Distance Matrix Generation for Dynamic Vehicle Routing Optimization in Transport Fleets Management

Radosław Belka and Mateusz Godlewski

October 7, 2021

Distance Matrix generation for Dynamic Vehicle Routing optimization in transport fleets management

Radosław Belka[1[0000-0002-6283-3724]] and Mateusz Godlewski

[1] Kielce University of Technology, Al. 1000-lecia P.P.7, 25-314 Kielce, Poland

lncs@springer.com

Abstract. The article focuses on the dynamic update of the distance matrix, one of the key datasets used in the optimization of transport issues. In the case of a dynamically changing list of destinations, a continuous and effective update of the data is required, e.g. from more and more popular services such as Mapping APIs. The time-consuming nature of this process, which may extend the planning process, was emphasized. The article discusses the possibility of generating temporary values of the distance matrix based on the correction of the quasi-Euclidean distance. The impact of update progress on the some optimization algorithms was investigated. The research was carried out on the example of the real VRP problem. The obtained post-optimization results could be compared with the results obtained manually by experienced planners. It was found that the degree of update of the distance matrix influences the cost reduction in a nondeterministic way.

Keywords: WorkForce Management, Vehicle Routing Problem, Distance Matrix, Greedy Algorithm, evolutionary algorithm

1 Introduction

One of the basic requirements signaled by the TLS (Transport, Logistic, Spedition) market is the development of an IT solution enabling effective, quasi-optimal and quick planning of deliveries to given recipients (destinations). The challenges are variants of well-known issues such as VRP (Vehicle Routing Problem) [1-2] and WFM (Work Force Management) [3-4]. However, optimization tasks are often performed manually, and the effectiveness of tasks is related to the competence and experience of logistics and planners. The proposed routes are usually beneficial in the case of a static list of delivery points, but they may differ from optimal solutions, in particular frequent changes in the distribution of delivery points. Therefore, improving management efficiency by implementing automated WFM problem optimization systems is still a key market challenge.

In the case of transport companies (transport fleets) the WFM is strongly related to the VRP, one of the most popular combinatorial, NP-difficult problem. The issue is to find an optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers. There are many different variants of the VRP problem and methods for solving them. In [5], scientific related paper from 2009-15 were analyzed, in which 327 computational models were presented. Among the methods of solving the problem, metaheuristic methods dominate: Genetic Algorithm, Simulated Annealing (SA), Tabu Search (TS) Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), which were used in over 70% of models. Often there are two-stage solutions, in which the first stage is based on a simple heuristic and gives an approximate solution, while the second one allows you to improve the obtained results. The above-mentioned paper [5] and authors' experience point to the growing importance of the Dynamic VRP variant, which takes into account dynamic request updates during the service provision. However, the concept of a request should be understood not only as the quantity and type of goods, but also the place of their destination. This means that the number and / or geolocation of destinations may change continuously, for example, in companies that operate portable toilets. In such a case, it may be necessary to re-plan routes, taking into account changes in the spatial distribution of recipients.

This article focuses on the issues of dynamic and quick recalculation of the value of the so-called Distance Matrix (DistMx) [6].It is one of the most key data structures used in planning and optimization algorithms. Regardless of the applied problem, the DistMx must present the current distances between the objects, generating from their current position. The proposed solution is dedicated to the dynamic VRP variant, in which the set of current delivery points (destinations) may change frequently. It was pointed out that a reasonable updating of the distance matrix may be a key factor affecting the effectiveness of the optimization algorithms in the variant of dynamically changing requests. Moreover, in evaluation the effectiveness of the optimization algorithm, the influence of the current values of the distance matrix is rarely examined. This aspect needs to be discussed.

2 Methods and Data Sets

In the case of transport companies, the issue of WFM is related to the proper assignment of resource space elements (vehicles, drivers) to the request space elements (orders, requests) in such a way as to minimize the defined cost function. Acceptable, quasi-optimized solution must be find within several minutes. It should be remembered that after planning, the goods are loaded in a given order, as a result, subsequent corrections may be limited. The challenge that was encountered belongs to both the Capacitated [7] and Split Delivery [8] variant of the VRP problem. Moreover, the considered variant takes into account the existence of several types of vehicles with different capacitance. In addition, selected destinations are subject to availability restrictions for a given type of vehicle. This means that not every vehicle can be used to service every request.

