# The method of calculating the assembly and delivery plan for groupage cargoes in the special VRPTW problem of intra-city food delivery 

Lada Alexander ${ }^{1,2}$, Skobelev Petr ${ }^{1}$<br>${ }^{1}$ Samara State Technical University, Russia<br>\{ladalexer, petr.skobelev\}@gmail.com<br>${ }^{2}$ SEC Smart Transport Systems, Samara, Russia

lada@kg.ru


#### Abstract

A method of calculating the delivery plan with regards to coordination with a storehouse picker's schedule in the consideration of intra-city food delivery has been analyzed and is being proposed. In order to assemble a client's order, storehouse pickers are employed, who, upon notification, collect the necessary items for packaging and delivery. The picker's assembly schedule should be consistent with couriers' intended schedule of travel around the city; delivering packages to clients, according to the client's preferred time windows. It is necessary to calculate a coordinated and conditionally optimal plan for storehouse pickers on the one hand, and for courier drivers on the other, taking into account their mutual interrelationship. At the same time, it is required to be able to pro-vide a desired time window for delivery of the order to the client. These calculations result in minimizing the courier's route, taking into account the forecast of traffic congestion at certain times of the day, adaptively redistributing the assembly of packages (order filling) to other stores, if all pickers in the current location are overloaded.


Keywords VRP, adaptive resource scheduling, transport logistics, LTL, groupage cargoes, assembly scheduling

## 1 Introduction

The problem of transportation optimization (Vehicle Routing Problem, VRP), was first described in [1], is one of the most significant tasks of the modern optimization theory. Classification of optimization problems of transport logistics is given in [2, 3]. A review and classification of VRP problems with the proposed solution methods is also given in [4].

Transportation process in modern transport logistics is divided into two types: FTL and LTL. FTL (Full Truck Load) transportation is characterized by the fact that the carrier has large contracts with customers for reserving a truck as a whole. This means that all the orders the truck performs consistently, one truck can simultaneously carry only one order: it loaded the first order, drove through the points of its unloading and unload all, went to the second order, performed it in the same way, went to the third order and etc. It is possible to substantially simplify the task of optimizing such transportations, not taking into account the possibility of partial loading while constructing the route. The description of models of FTL transportation is given in [5]. The solution of the FTL problem was described by the authors in [6-8]. LTL (Less than Truck Load) transportations in opposite allow parallel execution of orders, i.e. one truck can simultaneously execute several orders: it loaded the first order, drove through the points of its loading and on the way loaded the second order, un-loaded the first and part of the second, went to the third order, etc. The task of optimizing LTL transportations is much more complicated than FTL, in such tasks in addition it is necessary to take into account the time windows for the arrival of the truck for loading and unloading, which allows assigning these tasks to the class of VRPTW (Vehicle Routing Problem with Time Windows). Exact solutions to this problem, with the exception of checks, on large volumes of orders and resources are not currently known, but there are imitation methods that give an inaccurate solution, but of acceptable quality. As a rule, these are hybrids and varieties of branch and bound methods, as, for example, in [9], or various other heuristic methods (genetic algorithms, neural networks, etc.). The authors of this study have experience in solving similar problems [10, 11], as well as their implementation in applied multi-agent systems described in [12, 13].

In this paper, the special VRPTW task of calculating a plan for a courier delivery service for food products is considered, which, in addition, it is necessary to calculate a plan for assembling food into packages according to clients orders by the employees (pickers) of grocery stores. It is also required adaptability to reschedule the delivery in case of actual deviations from the plan during assembly caused by objective reality. With this setting, the original VRPTW task is further complicated by the need for constant "looking back" at the order-picking plan, which in turn requires a reconfiguration and reconciliation of the delivery plan with the assembly plan. In other words it is necessary to schedule and reschedule the courier route in real time, select the most convenient loading locations (order picking shops), and unload orders to customers along the way. It is also necessary to schedule the work of the order pickers, by creating an assembly plan for each order, adjusting it to constantly incoming orders and sending the couriers to them
in time. Finally, if the courier is stuck in a traffic jam - it can be considered as a mobile warehouse and connecting other couriers, for example, on motorcycles or even bicycles, in order to intercept a late order and deliver the goods on time to the consumer.

