ARCHIVES OF TRANSPORT
Volume 61, Issue 1, 2022

ISSN (print): *0866-9546*
e-ISSN (online): *2300-8830*
DOI: *10.5604/01.3001.0015.8269*

MACHINE LEARNING ALGORITHMS FOR THE PROBLEM OF OPTIMIZING THE DISTRIBUTION OF PARCELS IN TIME-DEPENDENT NETWORKS: THE CASE STUDY

Zbigniew TARAPATA[1], Wojciech KULAS[2], Ryszard ANTKIEWICZ[3]

[1, 2, 3] *Military University of Technology, Faculty of Cybernetics, Warsaw, Poland*

Abstract:

In the paper we present machine learning algorithms for the problem of optimizing the distribution of parcels in stochastic time-dependent networks, which have been built as a part of some Distribution Optimization System. The problem solved was a modified VRPTW (Vehicle Routing Problem with Time Windows) with many warehouses, a heterogeneous fleet, travel times depending on the time of departure (stochastic time-dependent network) and an extensive cost function as an optimization criterion. To solve the problem a modified simulated annealing (SATM) algorithm has been proposed. The paper presents the results of the algorithm learning process: the calibration of input parameters and the study of the impact of parameters on the quality of the solution (calculation time, transport cost function value) depending on the type of input data. The idea is to divide the input data into classes according to a proposed classification rule and to propose several strategies for selecting the optimal set of calibration parameters. These strategies consist in solving some multi-criteria optimization tasks in which four criterion functions are used: the length of the designated routes, the computation time, the number of epochs used in the algorithm, the number of designated routes. The subproblem was building a network model of travel times that is used in constructed SATM algorithm to determine the travel time between recipients, depending on the time of departure from the start location. An attempt has been made to verify the research hypothesis that the time between two points can be estimated with sufficient accuracy depending on their geographical location and the time of departure (without reference to the micro-scale, i.e. the detailed structure of the road network). The research was conducted on two types of data for Warsaw: from transport companies and one of the Internet traffic data providers. Learning the network model of travel times has produced very promising results, which will be described in the paper.

Keywords: *Modified Vehicle Routing Problem with Time Windows (M-VRPTW), Simulated Annealing Algorithm Learning, Network Matrix of Travel Times Estimation, Time-Dependent Networks, Optimization of Parcel Distribution*

To cite this article:

Tarapata, Z., Kulas, W., Antkiewicz, R., (2022). Machine learning algorithms for the problem of optimizing the distribution of parcels in time-dependent networks: the case study. Archives of Transport, 61(1), 133-147. DOI: https://doi.org/10.5604/01.3001.0015.8269

Contact:
1) zbigniew.tarapata@wat.edu.pl [https://orcid.org/0000-0001-8143-7869] – corresponding author; 2) wojciech.kulas@wat.edu.pl [https://orcid.org/0000-0001-6749-6563]; 3) ryszard.antkiewicz@wat.edu.pl [https://orcid.org/0000-0003-1489-4407]

1. Introduction

The routing problems are one of the most important in transport and logistics, from the practical point of view, and in operation research and theory of algorithms from theoretical point of view. One of the well-studied and useful is Vehicle Routing Problem with Time Windows (VRPTW) (Cordeau et al., 2002; Solomon, 1987). In the project described in (Development, 2020; Tarapata et al., 2020) we have developed modified Vehicle Routing Problem with Time Windows (VRPTW) which is a part of some Distribution Optimization System (Tarapata et al., 2020; Tarapata and Antkiewicz, 2021). The modification of the defined problem in relation to the classic VRPTW was based on the fact that we include many warehouses, a heterogeneous fleet, stochastic time-dependent travel times (stochastic time-dependent network (Vidal et al., 2020) and an extensive cost function (the total weighted costs of delivery of goods, including the costs of: permanent use of vehicles, work of vehicles resulting from the distance travelled and work of vehicle drivers resulting from their working time) as an optimisation criterion. There are many algorithms proposed to solve VRPTW problem. One of the first algorithm classifications was made by Solomon in (Solomon, 1987). Further papers (Cordeau et al., 2002) divide these algorithms into: heuristic approximation methods with upper limitations, heuristic methods with lower limitations, integer-based solutions. Authors of the paper (Tan et al., 2001) contains additions to the four algorithms: searching for a neighbourhood, simulated annealing, tabu search (He and Li, 2018), and genetic algorithms. The papers (Bräysy and Gendreau, 2005) present some metaheuristics.

The simulated annealing method has been classified as a meta-heuristic method with upper limitations (Afifi et al., 2013). Our simulated annealing algorithm (SATM) was developed based on the paper (Woch and Łebkowski, 2009). We have extended the algorithm by including several new elements: heterogeneous fleet, modification of searching initial solution, different methods of cost calculations, time-dependent network of travel, parallelization, and others. The algorithms have been tested on Solomon and Gehring-Homberger benchmarks (VRPTW, 2021), as well as based on real data from courier companies.

Our paper discusses two important problems that often occur when looking for a solution to a VRPTW problem: calibration of the parameters of the random solution search algorithm (in our case: simulated annealing, SATM) and construction of the time-dependent travel time matrix which is the basis for determining the transit time between two recipients in algorithms solving the VRPTW problem. Knowing the time of travel between two recipients makes it possible to calculate the time of travel to each recipient and determine whether we fit into a given (required) time window.

The paper is organized as follows. In section 2 we present learning method for constructing a model of the network of travel times. In section 2 we present learning method of our simulated annealing (SATM) algorithm for VRPTW problem solving which consist of an optimal selection of calibration parameters. Finally, we conclude results of our research and indicate problems for future works.

2. Modelling of the network of travel times

One of the subproblem in time-dependent VRPTW is the problem of constructing a network model of travel times, used to determine the travel times between recipients, depending on the time of departure (time-dependent network). The network of travel times is a set of locations to be visited (vertices of the network) together with the time of travel between them (edges of the network and their weights). The problem of constructing the network model of travel times can be considered as estimation the parameters of the road network based on GPS data from the monitored vehicles (mainly the travel time of the network sections). The problem can be divided into three phases: (1) collecting GPS data from vehicles (location, speed, azimuth, etc.); (2) "tying" data from travel to road sections (so-called map matching (Chen et al., 2014)); (3) estimating travel times for sections on the basis of data collected for the section on travel times (speeds) (we use proprietary solutions supported by the latest achievements in this area, described in (Bertsimas et al., 2019; Karurukan et al., 2018; Kim et al. 2021; Lan et al., 2019; Shi et al., 2017; Tan et al., 2001). In solving this problem, a number of specific problems arise (Chowdhury et al., 2017): processing of mass data (Big Data) from a large number of vehicles, for a large number of network sections; estimation of time parameters for each network section (network learning) or multidimensional classification of network sections or sub-areas forming virtual network vertices and estimation of parameters for the segment/edge category of directed graph modelling the network of recipients (the time of the section travel is a random variable whose distribution and parameters need to be estimated (Juhasz et al., 2017)); problems of efficiency and computational complexity of processing large and fast changing data sets and many others. Therefore, we try to verify the research hypothesis (one of the propositions was presented in (Fulton, 2015)) that the travel time between two points can be estimated with sufficient accuracy, depending on their geographical location and the time of departure (without referring to the micro-scale, i.e. the detailed structure of the road network). The research was carried out on two types of data: for Warsaw - from data of transport companies and one of the road traffic data providers acting in Internet. We have also conducted our research for New York (travel data from taxi company) and the results are very interesting (Development, 2020). The results of the research (learning based on regression models) were promising. They can be used in the next stage for improving the quality (accuracy) of the developed models and their learning methods.