For the needs of the study, a two-stage optimization algorithm was developed, ensuring a favorable balance between the degree of optimization and time-consumption, as well as characterized by good scalability. The first stage is based on the "greedy" algorithm, which in the selection of orders for vehicles is guided by the local best hop approach. this stage is deterministic, i.e. it always returns the same result with the same input data set. In the second stage, the orders are randomly adjusted based on the so-called "Reasonable" Exchange (RandSwap). It was a proprietary implementation of a variant of the 2-opt algorithm, often used to solve the traveling salesman problem [9]. Total distance and total fuel cost criteria were the main optimization criteria. Full and detail description of the algorithm and its performance tests go beyond the scope of this article. However, in order to investigate of the proposed solutions, a demonstration version of the algorithm was used to compare the influence of the matrix form on the efficiency of the optimizer.

The algorithm uses the following main inputs: RequestList, TruckList, and Distance Matrix. Representative and actual input data of RequestList and TruckList was obtained from a local shipping company – Alma-Alpinex Join-Stock Company, which redistributes food products and manages a fleet of approx. 100 vehicles. The sample output problem solution of the problem (Service Record), including route sequences, proposed by experienced planners, was obtained from the same source. Total 1103 pallets divided to 416 individual request were redistributed to 202 destination from one central depot. The six type of trucks with different capacity (27, 21, 20,18, 10 and 8 pallets) were considered as candidates. Obtaining the details data regarding the Destination List and Distance Matrix will be described in below section.

The Distance Matrix (DistMx) is a square matrix of values where the value DistMx [i] [j] denotes the distance (in km) from destination number i (DestNo = i) to destination number j (DestNo = j). Due to the possibility of the occurrence of one-way routes, the matrix is not necessarily symmetrical. In general, the quantities in the table can be any measure of any metric space, however the simplest example is the distance represented by total mileage. Alternative variations are elementary costs represented by travel time or fuel consumption. The significance of the distance matrix in transport issues was discussed e.g. in the work [10]. In [11] several different models that allow you to estimate the value of the distance based on the knowledge of the location of the destinations were presented. The trend of moving away from the universal, static matrix of distances towards individual and dynamic personalized variants was, in turn, shown in [12]. The basis for creating the DistMx matrix is a list of DestPos destinations, containing at least the destination identifier (DestId) as well as geolocation coordinates (latitude, longitude). Location coordinates are the basis for obtaining the distance value. For N destinations, a matrix with dimensions NxN will be created.

3 Results

3.1 Getting the Destination List

The basic step is to obtain an up-to-date list of destinations with their coordinates. In the system under development, these data are obtained in integration with the GPS-based vehicle location system - the ABC-Track system. The Information was obtained thanks to courtesy of the owners - ABC-Track Ltd company. In order to obtain resources, requests were sent using the HTTP - GET method to the address indicated in the documentation. The data returned in response to the sent request was in JSON (JavaScript Object Notation) format. The system includes many services – in this moment the most important was getPoints, responsible for providing information about individual points on the route map. The data was collected and processed using scripts written in Python. The obtained data structures in JSON format were imported to the newly created DataFrame object of the Pandas library for further analysis and visualization. As a result of a query to the site, a list of objects was returned. Each of the objects is a representation of a particular point on the map, and has the following information:

- pointId - unique identifier of the point.
- pointName - point name.
- pointLatitude - the latitude of the point.
- pointLongitude - the longitude of the point.
- pointStreet - the street where the point is located.
- pointHouseNumber - number of the house where the point is located.
- pointPostalCode - zip code of the place where the point is located.
- pointCity - name of the city in which the point is located.
- pointDescription - point description.

Using the APIs the basic list includes 203 points (1 Depot and 202 destinations) were downloaded . The presented destinations were located in an area with an estimated size of 160 (E-W) x 145 km (N-S).