In this paper, we consider the statement of the problem and the approach to its solution based on mobile resource management. It should be understood that the real business problem is formulated much wider than mathematical problem, which was described above, because it includes many additional criteria that need to be taken into account in the business process. One of the key requirements for the development is the need for the system to work in real time. In addition, it is required to take into account the workload of the order pickers at each moment of time and the fore-cast of the road situation (traffic jams) at the time of the courier's departure to the client.

In the first section, we will consider the formulation of the problem, in the second we will propose an approach to the solution, and in conclusion - we will consider the results of the first steps of implementation and the prospects for the development and application of the system.

## 2 The assembly problem

In the logistical system of intra-city delivery of food products under the LTL scheme, it is necessary to arrange the reception of orders, their processing, the assembling of goods into packages and delivering it by couriers. To do this, we have to solve two different tasks create an assembly and delivery plan.

### 2.1 The assembly plan problem definition

Suppose we have a number of stores, each based in a particular location. In each store we have a number of assembly workers (pickers). We have a flow of orders from clients. Each order is characterized by the location of delivery and the requested time window and by the number of goods that should be delivered. For each incoming order, it is necessary to determine the place of its assembly in one of the stores taking into account the current and projected overload of pickers. We consider that any good is available in any store with infinite quantity. The goal is to schedule the order assembly in the closest delivery location store, taking into account the real time speed of assembly and change the store to the next closest one if according to its schedule in the store it is impossible to deliver it in the requested time window. The task is solving by calculating the planned start and finish times of the assembly according to the following method.

### 2.2 The assembly plan construction method

It is assumed that the actual start and end time of the order comes to the scheduling system from the external information system in real time directly (e.g. from the mobile application for pickers). In addition to the start and finish times, the system calculates the planned start and finish times for all orders based on the assembly speed of the item in the store (depends on the number and productivity of the pickers). It is assumed that each time getting information about a new fact of assembling the goods in a particular store, we may recalculate the current speed of its assembly and recalculate the planned times of the beginning and ending of the assemblies that have not yet been collected.

It is assumed that the assembly times of all orders in the store are calculated in parallel queues, the order in which is determined according to the time of each order arrival in the system and is corrected by the actual times of the beginning and end of the assembly. The number of these queues is equal to the number of pickers in the store. Within each queue, orders are executed sequentially.

Let's look at the example, as it is proposed to do. Suppose we have a store with 2 pickers and a table of equivalent orders. At the initial moment of time, until there is no actual time we believe that the assembly time of each order by one picker is 5 minutes (Table 1).

Table 1 Chronological order processing table

| Order num- <br> ber | Order <br> time | Scheduled time to <br> start the assembly | Scheduled time to Actual time the <br> finish the assembly assembly started <br> Assembly fin- <br> ished |
| :--- | :--- | :--- | :--- |
| O1 | $13: 00$ | $13: 00$ | $13: 05$ |
| O2 | $13: 00$ | $13: 00$ | $13: 05$ |
| O3 | $13: 00$ | $13: 05$ | $13: 10$ |
| O4 | $13: 15$ | $13: 15$ | $13: 20$ |

Orders O 1 and O 2 are collected in parallel, since there are two pickers in the store that have the same estimation on the time of the beginning and end of the assembly.

Order O3 should wait until either the first or the second picker is released and therefore according to the plan will be collected only after the first two.

Order O4 was received at 13:15, by this time all the pickers have already been released and only one of them will execute the order, because it is believed that two pickers at the same time do not collect the order.

Next, consider the example of Table 2, how the scheduled start/end times of the assembly will be corrected when the actual data arrives.