2.1. Reduction method of number of time intervals

One of the objectives of the research was to reduce the number of time intervals (time windows) for which the network matrices of the travel times are built without too much loss of quality of travel time reasoning models. The travel time network matrix is built for a set of pairs (origin-destination) of points between which there is a need to estimate the travel time. This time depends on the month, day of the week and the moment of starting the travel from the point of origin of the pair of points, so for the given set of pairs (origin-destination) of points we must have as many matrices as there are moments of starting the travel (more precisely: time intervals of starting the travel, assuming that for each moment of starting the travel from the time interval, the time of the travel is identical between a fixed pair of points). If we divided the day into 15-minute intervals, we would have 4x24=96 times intervals, for which we would have to estimate the time matrix for each pair (origin-destination) of points, for separate weekday! The idea is to reduce the number of these time intervals. The aim of the research, already in detail, was to show in which time intervals of the travel start time, the travel times between a fixed pair of points are similar and in which they differ significantly. It was therefore a matter of grouping (clustering) the start time intervals due to similar travel times. Then, for each group of time intervals, one model can be built to estimate the travel time. This is important because, as shown in Table 2 the learning time of travel time network models increases exponentially with the increase in the groups of time intervals and the quality of the models (average percentage error of the travel time estimation) changes slightly.

There were several dozen (*m*) pairs of points (origin-destination) in Warsaw and for them Google Maps calculates the fastest driving times starting from the point of origin in 15-minute intervals, for the selected day (20.03.2019, Wednesday). Google Maps for routes calculated in the past or in the future gives the time interval of travel [*time_min*; *time_max*], so for each pair of starting points, a time interval has been obtained for each departure (start) time of the travel: [*time_min*; *time_max*]. Clustering for the *m* routes was calculated (at that time the vector had dimension m*2: $time\_min_i$ and $time\_max_i$, i=1,...m and there were 96 vectors). The k-means method was used to group the time windows. Euclidean distance was chosen as a measure of distance. Results of clustering are presented in Fig. 1.

From the Fig. 1. we see that with 4 clusters the whole day was divided into 4 periods:

1. night hours - from about 7:45 p.m. to 6:15 a.m.;
2. early morning hours (6:30 a.m. - 7:00 a.m.), mid-day hours (9:15 a.m. - 2:45 p.m.) and early evening hours (6:00 p.m. - 7:30 p.m.);
3. morning peak hours (7:15 a.m. - 9:00 a.m.);
4. afternoon peak hours (3:00 p.m. - 5:45 p.m.).

With 7 clusters (see Fig. 1.), optimal from k-means method, we can observe that 4 clusters have been divided in a characteristic way:

5. night hours have been narrowed down to 10:45 p.m. – 5:45 a.m.;
6. early morning hours (6:00 a.m. – 6:15 a.m.) and late evening hours (7:15 p.m. – 10:30 p.m.);
7. hours just before the morning peak (6:30 a.m. - 6:45 a.m.) and just after it and mid-day (9:45 a.m. - 2:00 p.m.), as well as just after the afternoon peak (6:30 p.m. - 7:00 p.m.);
8. early and late morning peak hours (7:00 a.m., 8:45 a.m. - 9:30 a.m.);
9. morning peak hours (7:15 a.m. - 8:30 a.m.);

10. hours just before the afternoon peak (2:15 p.m. - 3:00 p.m.) and just after it (5:30 p.m. - 6:15 p.m.);
11. the afternoon peak hours (3:15 p.m. - 5:15 p.m.).

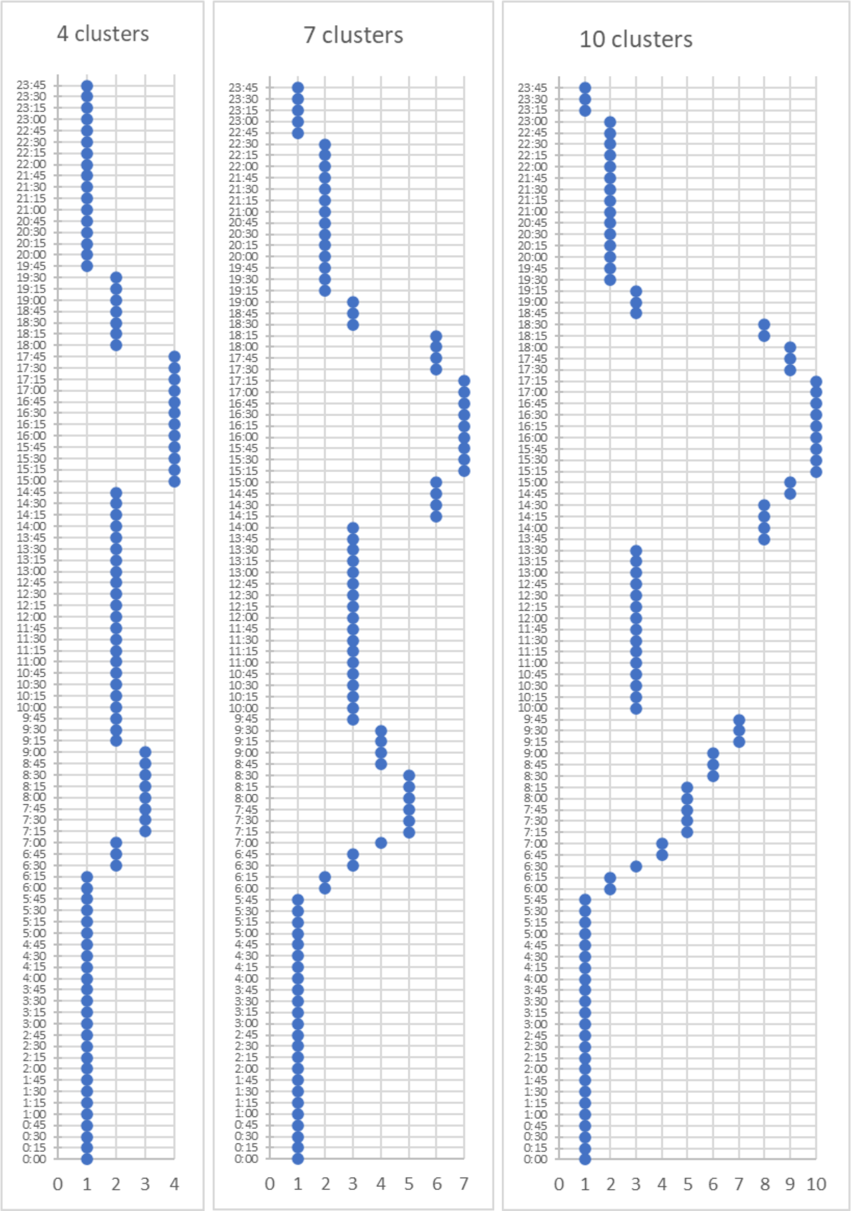


Fig. 1. Different number of clusters of the travel start time intervals in Warsaw (x-axis: cluster number (1,…,10), y-axis: start time of the 15-minute period for the start of the travel)