3.2 Getting the DistMx from Mapping APIs

Based on the list of destinations, it is possible to obtain information about mutual distances thanks to the use of "Mapping APIs". Currently, many geo-location services are known, capable of generating an distance matrix based on a given list of destinations [13-16]. One of the most popular is the Google API [16], associated with the popular Google Maps service. More precisely, it is the Distances-Matrix API service that allows you to calculate the distance between given points on the map. This service takes the coordinates of the given points and returns the travel distance in kilometers. Accessing the Distance Matrix service is asynchronous, since the Google Maps API needs to make a call to an external server. For that reason, you need to pass a callback method to execute upon completion of the request, to process the results. The access the Distance Matrix service is possible within a code via the google.maps.DistanceMatrixService constructor object. The DistanceMatrixService.getDistanceMatrix() method initiates a request to the Distance Matrix service, passing it a DistanceMatrixRequest object literal containing the origins, destinations, and travel mode, as well as a callback method to execute upon receipt of the response.

The limitation of this service is the number of API requests that can be made within the free account - it is possible to obtain a travel distance between 40,000 pairs of points on the map, which allows you to create an DistMx with a maximum size of 200 x 200. In addition, the service is paid, but the current the cost is estimated at around 5$ per 1 thousand queries. For about 200 destinations, the cost of the service is estimated at just over 200 USD.

An alternative solution is a free service offered by the Open Source Routing Machine (OSRM) [17]. OSRM is a C++ implementation of a high-performance routing engine for shortest paths in road networks. As part of the creation of the Distance Matrix, the Table service was used, which calculates the duration of the fastest route between all pairs of given coordinates. Returns the duration and distances between pairs of coordinates. This service takes 4 arguments:

- coords_src - list of geographical coordinates of the starting points.
- coords_dest - list of geographic coordinates of endpoints.
- ids_origin - list of identifiers of starting points.
- ids_dest - list of endpoint identifiers.

In order to create an Distance Matrix, the identifiers and geographical coordinates of all destinations should be provided as starting and ending points.

In the event that a new destination appears, there is no need to recalculate the entire matrix. A version of the script has been created, which works in such a way that for a particular destination (column) it calculates the distances between it and the rest of the destination (lines). Thus, iterating over N destinations creates an NxN matrix. This way is to work around the limit on the engine side of the simultaneous number of data sent in the query. In order to add a new destination, it will only need to calculate one new column for a given destination. And transpose that column into a row, which then also add to the Distance Matrix. Such a solution is beneficial in the situation of a dynamically changing list of destinations. It has been experimentally found that updating the distance table in the event of an additional destination takes an average of a few seconds.

3.3 Temporary distance values

It was found that acquiring distance values from the OSRM website is possible, however, it may be long. In the case of sending individual inquiries, the time to complete the distance matrix was approx. 5 hours for N = 203 points. In the case of sending aggregated questions, it was possible to shorten this time to about 20 minutes, with this time being approximately linearly dependent on N. In the event of unfavorable connectivity or problems with the service overflow, this time may be extended. Therefore, an alternative method of generating provisional values should be proposed.

In the absence of current values obtained, for example, from the OSMR website, the distance can firstly be estimated based on the quasi-Euclidean metric. In order to find the shortest route d between two points located on a sphere with radius R, the "haversine" formula can be used (1).

$$d = 2 \cdot R \cdot arcsin\left(\sqrt{hav(\varphi 1 - \varphi 2) + \cos(\varphi 1) \cdot cos(\varphi 2) \cdot hav(\lambda 1 - \lambda 2)}\right) \quad (1)$$

where
φ1 , φ2 - latitude of point 1 and 2, respectively
λ1 , λ2 - longitude of point 1 and 2, respectively
R – Earth's radius (~6371 km)

Used above the haversine function has the form

$$hav(\Theta) = sin^2\left(\frac{\Theta}{2}\right) = \frac{1-cos(\Theta)}{2} \tag{2}$$

The distance determined from relation 1 is usually underestimated as the routes connecting 2 points are usually not a straight line. The relative underestimation error is derived from the following formula (3).

$$Err = \frac{d_{OSM}-d_{hav}}{d_{hav}} \tag{3}$$

where:
d_{OSM} – the actual distance obtained from the OSMR service
d_{hav} – theoretical distance calculated from the dependence (1)
The dependence of the error value on the actual distance value was investigated. Figure 1 shows the scatter plot, in which the error values are presented on the horizontal axis (in linear scale), and the actual distance value on the vertical axis (in log scale). The range of the error value is 0.0184 – 7.358