Table 2 Chronological table for processing orders after the actual start of order assembly O1

| Order num- <br> ber | Order <br> time | Scheduled time to <br> start the assembly | Scheduled time to <br> finish the assem- <br> bly | Actual time the <br> assembly startedActual time the <br> assembly fin- <br> ished |
| :--- | :--- | :--- | :--- | :--- |
| O1 | $13: 00$ | $\mathbf{1 3 : 1 0}$ | $\mathbf{1 3 : 1 5}$ | $13: 10$ |
| O2 | $13: 00$ | $\mathbf{1 3 : 1 0}$ | $\mathbf{1 3 : 1 5}$ |  |
| O3 | $13: 00$ | $\mathbf{1 3 : 1 5}$ | $\mathbf{1 3 : 2 0}$ |  |
| O4 | $13: 15$ | $13: 15$ | $13: 20$ |  |

At the time 13:10 the fact of the start of assembly was received from O1, which started 10 minutes later than planned and, as a result, will now be collected at 13:15, this led to changes in planned start and end times for the remaining orders.

Order O2 according to the plan will be collected only at $13: 10$, because it's already $13: 10$, but the fact of the beginning of the assembly is not yet received.

Order O3 is shifted 5 minutes to the right.
According to the plan of order O4, no changes will occur, because it has some extra time in reserve.

New factual data was received (Table 3).

Table 3 Chronological table for processing orders after the actual start of order assembly O1

| Order num- <br> ber | Order <br> time | Scheduled time to <br> start the assembly | Scheduled time to <br> finish the assem- <br> bly | Actual time the <br> assembly started | Actual time the <br> assembly fin- <br> ished |
| :--- | :--- | :--- | :--- | :--- | :--- |
| O1 | $13: 00$ | $13: 10$ | $\mathbf{1 3 : 2 0}$ | $13: 10$ | $13: 20$ |
| O2 | $13: 00$ | $\mathbf{1 3 : 2 0}$ | $\mathbf{1 3 : 3 0}$ |  |  |
| O3 | $13: 00$ | $\mathbf{1 3 : 2 0}$ | $\mathbf{1 3 : 3 0}$ |  |  |
| O4 | $13: 15$ | $13: 30$ | $13: 40$ |  |  |

At the time 13:20 the fact of the completion of the assembly was received from O1, which was completed 5 minutes later than planned, and, therefore, the picker's assembling time was changed (instead of the initial 5 minutes now 10 minutes), which led to changes in planned starting points And finish the assembly of other orders.

Order O2 according to the plan will be collected only at 13:20, because it's already 13:20, and the fact of the beginning of the assembly is not yet received, and taking into account the new speed, the assembly will take 10 minutes, so it will end at $1: 30 \mathrm{pm}$.

Order O3 is collected in parallel with the order O2.
Order O4 on the plan will move and will begin to be collected immediately after the order of O 2 or O 3 .

New factual data was received (Table 4).

Table 4 Chronological table for processing orders after the actual start of order assembly O2

| Order num- <br> ber | Order <br> time | Scheduled time to <br> start the assembly | Scheduled time to <br> finish the assem- <br> bly | Actual time the <br> assembly started | Actual time the <br> assembly fin- <br> ished |
| :--- | :--- | :--- | :--- | :--- | :--- |
| O1 | $13: 00$ | $13: 10$ | $13: 20$ | $13: 10$ | $13: 20$ |
| O2 | $13: 00$ | $\mathbf{1 3 : 2 5}$ | $\mathbf{1 3 : 3 5}$ | $13: 25$ |  |
| O3 | $13: 00$ | $\mathbf{1 3 : 2 5}$ | $\mathbf{1 3 : 3 5}$ |  |  |
| O4 | $13: 15$ | $\mathbf{1 3 : 3 5}$ | $\mathbf{1 3 : 4 5}$ |  |  |

At the time 13:25 came the fact of the beginning of the assembly of the order O3 (for convenience, orders with facts immediately go up the queue), which began 5 minutes later than planned, which led to changes in the planned start and end times for the remaining orders

New factual data was received (Table 5).