With 10 clusters, we can observe even more accurate extraction of characteristic daily periods of time due to the travel time in Warsaw in relation to 7 groups. Among other things, the hours of high morning peak (7:15 a.m. - 8:15 a.m.) and afternoon peak (3:15 a.m. - 5:15 p.m.) and night hours (11:15 p.m. - 5:45 a.m.) are clearly visible.

On the basis of the conducted research, it can be concluded that it is possible to group the time intervals within the day in such a way that groups (clusters) contain such time intervals, which are characterized by similar travel times for each moment of starting the travel from such a time interval.

Table 1. Different number of clusters (4, 7, 10, 14) of the travel start time intervals in Warsaw (three of them (4, 7, 10) are presented on Fig. 1)

Number of cluster	4 clusters of start times periods	7 clusters of start times periods	10 clusters of start times periods	14 clusters of start times periods
1	19:45-6:29	22:45-5:59	23:15-5:59	15:45-16:14
2	6:30-7:14 + 9:15-14:59 + 18:00-19:44	6:00-6:29 + 19:15-22:44	6:00-6:29 + 19:30-23:14	9:00-9:44
3	7:15-9:14	6:30-6:59 + 9:45-14:14 + 18:30-19:14	6:30-6:44 + 10:00-13:44 + 18:45-19:29	22:30-5:59
4	15:00-17:59	7:00-7:14 + 8:45-9:44	6:45-7:14	7:00-7:29
5		7:15-8:44	7:15-8:29	14:00-14:44 + 18:15-19:14
6		14:15-15:14 + 17:30-18:29	8:30-9:14	14:45-14:59 + 17:45-18:14
7		15:15-17:29	9:15-9:59	16:15-16:59
8			13:45-14:44 + 18:15-18:44	15:00-15:29
9			14:45-15:14 + 17:30-18:14	7:30-7:44
10			15:15-17:29	15:30-15:44 + 17:00-17:44
11				8:30-8:59
12				6:00+6:29 + 19:15-22:29
13				6:30-6:59 + 9:45-9:59 + 10:00-13:59
14				7:45-8:29

2.2. Assumptions for modelling network of travel times and performance evaluation

A model of simple linear regression was used to determine the relationship between the travel time and the geographical distance between the origin-destination points:

$$\hat{y} = alfa_0 + alfa_1 \cdot x \qquad (1)$$

where:
$\hat{y}$ – estimated travel time (explained variable) [min];
x – geographical distance between origin-destination (explanatory variable) [km];
alfa₀, *alfa₁* – model parameters.

As a measure of the regression error, mainly the Mean Average Absolute Error (MAE) was used, which is the average of the sum of the absolute differences between each pair of points: a given value of y (travel time) and the estimated value $\hat{y}$, that is, the travel time (counted in minutes) calculated on the basis of the regression model used (i.e. $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, where n is the number of data); MAPE (Mean Average Percentage Error) was also used ($MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i-\hat{y}_i|}{y_i} \cdot 100\%\right)$). Wherever possible, the determination factor R^2 was also calculated.

The study was conducted for several time windows (clusters) in the day (see Fig. 1.), for one day of the week (Wednesday), without division of Warsaw into zones (see Fig. 2a) and with division into zones

(Fig. 2b). There were over 270 000 of learning data (pairs of start (origin) and stop (destination) points randomly selected from Warsaw and its neighbourhoods (see Fig. 2), at different times of the day, but with a probability distribution depending on the time of day: more points at peak times, less points at off-peak times; average 17 pairs of origin-destination for each pair of zones from Fig. 2b). For these pairs of points, the fastest route was calculated using the Google Maps API (GoogleMaps, 2021). For each pair of origin-destination, for each moment of starting the travel a time interval of travel time of fastest route was obtained from Google Maps: [*time_min; time_max*]. We don't know probability distribution of this travel time, so we assume that this distribution is normal with expected value equals *avg_time*=0,5*(*time_min* + *time_max*) and standard deviations equals 1/6*(*time_max* - *time_min*). For further calculations we assume that travel time of fastest rout is equal *avg_time*.

2.3. Analysis of the learning results of the travel time network models

Based on the determined different groups (clusters) of time intervals (Table 1) and data on travel from one of the Internet providers (Google Maps), the learning of travel time network models for Warsaw and its neighbourhoods was carried out. The results of the comparison of the quality of the taught regression models (research strategies) with and without zoning are presented in Table 2 and there are very promising. The study strategies described in Table 2 mean respectively (strategy number - number of zones - number of clusters of time intervals): 1-1×1-1; 2-1×1-1; 3-1×1-1; 4-1×1-7; 5-1×1-14; 6-5×5-1; 7-5×5-4; 8-5×5-7; 9-5×5-14. On the basis of the data contained in Table 2, it can be observed that the time between two points can be estimated with sufficient accuracy depending on their geographical location (i.e. the geographical distance between them) and the time of departure (without reference to the micro-scale, i.e. the detailed structure of the road network). The model without teaching and without dividing Warsaw into zones and without considering the moment of leaving the starting zone gave an average absolute error of estimation (MAPE) of travel time of about 43%, which was unacceptable. The learning of the model consisted in the fact that for subsequent research strategies (the research strategy consisted in extending the model by 5×5 zoning (Fig. 2a) and selecting the number of groups (clusters) of the time interval for starting the travel), new parameters of the model were estimated in order to minimize the average absolute error of estimating the travel time ((5×5)×(5×5) ×Number_of_clusters (1 or 4 or 7 or 14) regression models were learned, i.e. for each pair of zones and for each group of start time intervals, a model for the travel time between these zones depending on the geographical distance between start and destination points and the start time interval was taught).

