Multi-Layer Data Model for Transportation Logistics Solutions

Anton Ivaschenko, Sergey Maslennikov
Samara State Technical University
Samara, Russia
{anton.ivashenko, yetisergey}@gmail.com

Anastasia Stolbova, Oleg Golovnin
Samara National Research University
Samara, Russia
{stolbova, golovnin}@ssau.ru

***Abstract*—The paper presents an original multi-layer data model for software solutions to be used in transportation logistics. Based on an analysis of the conceptual model it is proposed to decompose the data into layers. As a basic decomposition approach, it is recommended to consider processing capacity characteristics and performance measures instead of objects and subjects classification typical for human perception. The proposed model allows increasing the efficiency of parallel computing algorithms implementation, specific for Big Data analysis. Two algorithms were proposed based on the method of criteria comparison. The sequential algorithm implements the classical approach to find a path on a graph, while the parallel one uses an approach that makes it possible to increase the efficiency of layer-by-layer task separation, taking into account the capabilities of a computing system for simultaneous parallel calculations. Experimental results prove the necessity to introduce original data structures for parallel processing of this data. The approach is implemented using Apache Spark and Stream API. A study conducted on real data on 21 settlements in the Netherlands showed an advantage in the execution time of the parallel computing algorithm over the usual sequential search of 38%.**

I. Introduction

Automated resource optimization using modern information and intelligent technologies have become an essential service for transportation utilities nowadays. Modern logistics companies introduce new solutions for scheduling, allocation, tracking, and management of transportation resources in order to reduce costs and provide the best service. These solutions are based on the active use of geographic information systems (GIS) and specialized software for data processing and visualization. Most services are accessible online.

The trend is concerned with a new approach to building the whole business of a transportation company on the basis of an IT solution. It becomes the main and integral part of the management system and requires reliable and sustainable access to the data. Such an opportunity allows considerable business expansion but makes it impossible to function independently without the support of information technologies.

Due to the growth of the popularity of IT systems for transportations logistics and extension of the requirements for complexity, efficiency, and performance of the implemented algorithms, modern services need to process large volumes of information. Transportation logistics remains one of the main areas of application of Big Data technologies in practice.

At the same time, the existing data models do not consider this factor. GIS and logistics databases are built according to the relational theory. Information about the roads, routes, signs, and other traffic descriptors as well as the data of orders and transportation instructions, cargo shippers and receivers, drivers, tracks and trailers are put into a solid database. Processing this data takes time and requires optimization.

To solve this problem, we present in this paper a new multi-layer data model. The main idea is to modify the approach to decompose the knowledge of the transportation logistics problem domain. Instead of its division to conceptual groups of objects (like separate data spaces for roads, drivers and customers), it is proposed to introduce data layers optimized from the processing capacity characteristics and performance measures. More details are provided below.

II. Motivation

Modern transportation logistics software generates and processes large volumes of data that are habitually measured by Exabytes. The real transport system is influenced by many factors, such as the traffic intensity of the vehicles, weather conditions, social events, that require consideration in decision-making support. Despite the growing computing capabilities of computers, their processing takes a long time [1].

Big data technologies [2] can help to solve this problem and provide productivity measures values acceptable for constant use. Due to comparatively easy separation of the transportation logistics objects and subjects across the electronic map, the algorithms of parallel computing become suitable for implementation.

Let us consider the process of long-distance delivery of goods as an important problem of the transportation logistics industry that requires Big Data algorithms application. The main task its constant optimization of costs for the efficient movement of goods [3, 4]. It is necessary to plan the movement of trucks, respond in a timely manner to changes in the functioning of the transport system, and optimize resource utilization. The problem of Big Data processing failures in the transport system leads to the inability to process operational information for route planning in an acceptable time [5].

This is specifically trending for transportation logistics online platforms that provide intermediary services for customers, shippers, and carriers. For example, 5 party logistics

(5PL) companies can hold no own transport resources but involve external drivers, tracks and trailers to develop the best consolidations and provide allocation services [6], [7]. For such business, information plays a general role.

The aim of the work is to increase the speed of searching for optimal routes of vehicles when performing long-distance delivery of goods by dividing factors describing the state of the transport system into separate graph structures, which allow taking advantage of computing parallelization.