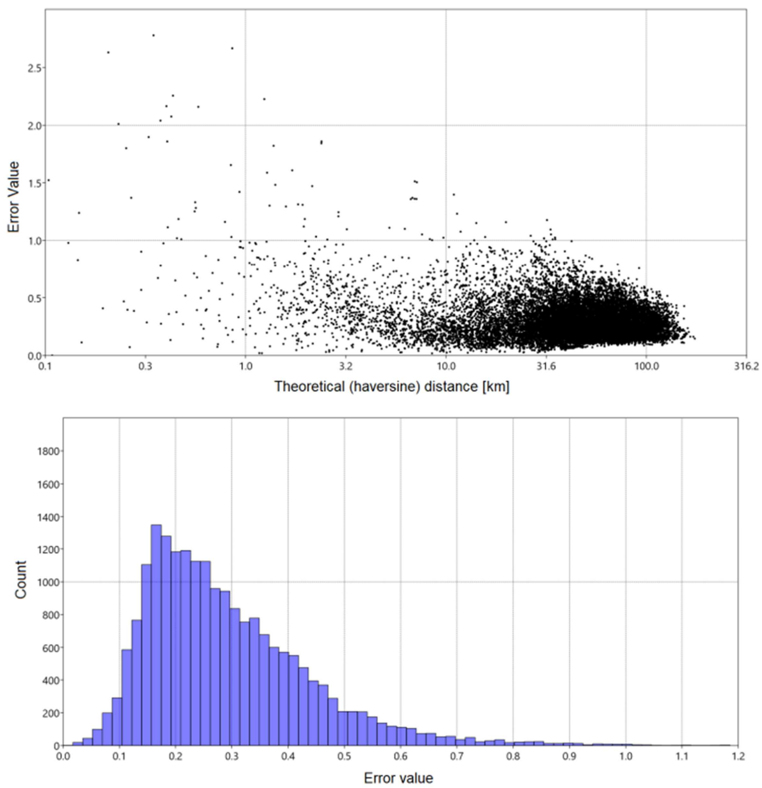

Fig. 1. Scatter plot of error-distance relation (up) and relative error distribution (down)..

The distribution is characterized by a clear skewness (mean value 0.296, median 0.260, standard deviation 0.1886, interquartile range 0.1832-0.3677). It was found that ~ 0.4% of the results refer to errors above 100%.
The scatter plot suggests that for short distances (<3 km) the error value (underestimation level) may be significantly higher than for the entire sample. Therefore, the data set was segmented in relation to the distance value and for each of the segments the average error value was determined with the 95% confidence interval - Figure 2. The preliminary analyzes show that for ultra-short distances (below 3 km) the theoretical value of the distance should be corrected (multiplied) by a factor as high as 1.7. For longest distances (3-6 km) the correction factor can be estimated at the level of 1.35, and for the remaining distances at the level of ~1.28.

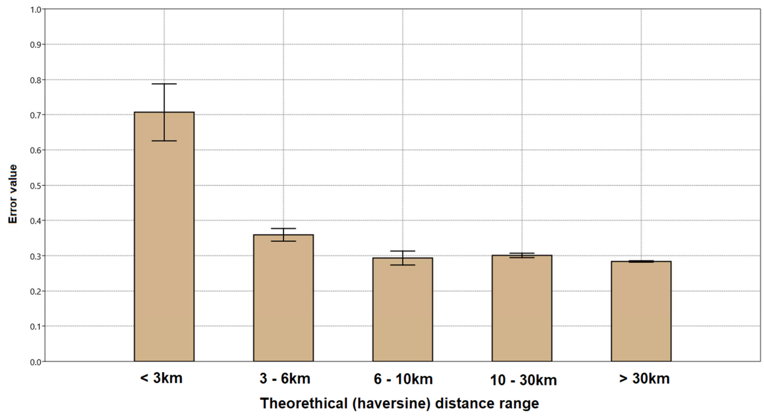

Fig. 2. Bar plot with errors representing relative error values in different distance range

Obtaining more correct values of the distance matrix can be achieved by designating an appropriate scaling function. This function can be an approximation of a d_OSM=f(d_hav) relation. On the other hand, it may be more advantageous to find a function that approximates the dependence of the error value (3) on the distance (1). In such a case, the general form of the function may be as follows (4):

$$Err_{approx}(d_{hav}) = a + b \cdot d_{hav} + c \cdot e^{-d \cdot d_{hav}} \quad (4)$$