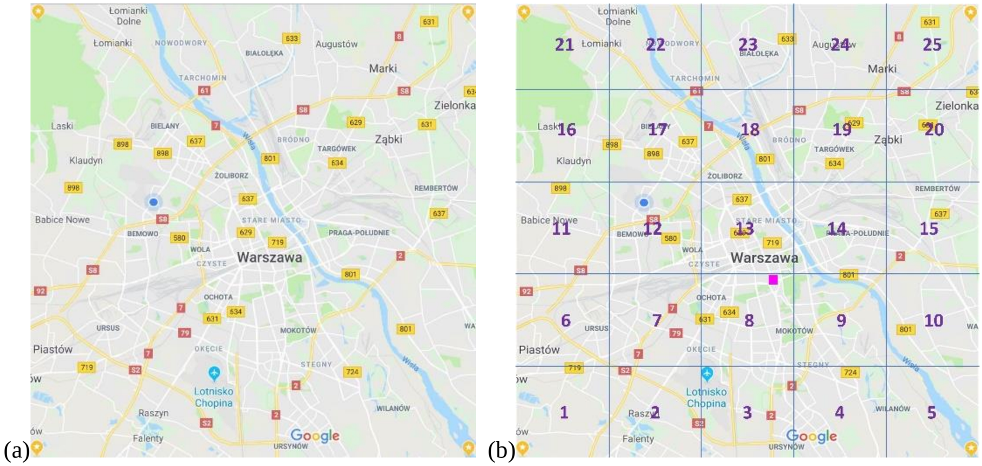


Fig. 2. (a) Area of Warsaw and its neighbourhoods (about 500 km^2) and (b) division into 5x5 square zones (length of the side of a single square about 4,5 km)

Table 2. Results of comparison of the quality of regression models (research strategies) with and without zoning

Number of strategy	Zone size	Number of clusters of time intervals	Weekday	Time of models learning [min]	MAE average	MAE median	MAE std. dev.	MAPE average	MAPE std. dev.	MAPE 75. percentile	MAPE 95. percentile
1	1×1	1 (1)	WED	0	16,25	14,41	10,73	43,33	18,20	43	70
2	1×1	1 (2)	WED	0	9,62	7,69	7,83	26,20	18,30	36	56
3	1×1	1 (3)	WED	0,12	8,64	7,18	6,63	29,47	37,29	36	82
4	1×1	7	WED	1,5	7,09	5,67	5,87	24,39	35,77	29	69
5	1×1	14	WED	1,3	7,11	5,69	5,88	24,45	35,90	29	70
6	5×5	1	WED	20	7,3	5,8	5,96	23,21	24,44	29	68
7	5×5	4	WED	84	4,77	3,66	4,24	15,62	11,11	20	44
8	5×5	7	WED	420	4,37	3,32	3,98	14,50	19,28	18	41
9	5×5	14	WED	2640	3,87	3,02	3,36	13,10	18,32	16	37

(1) model without learning, average speed of 36 km/h (data from Warsaw transport services), $\hat{y} = \frac{60}{36} \cdot x$;

(2) the model of dependence of route length on geographical distance[1] and an average speed of 36 km/h (data from Warsaw transport services), $\hat{y} = \frac{60}{36} \cdot (1{,}66 + 1{,}36 \cdot x)$;

(3) estimated model without zoning and without clusters of time intervals, $\hat{y} = 10{,}96 + 2{,}08 \cdot x$.

From Table 2 also result dependence between average absolute error of travel time models and the geographical distance between the starting and stopping point of the travel in Warsaw and its surroundings and considering the research strategy. The best results were achieved with division into 5×5 zones and with 14 clusters of travel start time intervals (MAPE=13% error), but already for 7 clusters of travel start time intervals (as in Table 1) satisfactory results were achieved (MAPE=14.5%) and the learning time of the model was 6 times shorter (about 7 hours instead of 44). This meant that the average percentage error in estimating the travel time between any pair of points in Warsaw and its vicinity, depending on the geographical distance between them, was 14.5%. Moreover, 75% of estimation errors were less than 18% and 95% of estimation errors were less than 41%.

The model was verified based on 300 data of real travels (Uber data) in Warsaw from the period IX.2016-III.2020 and we obtain MAPE=23% (too little data and in addition only between selected zone pairs). We can iteratively improve parameters of the models by solving some optimization problem. After solving the problem we reduce MAE error from 6.95 to 3.7 and MAPE from 23% to 13.7%.

3. Learning of simulated annealing (SATM) algorithm for VRPTW problem solving

3.1. General idea of SATM algorithm

Our simulated annealing algorithm (SATM) was developed based on the paper (Woch and Łebkowski, 2009). We have extended the algorithm by including several new elements: heterogeneous fleet, modification of searching initial solution, different methods of cost calculations, time-dependent network of travel, parallelization, and others. In details SATM algorithm is described in (Development, 2020; Kulas and Tarapata; 2021), but in this paper a proposal for a method of optimal selection of calibration parameters is the most important.

The initial step is to find an initial solution (acceptable of course). It is advisable that this should be a fairly good solution and quickly obtained. Typically,

[1] An interesting, additional observation from the research is that the average length (in km) of the shortest time route in Warsaw (despite roads of different categories and with different speed limits) between any pair of points start/origin and stop/destination is strongly correlated (the correlation coefficient is about 0.91) with the geographical distance between these points and is expressed by the formula: length [km] = 1,66+1,36 Geogr_distance [km].

a modified Solomon algorithm is used (Solomon, 1987). Then the search for the global extremes begins again. Initially, the algorithm tries to search the whole space of acceptable solutions trying to hit around the global extreme. During the calculations, the tendency of the algorithm to leave the local extremes (so-called system temperature decrease - annealing) is gradually reduced, if the longer the calculation takes, the more likely the current local extremes are to be global as well.

For each fixed temperature, repeat the search for the best solution by: determining an acceptable "adjacent" solution and performing a simple post-optimisation of the routes (this is one more our extension of the standard algorithm). If the new solution is better, then it is acceptable. If it is worse, it is assumed with probability depending on the distance from the best solution so far and the degree of annealing (the higher the distance and the lower the temperature, the lower the probability).

Finally, the conditions for completing the calculation for the fixed temperature and for annealing are checked (see Table 3).

Table 3. Pseudo-code of the SATM algorithm

```
determine initial solution x0;
specify the initial temperature t = TZERO;
for (epoch=1; epoch < EPOCH; epoch++) {
    for (iteration=1; iteration < ITER; iteration++) {
        determine solution x from neighborhood x0;
        d = F(x) - F(x0);
        if ( d < 0 ) {
            x0 = x; }
        else {
            draw the number b from the range [0.1];
            if ( b < exp(-d / t) ) {
                x0 = x; } } }
    t = A( t ); }
```

The key issues for the algorithm are: determination of the initial solution, the strategy for finding an adjacent solution and the annealing strategy. The key calibration parameters, used for example in the paper (Woch and Łebkowski, 2009), are the initial TZERO temperature, the type and parameters of the annealing function A(t) (LOG), the number of the epochs (EPOCH) and the number ITER of the iterations (list of all calibration parameters is presented in Table 4, parameters with id from 5 to 17 are additionally proposed by us in our SATM algorithm). Some recommendation to set initial values of calibration parameters are presented in (Development, 2020; Woch and Łebkowski, 2009) and in the 4th column in Table 4.