Table 5 Chronological table for processing orders after the actual completion of order assembly O2

| Order num- <br> ber | Order <br> time | Scheduled time to <br> start the assembly | Scheduled time to <br> finish the assem- <br> bly | Actual time the <br> assembly startedActual time the <br> assembly fin- <br> ished |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| O1 | $13: 00$ | $13: 10$ | $13: 20$ | $13: 10$ | $13: 20$ |
| O2 | $13: 00$ | $13: 25$ | $\mathbf{1 3 : 3 0}$ | $13: 25$ | $13: 30$ |
| O3 | $13: 00$ | $\mathbf{1 3 : 3 0}$ | $\mathbf{1 3 : 3 8}$ | $13: 30$ |  |
| O4 | $13: 15$ | $\mathbf{1 3 : 3 0}$ | $\mathbf{1 3 : 3 8}$ |  |  |

At the time 13:30 the fact of the beginning of the assembly of the O 2 order and the completion of the assembly about the ordering of O3 came, now the order assembly rate is estimated as the average value between the difference for the two facts of the beginning and the end already received and will be ((13:20$13: 10)+(13: 30-13: 25)) / 2=7,5$. The calculations are then carried out in a similar way.

### 2.3 Choosing the best store to assemble the order

After calculating the planned finish time for the order assembly for which the store is currently determined, it is checked for each store whether the courier will reach the client theoretically, from that time to the end of the unloading window, taking into account the time of arrival from the current store to the client. If it does, the result for this store is fixed as possible and saved. From all possible options of stores, we should choose the one that requires the minimum trip time from it to the client (that is, the minimum of the driver's work in the form of mileage), provided that the planned assembly time in this store does not go beyond the loading window, otherwise, we consider a more remote store, etc., till the end of the list of shops. We can do the same thing after we made the choice according to the real time if the picker did not finish the assembly of the order before the deadline (skipped the deadline for the assembly), we can try to plan the assembly of this order in another store.

## 3 The delivery problem

After the selection of the store is completed, we should create delivery schedule for each courier taking into account the number of available couriers, orders delivery time window, the trip time along the route line in the city with traffic jams forecasting. We also should adaptively change the plan according to real time events (delays during orders delivery process).

### 3.1 The delivery plan problem definition

Suppose we have a number of couriers each of them is characterized by maximum carried volume and weight. The location for each courier can be evaluated at any moment of time. We have a set of LTL-orders each of them is characterized by the package volume and weight and has only one location of loading (the store location) and one location of unloading (the client location). All orders must be delivered no later than the right window time. Delays are not considered acceptable. We also consider the case of finishing the delivery work exactly at the close of the window as acceptable (ie, the start and end boundaries of the time windows are included). The trip time based on distances between all locations with the traffic jams forecasting can be evaluated at any moment of time. There are external data providers for traffic forecast congestions. The most popular providers in Russia are Yandex [14] and Google [15]. The scheduling system requests the time for the route from point A to point B from the "jam" service for the time of the planned delivery, having received the result, uses it in future planning as the time of the execution of the route in the process of fulfilling the order. The goal is to allocate orders for available couriers, bringing the total execution time to a minimum, but maximizing the number of delivered orders and possibly reducing the total trip distance.

### 3.2 The delivery plan construction method

For construction of the plan, we propose the following method. All orders are sorted by the loading time (estimated time when the assembly ends). At the initial time moment, we suppose that the considered courier is allocated in the first order loading point (considered shop). Let the courier in this point appears in the start time $\mathrm{T}_{0}$, in advance of the earliest loading time of the first order. With respect to this starting point and the initial time $\mathrm{T}_{0}$ the table of the possible assignments is calculated. In the table, each order potential execution time is calculated according to the formula $\mathrm{OET}_{\mathrm{i}}$ (order execution time) $=\mathrm{TLi}_{\mathrm{i}}($ trip to loading $)+\mathrm{WLi}^{\text {( work }}$ to load) $+\mathrm{TU} \mathrm{U}_{\mathrm{i}}$ (trip to unloading) $+\mathrm{WU}_{\mathrm{i}}$ (work to unload). If OET is in order unloading time window, the order is considered as possible for delivery.

In the generated table we select order with the shortest OET. After that we change the location of the considered courier, which now corresponds to the chosen place of loading order, and calculate the time of its release as $\mathrm{T}_{1}=\mathrm{TLi}_{\mathrm{i}}+\mathrm{WLi}$.