The simple solution to the problem of optimizing the movement of vehicles is the analysis and enumeration of many options for system states [8]. These algorithms assume that the data structure is small enough to fit in the computer's RAM. However, real-world data during the operation of the transport system grow both in volume and complexity, which leads to multiple iterations when executing exhaustive algorithms. It almost always takes a long time to complete [9]. The operational calculation also takes a long time, thereby leading to risks and increasing the cost of decisions [10]. Maintaining dedicated servers for calculating the optimal traffic becomes expensive and not practical [11].

Solutions for Big Data processing in the transportation industry are studied in [12–14]. Most of them are based on the implementation of the MapReduce approach that presents good results of automatic data parallelization [15–17]. Nevertheless, its practical use requires special preparation of the input data and the subsequent combination of the output [18]. Engineers and researchers have proposed some algorithms for solving this problem for various problem domains to process the data with different structures and dimensions.

For example, in [19] the stages of Big Data pre-processing were considered in order to obtain from them the most important information. In [20], a mining algorithm was proposed that uses an iterative graph structure to represent data. The paper [21] presents the advantages of a combination of the tools for preparing, processing, and integrating Big Data with visualization on an electronic map.

In this paper, we propose an approach to data organization that uses graphs to represent information about the transport system. Each cluster of information is located on a separate model layer, while the layers are developing both in space and in time. From a practical point of view, such a link to space-time is due to the wide possibilities of using electronic maps, geographic information systems and monitoring services of vehicles as sources of timely and accurate information. As part of the algorithms implemented on the graphs, the method of criteria-based comparisons is used to obtain the possibility of combining optimal routes [22].

III. Transport System Concepts

Transportation operator company network is a large system of interconnected objects that can interact and significantly affect each other. The following dependency diagram of the main objects and subjects of the transport system is presented in Fig. 1.

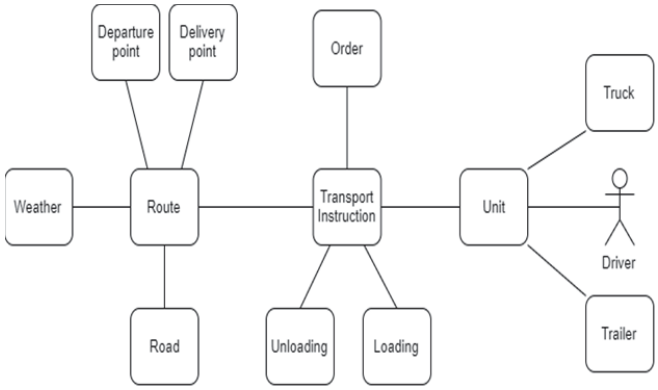


Fig. 1. Transport system basic entities

In addition to the identified main entities that affect the functioning of the transport network, there are many others in the real world. Therefore the system as a whole is much more complicated. In addition, the complexity of the system determines many factors that affect the state of the objects of the transport network: weather, the quality of the road surface, the number of road junctions, current traffic, etc.

Based on the analysis of the main entities of a transportation system there were formulated the concepts of the problem domain shown in Table I.

TABLE I. Transport system concepts

Concept	Name	Instance example
Shipping unit, single vehicle	Unit	Dump truck with driver and trailer
Transport instruction	TI	Loading, unloading on a specific order
Cargo loading	Loading	Loading of cargo that the transport unit will transport
Cargo discharge	Unloading	Unloading cargo at the delivery point specified in the transport instruction
Route	Route	The best way to transport cargo from the loading point to the unloading point

The transport network is formalized in a classical way in the form of an undirected connected graph, in which the edges indicate the methods of movement, and the nodes are the start, intermediate and end points of the logistics chain. Each edge of this graph has a weight that indicates the distance a vehicle needs to travel to move from node to node.

In the form of graphs, we will also present many factors that influence the route along with the transport network graph, among them: weather conditions, road surface quality, traffic congestion, the rules for the delivery of goods provided for by logistics companies, etc. All elements of this graph have spatial and temporal binding, i.e. may vary in space and time.