In these considerations, it was decided to approximate the error value using the (3) formula. The a, b, c and d coefficients were determined by the method of minimizing the mean square error (MSE). Optimization calculations were made using non-linear Generalized Reduced Gradient method [3, implemented in the Solver tool of MicroSoft Excel software. For the data set (~ 40,000 values), the following coefficients were obtained: a = 0.2871, b = 0, c = 1.254 and d = 0.7663. Based on the dependence 3, the distribution of the correction coefficient as a function of the original value of the distance calculated from the dependence (1) was also determined – fig. 3.

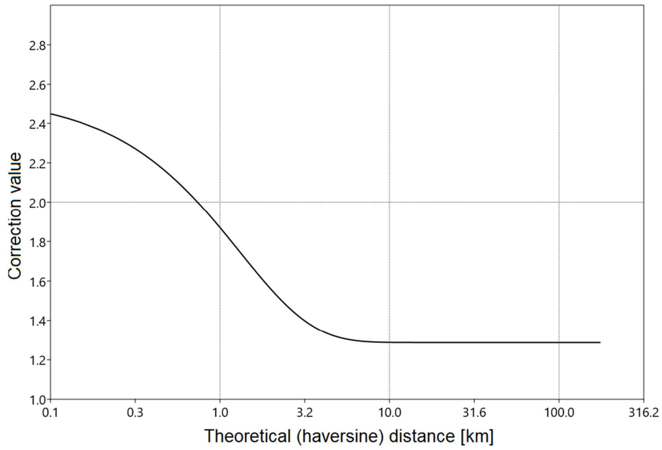

Fig. 3. Proposed distribution of correction factor vs original haversine distance

As a result of applying the dependencies (1) and (4), a distance matrix named DistHavCorr was calculated as follow:

$$DistHavCorr[i,j] = \begin{cases} 0 & if\ DistHav[i,j] = 0 \\ \big(1 + Err_{approx}(DistHav[i,j]) \cdot Disthav[i,j]\big) & otherwise \end{cases} \tag{5}$$

It was found that the arithmetic mean of the relative error between corresponding values of the DistOSRM and DistHavCorr matrices was <0.00001. Standard deviation (SD) was 0.1226 and was smaller than the deviation between DistOSRM and DistHav, where it was 0.1887. This was due to the introduction the exponential component in the approximation function, as a result of which the small values of the distance matrix were more effectively corrected. The average value of non-zero elements of the DistHavCorr matrix was about 0.5% higher than the corresponding value for DistOSRM.

3.4 Dynamic DistMx update

The idea of generating a dynamic Distance Matrix is presented in Figure 4. According to the proposed concept, the DistMx used in the optimization process is generated in the first stage on the basis of the relationship 1, taking into account experimentally proposed correction factors - see equation (5) and fig.3. However, the obtained corrected matrix is still characterized by a fairly large discrepancy in relation to the empirical data - the average error was estimated at approx. 12%.
For the generation and updating of the Distance Matrix for the purposes of the optimization algorithm the following flow-chart was proposed:

1. Get the actual DestPos list
2. Calculate the distance matrix based on the haversine distance (1).
3. Correct the haversine (quasi-Euclidean) distance according to formula (4).
4. Make an additional list containing distances sorted in increasing order
5. Send inquiries updating the distance values to the OSRM service, starting from the smallest values.
6. (Optional) Simultaneously send update queries to the database of routes included in the ABC-Track system. If the route is in the base (the distance is known), update the distance value.

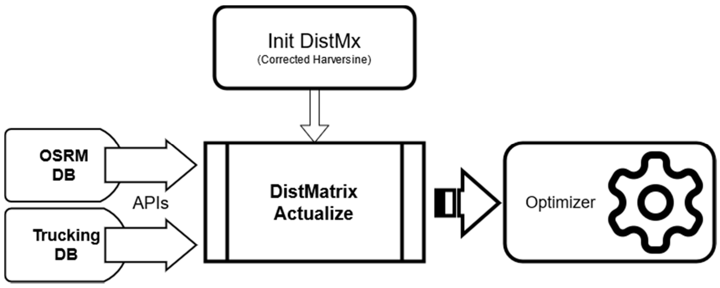

Fig. 4. Distance matrix actualisation idea for dynamic VRP problem

As indicated - step 6 is optional and depends on the completeness of the databases in the GPS-based vehicle traffic monitoring system. Such systems have become popular in recent years and are widely used in the process of managing vehicle fleets. One of such systems is the abovementioned ABC-track system. The resources of the system were available thanks to the getHistory service, which provides information about the historical journeys of individual vehicles, which was used to obtain the history of the journey of a particular vehicle. In order to obtain historical data, the sent query had to be parameterized with three attributes:

- vehicleId - unique identifier of the truck. The identifiers of each vehicle were obtained using the website getVehiclesExtended.
- datetimeStart - Date (Year-Month-Day hh:mm:ss format in the UTC time zone) that marks the start of the period from which to retrieve historical trip data.
- datetimeStop - Date (Year-Month-Day hh:mm:ss format in the UTC time zone) that marks the finish of the period from which to retrieve historical trip data.

Single inquiries about the history of journeys of individual vehicles, whose identifiers were collected from the getVehiclesExtended service, are sent to the website, from the period of 3 months (typical retention policy of database) The reply was a list of objects containing datetime, longitude, latitude, mileageKm (vehicle mileage in kilometers), as well as information such as heading (value read from the compass, which informs about the direction in which the vehicle is moving) and ignitionStatus (engine ignition status: 1 - ignition on, 0 - ignition off).

According to the assumption, the highest priority is assigned to routes and ABC-Track databases, then from the OSRM database and the lowest for data from the updated haversine distance. The ABC-Track distance has the greatest credibility, because it takes into account real driving conditions, restrictions imposed on trucks and the resulting vehicle behavior on the route, and also allows for labeling drivers in the long run. The least reliable is the theoretical temporary distances stored in DistHavCorr matrix. At the same time, the mean error is negligible, which makes it possible to accept these values as acceptable from the point of view of minimizing the global cost in the process of optimizing the WFM problem.

3.5 Influence of DistMx on optimizer cost reduction

One of the best methods of estimating the effectiveness of an optimizer is to compare the solutions it returns with the solutions proposed by experienced planners. As mentioned in section 2, both the order parameters (the size of requests, types and load capacity of vehicles, the location of destinations), as well as the proposed “manual” solution, including route sequences, were known. Thanks to this, it was possible to calculate the real distance (RealDist) as the resulting cost function. this value was a reference to the values obtained in the optimizer (OutCost). Obviously, both the RealDist and OutCost values are depended on the form of the assumed distance matrix. An open question is the dependence of the optimizer cost reduction on the form of the used matrix. This parameter can be defined simply as follows:

$$CostReduction[\%] = \frac{RealDist - OutCost}{RealDist} \cdot 100 \tag{6}$$

Using the optimization algorithm mentioned in section 2, OutCost values were obtained for both stage I (Greedy) and stage II (RandSwap). In the second stage, due to the stochastic approach, the 25 attempts with 100000 iteration were calculated.

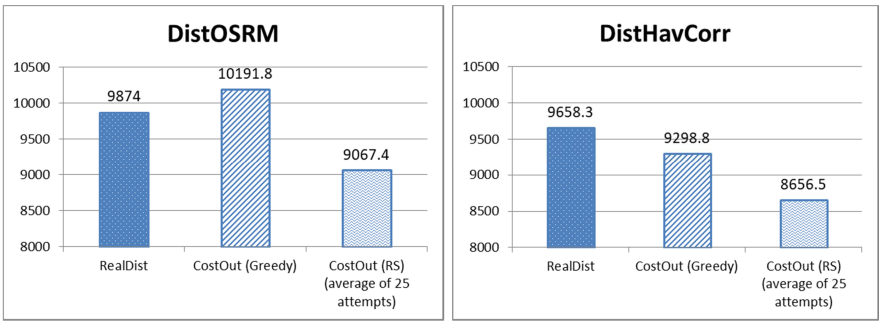

Fig. 5. Optimizer effectivity vs type of DistanceMatrix

Figure 4 shows a comparison of the cost values (in km of the route) obtained from the optimizer (stage I and stage II) for the two proposed forms of the used distance matrix. Although the average distance values in DistHavCorr were slightly higher than DistOSMR (~ 0.5%), the distance (cost) calculated on the basis of the result sequences was higher (by ~ 2%) for the case of using the temporary matrix. This result may be due to the fact that the proposed routes do not contain the hops corresponding to the largest values in distance matrix.