After the first assignment, we generate the table again, but now we exclude some orders from that if they match one of the following:

1. Summarized volume and weight for previously assigned orders with the considered one excel the courier maximum carried volume and weight.
2. There is no time for courier to move from current location to the considered loading location because it mismatch the considered order loading time window.
3. With regard to the considered order pick up, the courier has no time to unload previously assigned orders.
For the remaining after filtering orders, we again calculate their execution times also considering waiting times before opening loading windows with new location of the courier. We choose the shortest execution time again. Then the algorithm is repeated until we have no possible orders to allocate because of above described restrictions. Then the next considered courier is placed in the first order loading point, and repeats the same operation for the remaining orders and so on until there is no order left for allocation.

### 3.3 Changing the plan according to real time events

There are two types of possible problems during the order delivery by courier: delay and unavailability (crash, accident, etc.). In case of having a delay problem, which may occur when the courier rides with orders on the route for example, gets into a traffic jam or there is a problem with the car. Delay is also may occur when the courier arrives at the place and begin unloading the order, but in the process arose some difficulties (closed territory, need a pass, not working elevator, need to climb the stairs, etc.). In this case, all already carried orders by the courier will be potentially delivered with delay but the next orders that should be loaded by this courier in the future will be rescheduled to other couriers according to the same method.

Unavailability is a more major problem, which means that the courier can no longer proceed the delivery (because of crash, accident, etc.). In this case, the courier requests for help by creating an additional loading point in the place where he stopped moving and from where he cannot continue to carry out his orders. As a result, a new loading point with orders of this courier is created. These orders will be rescheduled to other couriers according to the same method.

## 4 The experimental results

We performed several experiments to test the described method by using the real data from our client company [16]. The goal of the experiments was to compare the proposed construction delivery plan method and the brute force method It was possible to obtain the brute force results only for the first four experiments at a suitable time. The initial data example for the experiments represents in following tables:

Table 5 Chronological table of orders with details in unified units

|  | LWs |  |  | LWf | LT | ULWs |  |  | ULWf | ULT | CL |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| $\mathrm{O}_{1}$ | 1 | 2 | 1 | 6 | 9 | 1 | 5 |  |  |  |  |
| $\mathrm{O}_{2}$ | 2 | 3 | 0 | 6 | 9 | 0 | 4 |  |  |  |  |
| $\mathrm{O}_{3}$ | 3 | 5 | 1 | 12 | 14 | 1 | 5 |  |  |  |  |
| $\mathrm{O}_{4}$ | 3 | 5 | 1 | 12 | 14 | 1 | 4 |  |  |  |  |
| $\mathrm{O}_{5}$ | 4 | 6 | 0 | 17 | 19 | 0 | 4 |  |  |  |  |

LWs - start loading window,
LWf - end loading window,
LT -loading working time,
ULWs - start unloading window,
ULWf - end unloading window,
ULT - unloading working time,
CL - cargo load capacity

Table 6 The matrix of distances between orders loading and unloading points in unified units

| $\mathrm{LO}_{1} \mathrm{LO}_{2} \mathrm{LO}_{3} \mathrm{LO}_{4} \mathrm{LO}_{5}$ |  |  |  |  | $\mathrm{LO}_{1} \mathrm{LO}_{2} \mathrm{LO}_{3} \mathrm{LO}_{4} \mathrm{LO}_{5}$ |  |  |  |  | $\mathrm{ULO}_{1} \mathrm{ULO}_{2} \mathrm{ULO}_{3} \mathrm{ULO}_{4} \mathrm{ULO}_{5}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathrm{LO}_{1}$ | 0,2 | 0,3 | 0,4 | 0,5 | ULO 3 | 2,5 | 3 | 2,5 | 2,6 | $\mathrm{ULO}_{1}$ | 0,3 | 0,4 | 0,5 | 0,4 |
| $\mathrm{LO}_{2} 0,2$ |  | 0,6 | 0,5 | 0,5 | ULO25 | 4 | 6 | 4 | 4 | $\mathrm{ULO}_{2} 0,3$ |  | 0,2 | ,3 | 0,4 |
| $\mathrm{LO}_{3} 0,3$ | 0,6 |  | 0,4 | 0,3 | $\mathrm{ULO}_{3} 4$ | 5 | 6 | 4 | 3 | $\mathrm{ULO}_{3} 0,4$ | 0,2 |  | 0,1 | 0,5 |
| $\mathrm{LO}_{4} 0,4$ | 0,5 | 0,4 |  | 0,1 | ULO46 | 7 | 5 | 6 | 4 | ULO 40,5 | 0,3 | 0,1 |  | 0,6 |
| $\mathrm{LO}_{5} 0,5$ | 0,5 | 0,3 | 0,1 |  | ULO57 | 5 | 6 | 4 | 3 | ULO 50,4 | 0,4 | 0,5 | 0,6 |  |