Fig. 2 shows a schematic model for representing graphs in space at a specific time. Traditionally such a graph is developed manually by system input of data describing all the objects using a specifically designed UI or in the automated mode. As a result, the objects of the same type (drivers or transport instructions) are usually grouped together and refer to one layer.

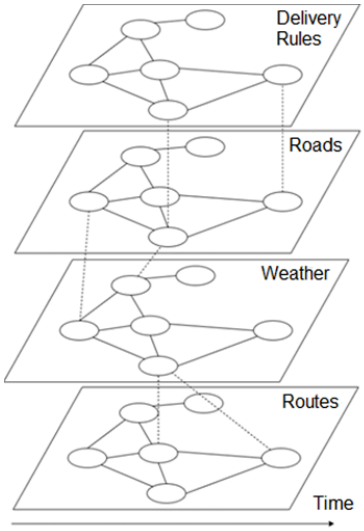


Fig. 2. Transportation network multi-layer description

This approach is sufficient from the perspective of human perception. However, parallel algorithms require different data representation. Considering the necessity to distribute different objects that refer to one situation between the computing facilities it is better to group them according to their negotiation rather than type (see Fig. 3).

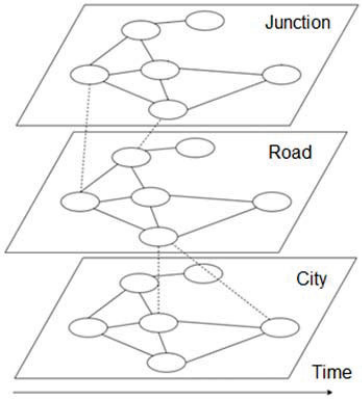


Fig.3. Multi-layer computing oriented modification

At which point distribution of objects between the layers according to the principles of parallel computing is more suitable.

IV. Multi-layer Graph Model

To formalize this solution there was developed a multi-layer graph model. Imagine one data record coming from a data source (Source), as:

$$S = (s_1, s_2, \dots, s_n), \forall\, s \in S: s \in D \cup \emptyset, \quad (1)$$

where s is one value in the data record;

D is the value domain, in the general case:

$$D \subset \mathbb{N} \cup \mathbb{R}. \quad (2)$$

The relationship (binding) between the values of the data record is denoted by the similarity of the adjacency matrix (Binding):

$$\mathcal{B} = \begin{bmatrix} b_{1,1} & \dots & b_{1,n} \\ \dots & & \\ b_{n,1} & \dots & b_{n,n} \end{bmatrix}, \forall\, b \in \mathcal{B}: b \in [0,1] \subset \mathbb{R}, \quad (3)$$

where b is the degree of data connectivity.

If $b_{i,j} = b_{j,i} = 0$, there is completely no connectivity between s_i and s_j, and if $b_{i,j} = b_{j,i} = 1$, s_i and s_j have stable connectivity. Intermediate values $b_{i,j} = b_{j,i} \in (0,1)$ describe, respectively, intermediate connectivity states.

For the adjacency matrix, the following relations should be satisfied:

$$\forall\, j: \sum_{i=1}^{n} b_{i,j} > 0, \quad (4)$$

$$\forall\, i,j: b_{i,j} = b_{j,i},$$

$$\forall\, i: b_{i,i} = 0.$$

We will equate situations in the confidence interval with situations of stable communication for $|1 - b| < \xi$ or lack thereof for $|b - 0| < \xi$, where ξ is a sufficiently small positive number that is acceptable in the current situation.

All incoming records pass through an adaptive filter that maps a lot of source data to a lot of resulting L layers:

$$f: S \to L, \quad (5)$$

$$f^{-1}(l) = \{s \in S \mid f(s) = l\}.$$

On the set of resulting layers, we define an adjacency matrix Γ similar to (1):

$$\Gamma = \begin{bmatrix} \gamma_{1,1} & \dots & \gamma_{1,n} \\ \dots & & \\ \gamma_{n,1} & \dots & \gamma_{n,n} \end{bmatrix}, \forall\, \gamma \in \Gamma: \gamma \in [0,1] \subset \mathbb{R}. \quad (6)$$