For the temporary distance matrix (DistHavCorr) obtained from the dependencies (1) and (5) it was found that after the first optimization stage it is possible to obtain a positive efficiency, which is then improved in stage II. However, in the case of using real distances obtained from the OSRM (DistOSRM) service, the Greedy stage is characterized by negative cost reduction. Only stage II allows you to obtain satisfactory results.
The observation made shows how important the correct form of the Distance Matrix is from the point of view of the optimizer's effectiveness. Assuming the theoretical values based on Euclidean metrics, one can erroneously hypothesize that the greedy algorithm is sufficiently effective (~ 3.7% cost reduction) in the process of planning deliveries. Meanwhile, substituting real values contradicts previous observations. Another unfavorable information is the fact that the efficiency of the optimizer turns out to be on the level of ~8.17% compared to the previously assessed ~10.37%, although it is still a satisfactory and acceptable value.

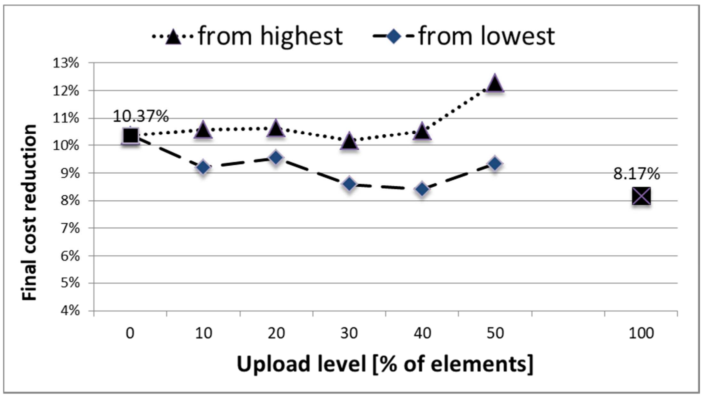

Fig. 5. Evolution of cost reduction in case dynamic matrix upload (starting from highest and lowest matrix values, respectively)

The order of updating individual elements of the distance matrix is another aspect worth to analyzing. As mentioned, it may take several hours to get the full real distance data. Communication interruptions and service loads must be taken into account. For this reason, it is advisable to successively update the DistHavCorr matrix values. The analysis of the resulting sequence of the optimization algorithm allowed us to make the following observations. First of all, the maximum value of a single distance (hop) is comparable to half of the maximum value contained in the DistMx table. This means that extremely large values do not need to be updated urgently. It was estimated that on the basis of the considered case, such values constitute approx. 10% of all matrix elements. Moreover, approx. 60% of the appeals used in the case of distances shorter than 20 km. For this reason, it makes sense to update the short distances first. They are, firstly, more often used in the optimizer, and secondly, the relative error value is greater for smaller distance values.

According to the assumptions made, more in-depth analyzes were made by examining the impact of the update level (in %) on the level of cost reduction. The 10 different mixed matrices were generated, divided to two series. In first series matrices data (so called mixed-matrices) was uploaded starting from lowest distance values, in the second – starting from highest ones, respectively. Upload level was 10%, 20%, 30%, 40% and 50% numbers of matrix's elements. Total cost reduction (in %) were calculated for any case. Results are presented in figure 5.

In both cases, the evolution of the cost reduction value starts from the 10.37% value previously returned for the DistHavCorr temporary matrix. With the "from highest" approach, the trend seems to be quasi constant in the update range of the first 50% of the values and is not tending to the final value of ~ 8.17%. In the "from lowest" approach, the trend seems to be tending towards the final value. This allows to defend the thesis about the validity of updating distances starting from smaller values.

4 Conclusion

In the article some aspects of the optimization issues in the field of WFM and VRP were discussed. In the case of dynamically changing destination lists, constant updating of the matrix form is required. The update process is possible thanks to services such as Mapping APIs, such as OSRM, and it can take a long time. The article proposes a method of initial initialization of the matrix form and then a reasonable approach in the method of successive updating of individual values.