The determination of the first (initial) solution is based on the proposal in (Woch and Łebkowski, 2009). The parallelization of the SATM algorithm was developed by considering the conclusions from the paper (Czech and Czarnas, 2002) and research from the paper (Wieczorek, 2011).

3.2. A method to optimise the selection of calibration parameters

In our research we use Gehring-Homberger benchmarks (VRPTW, 2021) and we define input data characteristics for VRPTW problem to classify the type of data, see Table 5. Each file of input data has set of deliveries (customers), their geo-coordinates (to calculate distance between them), demands and service time window. Values of input data characteristics for a few exemplified Gehring-Homberger benchmarks are presented in Table 6. Number of deliveries #DELIV = #WIND.

We define type of input data and describe it by *xyz*, where:

x - degree of clusterization of deliveries,

$$x = \begin{cases} 0, \text{when } \#\text{CLUST} = 1 \text{ or } \text{STD} > 3 \cdot \text{AVG} \\ 1, \text{when } 5 \geq \#\text{CLUST} > 1 \text{ and } \text{STD} \leq 3 \cdot \text{AVG} \\ 2, \text{when } \#\text{CLUST} > 5 \text{ and } \text{STD} \leq 3 \cdot \text{AVG} \end{cases} \quad (2)$$

y – relation between number of short and long-time windows,

$$y = \begin{cases} 0, \text{when } \#\text{WIND}/2 + \#\text{WIND}/3 + \#\text{WIND}/4 \\ \quad + \#\text{WIND}/5 + \#\text{WIND}/10 > 0.5 \cdot \#\text{WIND} \\ 1, \text{otherwise} \end{cases} \quad (3)$$

z - relation between number of medium length time windows and very short time windows,

$$z = \begin{cases} 0, \text{when } \#\text{WIND}/2 + \#\text{WIND}/3 + \#\text{WIND}/4 \\ \qquad \leq \#\text{WIND}/5 + \#\text{WIND}/10 \\ 1, \text{otherwise} \end{cases} \quad (4)$$

For example, 201 describes type of input data with strong clusterization of deliveries (2), the advantage of short service windows over long ones (0) and the advantage of medium length time windows over very short time windows (1).

Criteria for assessing the quality of a VRPTW solution using SATM algorithm are as follows: c_1 – total length of routes from the calculation; c_2 – calculation time in seconds; c_3 – the number of the epoch of calculation that already gave the best result; c_4 – number of routes from the calculation.

We use following notations:

N – set of input data id for VRPTW problem, $N=\{1, 2, \ldots, n\}$;

S_tid – set of input data types, $S\_tid = \{xyz: x \in \{0,1,2\}, y \in \{0,1\}, z \in \{0,1\}\}$, xyz calculated using (2)-(4);

f_tid – a function that returns type of input data id, f_tid: $N \rightarrow S\_tid$;

M – set of identifiers for sets of calibration parameters, $M=\{1, 2, \ldots, m\}$;

$CP = \{cp_1, cp_2, \ldots, cp_m\}$– tested sets of calibration parameters, where $cp_j = \langle cp_{j1}, cp_{j2}, \ldots, cp_{j17} \rangle$, $j \in M$ describes vector of values of calibration parameters from Table 4;

$c_1(i,j)$ – value of total length of routes from the calculation for the i-th input data using the j-the set of calibration parameters, $i \in N$, $j \in M$;

$c_2(i,j)$ – value of calculation time in seconds for the i-th input data using the j-the set of calibration parameters, $i \in N$, $j \in M$;

$c_3(i,j)$ – value of the number of the epoch of calculation that already gave the best result for the i-th input data using the j-the set of calibration parameters, $i \in N$, $j \in M$;

$c_4(i,j)$ – value of number of routes from the calculation for the i-th input data using the j-the set of calibration parameters, $i \in N$, $j \in M$;

Table 4. Type of calibration parameters in SATM algorithm and their interpretation

Calibration parameter id	Name	Interpretation	Values for calibration tests
1	EPOCH	Number of epochs (fixed for one thread, with more threads should be increased)	100, 200, 400, 600
2	ITER	Length of the epoch - repetition of the search in one epoch (should increase with the size of the task - exponentially / polynomially)	200, 500
3	TZERO	Initial system temperature (determines the initial search of a solution space)	80, 100, 150, 200
4	LOG	Type of the annealing function: 0 - geometric, 1 - logarithmic (practically only the geometric function is used)	0
5	TAU	Weighting factor of the criterion for the total duration of deliveries	0
6	ALFA	Weighting factor of the criterion for a given number of routes	10000
7	BETA	Weighting factor of the criterion for the total length of routes	1
8	GAMMA	Cost (penalty) for using a vehicle exceeding the limit	1000000
9	ETA	Cost (penalty) for assigning a customer to a vehicle exceeding the limit	1000
10	DELTA	The factor of the system's temperature change function (determines the speed to narrow the search area)	0.965
11	MT_CLIENT	Number of troublesome customers initially considered when determining a simple initial solution	5
12	LT_CLIENT	Number of least troublesome customers initially considered when determining a simple initial solution	2
13	CNUMBER	Some of the customers changed while looking for a neighbour's solution	0.125
14	RADIUS	Step of increasing the radius of the circle, in which the neighbours of a solution are looked for	10, 20, 40, 100, 1000
15	PROBAB	Probability of drawing a customer from the first district	0.2, 0.3
16	MIN	Adopting at the end of the epoch the best of the existing solutions - minimal (MIN != 0)	1
17	THREADS	Number of calculation threads (for parallelization)	5

Table 5. Input data characteristics for VRPTW problem

Input data characteristics id	Characteristics name	Description
0	#DELIV	Number of deliveries (customers, clients)
1	SUM	Sum of distances between all pairs of deliveries
2	MAX	Maximum distance between pair of deliveries
3	AVG	Average distance between all pairs of deliveries
4	STD	Standard deviations of distance between all pairs of deliveries
5	#INCLUST	Number of clients in clusters
6	#CLUST	Number of clusters
7	#WIND	Total number of defined time windows for deliveries
8	#WIND/2	Number of deliveries with a time window from 1/3 to 1/2 of the warehouse opening time
9	#WIND /3	Number of deliveries with a time window from 1/4 to 1/3 of the warehouse opening time
10	#WIND /4	Number of deliveries with a time window from 1/5 to 1/4 of the warehouse opening time
11	#WIND /5	Number of deliveries with a time window from 1/10 to 1/5 of the warehouse opening time
12	#WIND/10	Number of deliveries with a time window not greater than 1/10 of the warehouse opening time