We define a mapping function f as follows:

$$f(s) = \operatorname{argmax}_{l \in L} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\gamma_{i,j}(l) - b_{i,j}(s) \right)^2 \right)^{\frac{1}{2}}. \quad (7)$$

Define a filter adaptation method for each display operation:

$$\forall\, i,j : \gamma_{i,j}(s \equiv l) = \begin{cases} \gamma_{i,j}(l), & g_{i,j}(l) = 0; \\ 0, & g_{i,j}(l) + \gamma_{i,j}(l) \le 0; \\ 1, & g_{i,j}(l) + \gamma_{i,j}(l) \ge 1; \\ g_{i,j}(l) + \gamma_{i,j}(l), & otherwise; \end{cases}$$

$$g_{i,j}(s \equiv l) = \begin{cases} 0, & |\gamma_{i,j}(l) - b_{i,j}(l)| < \varepsilon; \\ \dfrac{\gamma_{i,j}(l) - b_{i,j}(l)}{k}, & |\gamma_{i,j}(l) - b_{i,j}(l)| \ge \varepsilon, \end{cases}$$

where ε is the sensitivity to adaptation, a small positive number: $0 < \varepsilon < 1$;

k is the adaptation rate, $k > 0$.

Thus, by adjusting the tolerance variables ξ, ε and varying the filter adaptation rate k, tuning to the current data processing situation is achieved.

Spatial data binding is described by attributes of a special kind, which are introduced into the model by expanding the domain model:

$$\widehat{D} = D \cup M, \tag{8}$$

where M is the set of spatial positions determined by the available sets:

$$M = M_P \cup M_{LS}, \tag{9}$$

where M_P is the set of point objects (Point);

M_{LS} – many line objects (Line String).

The presence of only two sets is due to the specifics of the problem being solved, since these two geometric primitives are sufficient to represent the graph. The line is represented by a broken line, since such a representation is accepted in geoinformatics.

Time indicators are entered into the model using the following display:

$$h: S \to D, \tag{10}$$
$$h^{-1}(d) = \{s \in S \mid h(s) = d\}.$$

The function $h(s)$ itself uses discrete time to determine the required value at t_i time instant:

$$h(s) = \begin{cases} s, \ t_{i+1} \in [t_i, t_i + \Delta t]\,; \\ \emptyset, \ otherwise. \end{cases} \tag{11}$$

The parameter Δt determines the lifetime of the value s. Outside the lifetime, the value of s will be considered undefined. This approach is due to the relatively low cost of irrelevant single attributes from Big Data arrays. Undefined attributes do not participate in the calculations and do not affect the adaptation of the filter.

V. Implementation Results

A. Routing Algorithms

In the work, the method of criteria comparisons is used in two algorithms. The sequential algorithm implements the classical approach to find a path on a graph, while the parallel one uses an approach that makes it possible to increase the efficiency of layer-by-layer task separation, taking into account the capabilities of a computing system for simultaneous parallel calculations.

Using a parallel algorithm on the same input data can lead to different results. The variability that arises during the operation of the parallel algorithm is due to the neglect of the revaluation of the graph weights and normalization at the final stages.

A sequential algorithm consists of the following steps:

Step 1. Successively sort through the weights of each criterion, normalizing them.

Normalization occurs using the reference weights of the graph vertices and expert assessment:

$$F = \sum_{k=0}^{n} \left(\frac{f_k}{f_k^6}\right)^{\alpha_k}, \tag{12}$$

where n is the number of criteria by which optimality is evaluated;

α_k – expert assessment;

f_k is the weight of the vertex of the graph;

f_k^6 is the reference weight.

Step 2. Weigh the graph.

Step 3. Find the optimal path using the Dijkstra algorithm.

The parallel algorithm implements the idea of layer-by-layer separation and consists in performing the following steps:

Step 1. Separate the data according to the criteria.

Step 2. Weigh each graph according to a given criterion (for example, weather, road surface quality).

Step 3. Using a parallel algorithm (the type of algorithm may vary) we find the shortest paths from a given point to a final one using the Dijkstra algorithm.

Step 4. We combine the routes using the reference value for each route according to the specified criterion and select the most suitable one in accordance with the expert assessment.

