Szyszka G., *Innowacyjna logistyka, Fakty Magazyn Gospodarczy*, 2(68), str. 12-13, 2014
- [14] Wolfenbutg, A., Zastosowanie Symulacji Dyskretnej do Rozwiązania Zadań Optymalizacyjnych, *Prace Naukowe Instytutu Organizacji i Zarządzania Politechniki Wrocławskiej*, Nr 83, Seria Studia i Materiały 25, pp. 201-213, 2007, Wrocław
- [15] Wolfenbutg, A, Metoda BBS i jej wykorzystanie do optymalnego harmonogramowania prac na maszynach, monografia: *Modele symulacyjne i gry menedżerskie we wspomaganiiu decyzji w dydaktyce*, Oficyna Wydawnicza Politechniki Wrocławskiej, pp. 111-118, 2010, Wrocław
- [16] Wolfenbutg, A, Optymalne kierowanie ruchem pociągów w obszarze sieci kolejowej, *Wydawnictwa PWSZ w Gorzowie Wlkp.*, 2011, Gorzów Wlkp.
- [17] Xu H., Chen Z.L., Rajagopal S., Arunapuram S., Solving a practical pickup and delivery problem. *Transport Science* 37, pp. 347-364, 2004
- [18] Wolfenbutg A., Wyznaczanie i harmonogramowanie tras pojazdów przy wykorzystaniu metody BBS, *Logistyka* 4/2014, CD nr 2, Poznań 2014