Table 6. Input data characteristics for selected Gehring-Homberger benchmarks

Input data file (benchmark)	SUM	MAX	AVG	STD	#IN CLUST	#CLUST	#WIND	#WIND /2	#WIND /3	#WIND /4	#WIND /5	#WIND /10
C1_210	14207	190	75	4	160	16	200	124	64	8	3	0
C2_4_8	34388	228	88	33	345	3	400	0	0	0	400	0
R1_6_7	91154	405	151	52	521	4	600	0	0	0	0	300
R2_2_4	15037	184	72	53	184	1	200	0	0	0	0	50
RC1_6_4	87542	398	146	44	529	5	600	0	0	0	0	150
RC2_610	87542	398	146	105	537	1	600	0	0	0	600	0

Because range of the values of criteria may be different (for example a typical value of c_4 is up to several dozen and typical value of c_1 is up to several tens of thousands), we calculate normalized $\overline{c}_k$values of criteria c_k, k=1,…,4:

$$\overline{c}_k(i,j) = \frac{c_k^{\max}(i) - c_k(i,j)}{c_k^{\max}(i) - c_k^{\min}(i)} \in [0,1] \qquad (5)$$

where:

$$c_k^{\max}(i) = \max_{j \in M} c_k(i,j) \qquad (6)$$

$$c_k^{\min}(i) = \min_{j \in M} c_k(i,j) \qquad (7)$$

Because optimal value of $\overline{c}_k(i,j)$ is equal to 1, for each k=1,…,4, so we can define mc function as Euclidean distance between optimal and current values of criteria:

$$mc(i,j) = \sqrt{\sum_{k=1}^{4}\left(1 - \overline{c}_k(i,j)\right)^2},\ i \in N, j \in M \qquad (8)$$

We define a few strategies to recommend a set SCP_i of calibration parameters for the i-th input data:

1. Strategy S_1: to determine such a set $SCP_i = cp_{j^*} \in CP$ of calibration parameters, that:

$$mc(i, j^*) = \min_{j \in M} mc(i,j) \qquad (9)$$

2. Strategy S_2: to determine such a subset $SCP_i \subset CP$, that:

$$\forall_{scp \in SCP_i} \; mc(i, scp) \le 1.1 \cdot mc(i, j^*) \tag{10}$$

3. Strategy S_3: to determine such a subset $SCP_i \subset CP$, that:

$$\forall_{scp \in SCP_i} \forall_{k=1,\ldots,4} \; \bar{c}_k(i, scp) \ge 0.9 \tag{11}$$

Strategy S_1, for the i-th input data, allow us to select such a set SCP_i of calibration parameters for which the value of mc function is minimal, using strategy S_2 we select such a subset SCP_i of calibration parameters that the value of mc function is not less than 10% lower than the best value, using strategy S_3 we determine such a subset SCP_i of calibration parameters that for each criterion the normalized value of this criterion is not less than 90% its optimal value (let's recall that after normalisation, the optimal value of the criterion is equal to). Let us note that we solve the problem of optimizing the multi-criteria selection of a set of calibration parameters and the presented strategies are methods to solve this problem.

Considering presented explanations, procedure of recommendation a set of calibration parameters for the input data types is presented in Table 7.

Interpretation: the a_{pr} means the number of input data which are of type r and for which the set p of calibration parameters was recommended. Interpretation of the b_{pr}*100% can be as follows: this is a percent of input data of the r-th type for which the p-th set of calibration parameters gave the best results. Another interpretation: when we solve VRPTW problem using SATM algorithm and we have input data of type r, we should use the set p of calibration parameters with frequency ("probability") equal to b_{pr} in order to obtain the best solution (depend on strategy S) from the point of view of many criteria c_1,.., c_4 .

3.3. Analysis of results

We test n=36 Gehring-Homberger benchmarks (for #DELIV: 200, 400 and 600) for different sets of calibration parameters (m=160, see Table 4, column 4), for THREADS=5 (we have conducted calculations also for different numbers of threads but it is not described in this paper). Criteria c_3 and c_4 was calculated but were not considered in this analysis (i.e. $c_3(i,j)=c_4(i,j)=0$, $i \in N$, $j \in M$). All calculations have been conducted on computer with processor Intel® Core™ i7-9700 (8 cores) and 16 GB RAM (relevant for calculation time).

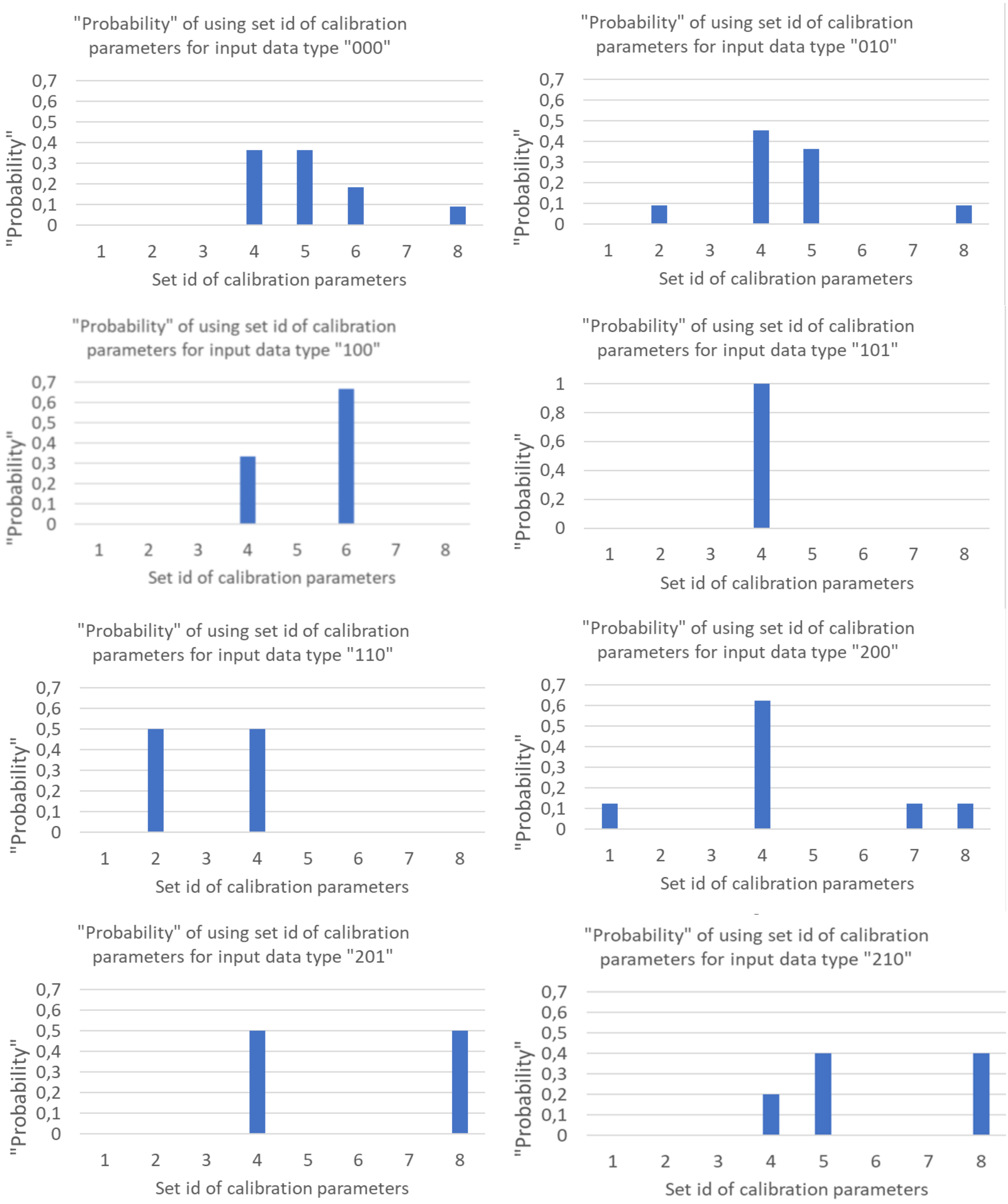
In Table 8 the best sets SCP of calibration parameters values has been presented.