Step 5. Combine the results of Dijkstra's algorithms, as well as in the sequential algorithm, using the criteria-based comparison formula (2).

B. Data Layers Construction

A greedy algorithm was developed to construct data layers.

Step 1. An initial number of layers is generated for optimization tasks.

Step 2. Each layer attracts instances of involved entities. One entity can enter several layers.

Step 3. Layers that have intersections of objects and subjects upper than the predefined limits are merged.

Step 4. New incoming tasks first try to enter the existing layers. In the case of a low intersection, they generate new layers.

Step 5. Entities with low involvement are reduced to separate layers.

Step 6. Layers with low use are destroyed.

The algorithm requires the proactivity of entities and layers. Therefore, it is recommended to use multi-agent technology for its implementation.

C. Case Study

The data sample for testing the solution covers 21 settlements and 34 roads in the Netherlands (see Fig. 4).

To determine the performance of the algorithms, we will conduct performance tests. In this test configuration, 3 algorithms with 30 iterations each were launched:

1) The ApacheSpark algorithm is a variation of the parallel algorithm that uses MapReduce provided by the Apache Spark library.

2) The JavaStreams algorithm is a variation of the parallel algorithm that uses MapReduce, provided by the Stream API and supports parallel threads to use several processor cores.

3) The SimpleIteration algorithm is a sequential algorithm that combines the weight of the criteria into one.

Fig. 5 shows the routes constructed by each of the presented algorithms for organizing transportation from Hoek van Holland to Groningen.

Performance test results are shown in Table II.

The resulting execution time of the iterations of the algorithms is shown in Fig. 6.

Fig. 4. The Netherlands road net weighted graph model

Fig. 5. Possible routes

TABLE II. PERFORMANCE ANALYSIS RESULTS

Benchmark	Mode	Count	Score	Error	Units
ApacheSpark	Avgt	30	39,266	8,600	ms/op
JavaStreams	Avgt	30	0,318	0,038	ms/op
SimpleIteration	Avgt	30	0,514	0,041	ms/op

The minimum, maximum and average time for calculating the optimal path by various algorithms is shown in Fig. 7. The dispersion and standard deviation are shown in Fig. 8.

Based on the test results for the average run time, we can conclude that for a parallel (or distributed) JavaStreams algorithm, which implements the criterion separation of transport system parameters, 38% less time is required compared to a sequential algorithm.

However, the implementation method parallel processing greatly affects the final result.