There was shown, the form of the distance matrix may also affect the assessment of the effectiveness of individual algorithms. Obsolete or approximate (approximated) values may significantly affect the assessment of the effectiveness of a certain class of algorithms, in particular the greedy algorithm. Differences of more than 7 percentage points were found here. On the other hand, in the case of the RandSwap metaheuristic algorithm, these differences are slightly smaller - at the level of less than 2.5 percentage points.

The experiments and analyzes performed are part of a wider project, the aim of which is, inter alia, development of a practical, fast and effective quasi-optimal resource allocation algorithm in the conditions of dynamically changing conditions of the request space. Such issues were named as Agile Work Force Management and constitute an advantage over the used static solutions.

Acknowledgement

This research was founded by the European Union's Smart Growth Operational Programme 2014-2020, under grant agreement no POIR.04.01.04-00-0091/19-00.

The authors also thank the companies Alma-Alpinex Join-Stock and ABC-Track Ltd for providing services and valuable data,

References

1. Zhang, H., Ge, H., Yang, J., Tong, Y.: Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms. Archives of Computational Methods in Engineering, pp. 1-27 (2021).
2. Konstantakopoulos, G.D., Gayialis, S.P. & Kechagias, E.P.: Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. Oper Res Int J (2020).
3. Hota, J., Debjani Ghosh, D.: Workforce Analytics Approach: An Emerging Trend of Workforce Management. AIMS International Journal of Management 7(3), 167-179 (2013)
4. https://asana.com/pl/resources/workforce-management, last accessed 2021/09/05.
5. Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I.: The vehicle routing problem: State of the art classification and review. Computers & Industrial Engineering 99, 300-313 (2016).
6. Jambu, M.: Classification and Analysis of Proximities Data Sets, in Exploratory and Multivariate Data Analysis, 407-418 (1991).
7. Zhang, Q., Wei, L.R., Hu, R., Yan, R., Li, L.H., Zhu, X.N.: A Review on the Bin Packing Capacitated Vehicle Routing Problem. Advanced Materials Research 853, 668–73 (2013).
8. Archetti C., Speranza M.G. The Split Delivery Vehicle Routing Problem: A Survey. In: Golden B., Raghavan S., Wasil E. (eds) The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces 43. Springer, Boston, MA (2008)
9. Karagul, K., Aydemir, E., Tokat, S.: Using 2-Opt based evolution strategy for travelling salesman problem. International Journal of Optimization and Control: Theories & Applications 6(2), 103-113 (2016)
10. T. Karner, T., Weninger, B., Schuster, S., Fleck, S., Kaminger, I.: Improving Road Freight Transport Statistics by Using a Distance Matrix. Austrian Journal of Statistics 46(2):65-80 (2017).
11. Zgonc, B., Metka Tekavčič, M., Jakšič, M.: The impact of distance on mode choice in freight transport. European Transport Research Review 11, Article number: 10 (2019)
12. Kuehnel, N., Ziemke, D., Moeckel, R., Nagel,K.: The end of travel time matrices: Individual travel times in integrated land use/transport models. Journal of Transport Geography 88 (102862), pp.1-12 (2020).
13. https://distancematrix.ai/logistics-solutions, last accessed 2021/09/05.
14. https://traveltime.com/features/distance-matrix, last accessed 2021/09/05.
15. https://www.geoapify.com/time-distance-matrix-and-three-ways-to-use-it, last accessed 2021/09/05.
16. https://developers.google.com/maps/documentation/distance-matrix/overview, last accessed 2021/09/05.
17. http://project-osrm.org/ , last accessed 2021/09/05.
18. Kim, N. S., Van Wee, B. The relative importance of factors that influence the break-even distance of intermodal freight transport systems. J Transp Geogr, 19, 859–875. (2011).
19. Cooper, J. C. The use of straight line distances in solutions to the vehicle scheduling problem. J Oper Res Soc, 34 , 419 (1983).
20. Barthélemy, M. Spatial networks. Phys Rep, 499(1–3), 1–101 (2011).
21. Domínguez-Caamaño, P., Benavides, J. A. C., Prado, J. C. P. An improved methodology to determine the wiggle factor: An application for Spanish road transport. Brazilian J Oper Prod Manag, 1, 52. (2016).