$\mathrm{LO}_{\mathrm{i}}-$ order loading point, $\mathrm{ULO}_{\mathrm{i}}-$ order unloading point.

Table 7 The matrix of distances between couriers initial points and orders loading points in unified units

|  | $\mathrm{LO}_{1}$ | $\mathrm{LO}_{2}$ | $\mathrm{LO}_{3}$ | $\mathrm{LO}_{4}$ | $\mathrm{LO}_{5}$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| $\mathrm{C}_{1}$ | 0,3 | 0,4 | 0,5 | 0,4 | 0,6 |
| $\mathrm{C}_{2}$ | 0,5 | 0,2 | 0,3 | 0,4 | 0,7 |


| $\mathrm{C}_{2}$ | 0,5 | 0,2 | 0,3 | 0,4 | 0,7 |
| :--- | :--- | :--- | :--- | :--- | :--- |

$\mathrm{C}_{\mathrm{i}}$ - the starting point of courier with a load capacity of 25 units (all the orders can be potentially delivered by one courier because the sum of all $\mathrm{CL}=5+4+5+4+4<25$ )

Table 8 The experimental results table

| Number of orders | Number of couriers | Proposed method result in unified units | Proposed method calculation time in seconds | Brute force method result in unified units | Brute force method calculation time in seconds |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 5 | 2 | 17 | 0,001 | 17 | 10 |
| 10 | 2 | 25 | 0,001 | 24 | 1983 |
| 15 | 3 | 30 | 0,001 | 28 | 19865 |
| 20 | 4 | 50 | 0,001 | 49 | 167541 |
| 25 | 5 | 70 | 0,001 |  |  |
| 30 | 5 | 85 | 0,001 |  |  |
| 35 | 6 | 95 | 0,001 |  |  |
| 40 | 7 | 111 | 0,001 |  |  |
| 45 | 8 | 122 | 0,001 |  |  |
| 50 | 9 | 131 | 0,003 |  |  |
| 55 | 9 | 139 | 0,001 |  |  |
| 60 | 10 | 153 | 0,001 |  |  |
| 65 | 10 | 165 | 0,003 |  |  |
| 70 | 12 | 195 | 0,001 |  |  |
| 75 | 15 | 205 | 0,011 |  |  |
| 80 | 16 | 223 | 0,014 |  |  |
| 85 | 17 | 238 | 0,01 |  |  |
| 90 | 17 | 243 | 0,01 |  |  |
| 95 | 20 | 260 | 0,08 |  |  |
| 100 | 20 | 284 | 0,08 |  |  |

## 5 Conclusion

The method to the solution of the assembling and delivering problem for groupage cargoes in the special VRPTW problem of intra-city food delivery is developed and described in detail. Based on the developed method the intellectual dispatching management software was created, which is used in the daily work of the food delivery service of Instamart company [16] (the authors received a letter of gratitude). This allowed to reduce the average assembly time of the order by $15 \%$ mainly because of taking into account the actual assembly speed of each store picker and adaptation to the current situation by redirection the order assembly to another store. From the delivery part this allowed to reduce the average number of delays by $22 \%$ because of quick delivery route evaluation for couriers considering traffic forecast situation and respond to emergency situations during delivery. When one courier has a problem, it is possible to solve it with help of the others by adaptive change of their current delivery plan. The proposed approach complements a collection of methods and tools developed in [17, 18]