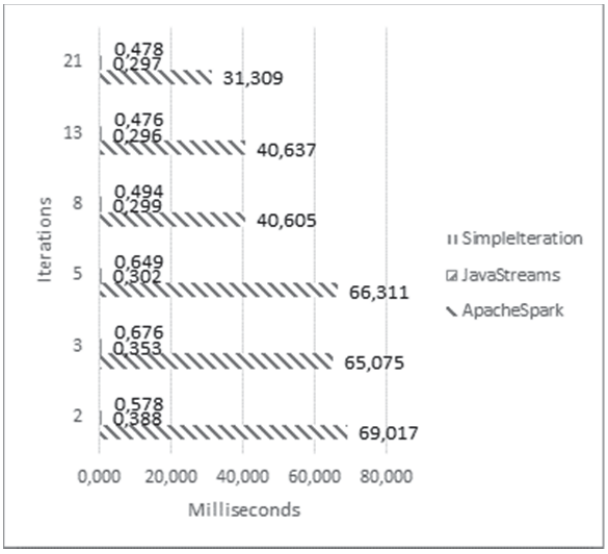


Fig. 6. Algorithm performance time depending on iterations number

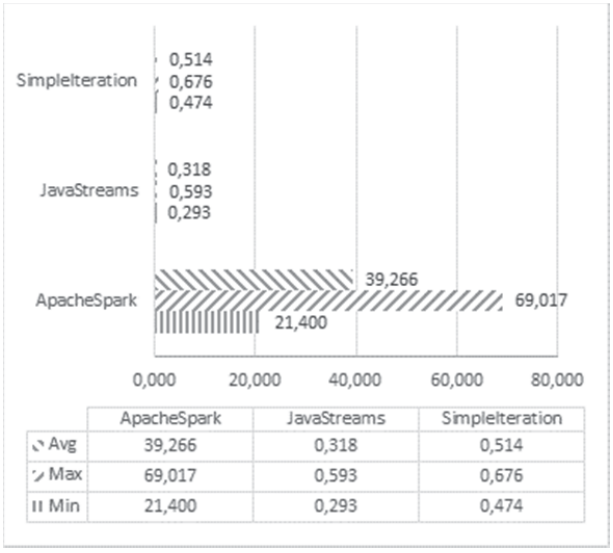
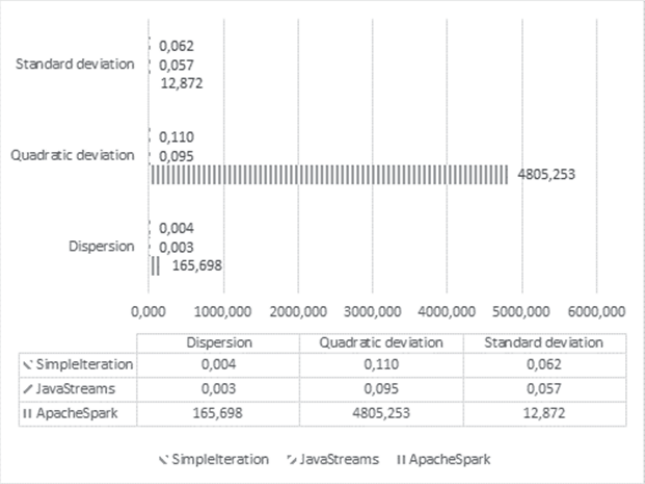


Fig. 7. Route optimization time



	Dispersion	Quadratic deviation	Standard deviation
SimpleIteration	0,004	0,110	0,062
JavaStreams	0,003	0,095	0,057
ApacheSpark	165,698	4805,253	12,872

Fig. 8. Deviations

In this case, the implementation based on the Stream API was successful comparing to Apache Spark.

VI. Conclusion

Transportation logistics is a network of cooperating and evolving subjects and objects. Both entities emit data flows that describe the events that happen in the system. Modern software solutions are capable of processing this data in real-time and utilize it for decision-making support. Experimental results prove the necessity to introduce original data structures for parallel processing of this data. The provided approach allows improving performance in calculating optimal routes for carrier companies by dividing the criteria for the functioning of the transport system into multiple graph layers.

The approach is implemented using Apache Spark and Stream API. A study conducted on real data on 21 settlements in the Netherlands showed an advantage in the execution time of the parallel computing algorithm over the usual sequential search of 38% that considers the features of parallel computing. Despite the difference between the high execution time in transportation logistics and comparatively low time required for decision-making, optimization of scheduling allows increasing the number of options being considered by a dispatcher and sufficiently reduce costs. In addition to this, the scheduling of transportation resources is often time and labor-consuming. Therefore, the application of the proposed approach is reasonable in various problem domains.

Possible limitations are concerned with the automatic generation of layers that do not consider the problem domain specifics. Due to this fact, both approaches of human and automated layers construction should be combined in practice. To provide such an opportunity each generated layer should be supplemented with a description that explains the proposed solution to a human decision-maker with formal reasoning.

Further improvement of the approach is seen in improving the system of classification and clustering of layers, for example, on the basis of a solution previously tested by the authors in [23]: by correctly assessing and identifying criteria for the functioning of the transport system into separate layers, it is possible to increase the speed of parallel computing.

References

[1] O. K. Golovnin, “Data-driven profiling of traffic flow with varying road conditions”, *CEUR Workshop Proceedings*, vol. 2416, May 2019, pp. 149-157.

[2] B. Baesens, “Analytics in a Big Data world: The essential guide to data science and its applications”. Wiley, 2014, 232 p.

[3] G. Clarke and J. Wright, “Scheduling of vehicles from a central depot to a number of delivery points”, *Operations Research*, vol. 12 (4), Aug. 1964, pp. 568-581.