Table 7. Procedure of recommendation a (best) set of calibration parameters for the input data types

1. Run SATM algorithm for each input data $i \in N$, for each set of calibration parameters $j \in M$;
2. Choose a strategy $S \in \{S_1, S_2, S_3\}$ and determine $SCP_i \subset CP$ using formulas (9) or (10), or (11) (depend on selected strategy S) for each input data $i \in N$;
3. Determine set SCP of the best (recommended) set of calibration parameters as follows: $SCP = \bigcup_{i \in N} SCP_i$;
4. Classify each input data $i \in N$ calculating $f\_tid\,(i)$ using formulas (2)-(4);
5. Determine matrix $A = [a_{pr}]_{\overline{\overline{SCP}} \times \overline{\overline{S\_tid}}}$, where $a_{pr} = \overline{\overline{\{i \in N: p \in SCP_i \wedge f\_tid(i) = r\}}}$ and $\overline{\overline{X}}$ describes cardinality of set X.
6. Determine matrix $B = [b_{pr}]_{\overline{\overline{SCP}} \times \overline{\overline{S\_tid}}}$, where $b_{pr} = \frac{a_{pr}}{\sum_{q \in SCP} a_{qr}} \in [0,1]$.

Table 8. The best sets SCP of calibration parameters values

Calibr. param. (CP) sets id	EPOCH	ITER	TZERO	RADIUS
1	100	200	100	20
2	200	200	80	20
3	200	200	100	10
4	200	200	100	20
5	200	200	100	40
6	200	200	100	100
7	200	200	200	20
8	200	500	100	20


"Probability" of using set id of calibration parameters for input data type "000"

"Probability" of using set id of calibration parameters for input data type "010"

"Probability" of using set id of calibration parameters for input data type "100"

"Probability" of using set id of calibration parameters for input data type "101"

"Probability" of using set id of calibration parameters for input data type "110"

"Probability" of using set id of calibration parameters for input data type "200"

"Probability" of using set id of calibration parameters for input data type "201"

"Probability" of using set id of calibration parameters for input data type "210"

Fig. 3. "Probability" distribution of using set id of calibration parameters from Table 8 for different input data type ("xyz") in order to obtain the best solution for two c_1, c_2 criteria (strategy S_1)

Table 9. Matrices B for strategies S_2 and S_3

CP sets id	Strategy 2 (values for input data type "xyz")								Strategy 3 (values for input data type "xyz")							
	000	**010**	**100**	**101**	**110**	**200**	**201**	**210**	**000**	**010**	**100**	**101**	**110**	**200**	**201**	**210**
1	0	0	0	0	0	0,11	0	0	0	0	0	0	0	0	0	0
2	0	0,07	0	0	0,50	0,11	0,08	0	0	0	0	0	0	0,06	0	0
3	0	0	0	0	0	0	0,46	0	0	0	0	0	0	0	0	0
4	0,35	0,50	0,14	1	0,50	0,56	0,38	0,33	0,55	0,75	1	1	1	0,61	1	0,50
5	0,50	0,29	0,29	0	0	0	0	0,33	0,18	0	0	0	0	0	0	0
6	0,10	0	0,57	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0,11	0	0	0	0	0	0	0	0,17	0	0
8	0,05	0,14	0	0	0	0,11	0,08	0,33	0,27	0,25	0	0	0	0,17	0	0,50

In Fig. 3 and Table 9 we present values or visualization of calculated matrices *B* for strategies S_1, S_2 and S_3. From this analysis results that the best solutions have been obtained for the 4th and 5th calibration parameters set id. For example (see Table 9), in strategy 2, for "000" input data type we have obtained the best results for solving VRPTW problem for (0,35 + 0,50)*100%=85% input data using the 4th and 5th calibration parameters set id from Table 8. In turn, calibration parameters set id=1 for the smallest number of input data gave the best results. Very interesting result for strategy 3 has been obtained: for subset *IDT*={100, 101, 110, 201} of input data types the best set of calibration parameters is only set with id=4. It means that for 100% of input data of the type belonging to *IDT* subset we obtain single set of calibration parameters for which the normalized value of this criterion c_1 (total length of routes) and c_2 (calculation time) is not less than 90% its optimal value.

4. Conclusions

The article discusses two problems that often occur when looking for a solution to a VRPTW problem: calibration of the parameters of the random solution search algorithm (in our case: simulated annealing, SATM) and construction of the time-dependent travel time matrix which is the basis for determining the transit time between two recipients in algorithms solving the VRPTW problem.

Because the SATM algorithm (like whole group of randomized algorithms) is very sensitive to the values of the calibration parameters (especially when it comes to calculation time, but also to the quality of the routes to be mapped), the next research step should be the typing (grouping) of calibration parameters and performing many more tests (much more learning sets). As we did in the case of the definition of input data types, the further aim of the research will be to divide the sets of calibration parameters into their types and ultimately - automatic selection of the type of calibration parameters when starting the algorithm based on the input data type.

A very interesting problem is the selection of calibration parameters of the algorithm to obtain the best possible solution after a maximum set time, for example 5 minutes. This is very important from a practical point of view because we want to have a solution acceptable from the point of view of the desired criteria quickly. It can be the next step for research.

The advantage of our proposal to construct a network of travel time matrix is that to estimate the travel time between a pair of customers we only need their geographical locations and the moment they leave the warehouse. Having once learnt the regression models of the dependence of travel times on the distance in a given area (e.g. Warsaw), we can use them many times for different groups of customers from this area. Normally, in case of a new set of customers to be visited, we must recalculate/receive the time matrix between each pair of them each time. We are not able to predict in advance for which customer pairs we will need these travel times next time.

Moreover, relating to the construction of the network of travel times we conclude that zoning has a positive impact on the quality of travel time estimates based on distance and departure time. The results of the additional research (Development, 2020) allowed to conclude that the absolute error of estimation for zones 6x6 in relation to zones 5x5 decreased by about 15%. Generally, the conclusion is that the denser the road infrastructure (e.g. large cities), the smaller should be the size of a single zone (square), the rarer this structure (e.g. out-of-town areas), the bigger should be the size of the square (multi-resolution model, e.g. quadtree).