[4] O. Surnin, P. Sitnikov, A. Suprun, A. Ivaschenko, A. Stolbova, and O. Golovnin, “Urban Public Transport Digital Planning based on an Intelligent Transportation System”, in *Proc. of the FRUCT’25*, Nov. 2019, pp. 292-298.

[5] M. Batty, “Big data, smart cities and city planning”, *Dialogues in Human Geography*, vol. 3(3), Nov. 2013, pp. 274-279.

[6] A. Ivaschenko, M. Frolova “Intelligent intermediaries for multiple logistics parties”, in *Next Generation Logistics: Technologies and Applications*. SPH – Scientific Publishing Hub, 2017, pp. 21-42

[7] A. Ivaschenko, “Multi-agent solution for business processes management of 5PL transportation provider”, *Lecture Notes in Business Information Processing*, Vol. 170, 2014, pp. 110-120

[8] A. B. Morton and I. M. Y. Mareels, “An efficient brute-force solution to the network reconfiguration problem”, *IEEE Trans. on Power Delivery*, vol. 15(3), July 2000, pp. 996-1000.

[9] I. A. Kanj, L. Nakhleh, C. Than, and G. Xia, “Seeing the trees and their branches in the network is hard”, *Theoretical Computer Science*, vol. 401(1-3), 2008, pp. 153-164.

[10] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, “A cost-effective recommender system for taxi drivers”, in *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Aug. 2014, pp. 45-54.

[11] D. Taillard, “Parallel iterative search methods for vehicle routing problems”, *Networks*, vol. 23, Dec. 1993, pp. 661-673.

[12] L. Zhu, F.R. Yu, Y. Wang, B. Ning, and T. Tang, “Big Data analytics in intelligent transportation systems: a survey”, IEEE Trans. on Intelligent Transportation Systems, vol. 20(1), April 2018, pp. 383-398.

[13] T.-H. Kim, S.-J. Kim, H. Ok, “A study on the cargo vehicle traffic patterns analysis using Big Data”, in *ACM Int. Conf. Proc. Series*, Oct. 2017, pp. 55-59.

[14] M. A. Javed, S. Zeadally, E. B. Hamida, “Data analytics for cooperative intelligent transport systems”, *Vehicular communications*, vol. 15, Jan. 2019, pp. 63-72.

[15] K.-H. Lee, Y.-J. Lee, H.Choi, Y.D. Chung, and B. Moon, “Parallel data processing with MapReduce: a survey”, *SIGMOD Record*, vol. 40(4), Jan. 2011, pp. 11-20.

[16] C. T. Chu, S. K. Kim, Y. A. Lin, Y. Yu, G. Bradski, K. Olukotun, and A. Y. Ng, “Map-reduce for machine learning on multicore”, *Advances in neural information processing systems*, 2007, pp. 281-288.

[17] D. Zhang, Y. Shou, and J. Xu, “A mapreduce-based approach for shortest path problem in road networks”, *J. Ambient. Intell. Human Comput*., 2018. pp. 1-9.

[18] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters”, *Communications of the ACM*, vol. 51(1), Jan. 2008, pp. 107-113.

[19] B. Keswani, P. Vijay, and P. Keswani, “Enhanced approach to attain competent Big Data pre-processing”, *Int. J. of Adv. Studies of Scientific Research*, vol. 4, May 2019, pp. 1-4.

[20] M. A. Bhuiyan and M. Al Hasan, “An iterative MapReduce based frequent subgraph mining algorithm”, *IEEE Trans. on Knowledge and Data Engineering*, vol. 77, Aug. 2014, pp. 608-620.

[21] J. Yu, A. Tahir, and M. Sarwat, “GeoSparkViz in action: a data system with built-in support for geospatial visualization”, in *Int. Conf. on Data Engineering*, June 2019, pp. 1992-1995.

[22] M. L. Fisher and R. Jaikumar, “A generalized assignment heuristic for vehicle routing”, *Networks*, vol. 11, 1981, pp. 109-124.

[23] A. Ivaschenko, A. Stolbova, and O. Golovnin, “Spatial clustering based on analysis of Big Data in digital marketing”, *Communications in Computer and Information Science*, vol. 1093, Oct. 2019, pp. 335-347.