The number of clusters in the time interval of the start of the trip has also an influence on the quality of the estimation of the time interval network models; in the conducted studies the data on trips from one day of the week (Wednesday) were used. Other days of the week and months should be included in “clustering". In this way we would achieve full year-round coverage and the model would be even more accurate.

Moreover, reliable learning data (travel times between points, depending on start time of travel) for construction of the network of travel times should be given. We had data for real travels from several courier companies, but these data were unreliable (too many so-called unusual observations, difficult to filter out which have influence on built regression models), so we used GoogleMaps data. As a result of the project described in (Development, 2020; Tarapata et al., 2020) we have also obtained Smart Tracking Device (IUM) and based on historical data derived from routes and IUMs installed in many vehicles we plan to learn the network of travel times anew.

The presented suggestions and comments should be a basis for further research.

Acknowledgement

The study funded by the National Centre for Research and Development (NCBiR) in Poland under R&D project no. POIR.01.01.01-00-1146/17 - “Development and research into new methods and techniques for optimization of processes supporting the distribution of goods and services” and University Research Grant of the Institute of Computer and Information Systems of the Faculty of Cybernetics at the Military University of Technology, Warsaw (Poland).

References

[1] Afifi, S., Dang, D-C., Moukrim, A., (2013). A Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows and Synchronization Constraints. *7th International Conference, Learning and Intelligent Optimization* (LION 7), Catania, Italy, 259-265.

[2] Bertsimas, D., Delarue, A., Jaillet, P., Martin, S., (2019). Travel Time Estimation in the Age of Big Data, *Operations Research*, vol.67 (2), 498-515.

[3] Bräysy, O., Gendreau, M., (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms, *Transportation Science*, 39(1), 104–118.

[4] Chen, B.Y., Li, Q., Lam, W., Shaw, S-L., Yan, K., (2014). Map-matching algorithm for large-scale low-frequency floating car data, *International Journal of Geographical Information Science*, 28(1), 22-38.

[5] Chowdhury, M., Apon, A., Dey, K., (2017). *Data Analytics for Intelligent Transportation Systems*, Elsevier, Amsterdam.

[6] Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M. M., Soumis, F., (2002). The VRP with time windows, in: P. Toth D. Vigo (eds.), The Vehicle Routing Problem, *SIAM Monographs on Discrete Mathematics and Applications*, SIAM, 157-194.

[7] Czech, Z., Czarnas, P., (2002). Parallel simulated annealing for the vehicle routing problem with time windows, in *Proceedings 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, Canary Islands, Spain.

[8] Development and research of new methods and techniques for optimization of goods and services distribution processes - selected research problems and results obtained during the project (in Polish: Opracowanie i badanie własności nowych metod i technik optymalizacji procesów dystrybucji towarów i usług - wybrane problemy badawcze i wyniki uzyskane w trakcie projektu), Technical Report of the project POIR.01.01.01-00-1146/17, Track-Mate, Warsaw, 2020 (in Polish).

[9] Fulton, R., (2015). Estimating Delivery Times. *A Case Study In Practical Machine Learning*, https://blog.postmates.com/estimating-delivery-times-a-case-study-in-practicalmachine-learning-e70f677e736c.

[10] *GoogleMaps API Website* (2021): https://developers.google.com/maps/documentation/directions/, last accessed 2021/05/10.

[11] He, P., Li, J., (2018). Vehicle routing problem with partly simultaneous pickup and delivery for the cluster of small and medium enterprises, *Archives of Transport*, 45(1), 35-42.

[12] Juhász, M., Mátrai, T., Koren, C., (2017). Forecasting travel time reliability in urban road transport, *Archives of Transport*, 43 (3), 53-67.

[13] Karurukan, V.I., Nugraha, I., Kumiawan, N., (2018). The Estimation Model of Travel Time Between Point of Interest for Tourist Travel Recommendations with Regression Analysis, in *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, DOI:10.1109/ICITSI.2018.8696070.

[14] Kim, S., Yu, H., Yeo, H., (2021). A Study on Travel Time Estimation of Diverging Traffic Stream on Highways Based on Timestamp Data, *Journal of Advanced Transportation*, 2021, 1-13.

[15] Kulas, W., Tarapata, Z., (2021). Simulated Annealing Algorithm for Solving Modified VRPTW Problem in Time-Dependent Networks: A Case Study, *Proceedings of the 37th International Business Information Management Association Conference* (IBIMA), 30-31 May, Cordoba, Spain, ISBN: 978-0-9998551-6-4, ISSN: 2767-9640, 7351-7360.

[16] Lan, W., Xu, Y., Zhao, B., (2019). Travel Time Estimation without Road Networks: An Urban Morphological Layout Representation Approach, *IJCAI*, 2019, 1772-1778.

[17] Shi, Ch., Chen, B.Y., Li, Q., (2017). Estimation of Travel Time Distributions in Urban Road Networks Using Low-Frequency Floating Car Data, *ISPRS International Journal of Geo-Information*, 6, 253.

[18] Solomon, M. M., (1987). Algorithms for the Vehicle Routing and Scheduling with Time Windows Constraints, *Operating Research*, 35(2), 254-265.

[19] Tan, K.C., Lee, L.H., Zhu, Q. L., Ou, K., (2001). Heuristic methods for vehicle routing problem with time windows, *Artificial Intelligence in Engineering*, 15, 281-295.

[20] Tarapata, Z., Nowicki, T., Antkiewicz, R., Dudziński, J., Janik, K. (2020). Data-Driven Machine Learning System for Optimization of Processes Supporting the Distribution of Goods and Services – a case study, *Procedia Manufacturing*, 44C, 60-67.

[21] Tarapata, Z., Antkiewicz, R. (2021). Selected Distribution Network Optimisation Problems and Solving Methods, *Proceedings of the 37th International Business Information Management Association Conference* (IBIMA), 30-31 May, Cordoba, Spain, ISBN: 978-0-9998551-6-4, ISSN: 2767-9640, 4165-4173.

[22] Vidal, T., Marinelli, R., Pham, T.A., Ha, M.H., (2020). Arc Routing with Time-Dependent Travel Times and Paths, Technical Report, PUC-Rio, 29 April.

[23] VRPTW benchmarks files: https://www.sintef.no/projectweb/top/vrptw/, last accessed 2021/05/10.

[24] Wieczorek, B., (2011). Parallel Independent Simulated Annealing Searches to Solve the VRPTW. In: Czachórski T., Kozielski S., Stańczyk U. (eds) *Man-Machine Interactions 2. Advances in Intelligent and Soft Computing*, 103, Springer, Berlin, Heidelberg.

[25] Woch, M., Łebkowski, P., (2009). Sequential Simulated Annealing for the Vehicle Routing Problem with Time Windows, *Decision Making in Manufacturing and Services*, 3(1/2), 87-100.