Acknowledgments This work was supported by Russian Foundation for Basic Research, project number 16-01-00759

## References

[1] Dantzig G.B., Ramser J.H. The Truck Dispatching Problem // Management Science. 1959. Vol. 6, No. 1. P. 80-91.
2] Michael Browne; Alan McKinnon; Sharon Cullinane. Green Logistics by Anthony Whiteing Published by Kogan Page, 2010
[3] Bertazzi, L., Savelsbergh, M., \& Speranza, M. G. (2008). Inventory Routing. In Bruce Golden, S. Raghavan, \& E. Wasil (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges (Vol. 43, pp. 49-72). New York: Springer Science+Business Media. doi:10.1007/978-0-387-77778-8
[4] The vehicle routing problem: state of the art classification and review De Jaegere N, Defraeye M, Van Nieu-wenhuyse, KU Leuven. [Internet] // Available from: https://lirias.kuleuven.be/bitstream/123456789/457452/1/KBI 1415.pdf [Cited 2015 Jan 14].
[5] Oleg Granichin, Petr Skobelev, Alexander Lada, Igor Mayorov, Alexander Tsarev. Cargo transportation models analysis using multi-agent adaptive real-time truck scheduling system. // Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART'2013), February 15-18, 2013, Barcelona, Spain. - SciTePress, Portugal, 2013, Vol. 2. - pp. 244-249.
[6] Petr Skobelev, Alexander Lada, Igor Mayorov. Finding an initial plan of transport resources FTL allocation in a special VRP problem using linear programming methods // Proceedings of the 19th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2015), Orlando, Florida, USA, July 12-15, 2015, vol.1. - International Institute of Informatics and Systemics, - P. 16-21. ISBN 978-1-941763-24-7.
[7] P.O. Skobelev, A.N. Lada. A Solution to the Subtask of Initial Distribution of Transport Resources in a Special Optimization FTL Transportation Problem in Real-time Using the Hungarian Algorithm // Indian Journal of Science and Technology. - Vol 9(12), March 2016. -pp.1-8. DOI: 10.17485/ijst/2016/v9i12/89520
[8] A.N. Lada, V.V. Sazonov, P.O. Skobelev. Method for Transportation Cost Calculation on the Basis of Full Cycle (Round Trip) // Indian Journal of Science and Technology. - Vol 9(20), May 2016. - pp. 1-8. DOI: 10.17485/ijst/2016/v9i20/94478
[9] Andreas Bortfeldt. A Hybrid Algorithm for the Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints // Andreas Bortfeldt - Diskussionsbeiträge der Fakultät für Wirtschaftswis-senschaft der FernUniversität in Hagen, 2010-15 C.
[10] V.V. Sazonov, P.O. Skobelev, A.N. Lada, I.V. Mayorov. Application of multi-agent technologies in the transport task with time windows and several loading points // Management of large systems (electronic journal). Issue 64. Moscow: IPP RAS, 2016. p. 65-80
[11] V.V. Sazonov, P.O. Skobelev, A.N. Lada, I.V. Mayorov. Multi-agent approach to the solution of the transport problem with several loading points and time windows // Proceedings of the XVIII International Conference "Problems of Control and Modeling in Complex Systems", Samara, September 20-25, 2016 - Samara 2016. - p. 171-179.
[12] P.O. Skobelev, A.N. Lada, A.V. Tsarev. Development of intelligent management system for groupage cargoes transportation in real time // Bulletin of the Samara State Technical University, series "Technical Sciences". 2013. № 3(39).- p. 65-74.
[13] P.O. Skobelev, A.N. Lada. SMART LOGISTICS: multi-agent system for groupage cargoes managing in the intraregional deliveries // Proceedings of the XVI International Conference "Problems of Control and Modeling in Complex Systems", Samara, June 30-July 03, 2014. Samara: SSC RAS, 2014. - p. 253-261. - ISBN 978-5-93424-703-5
[14] Traffic jams - Yandex Technologies [Internet] / Available from: https://tech.yandex.ru/maps/doc/jsapi/1.x/mod/concepts/traffic-docpage/ [12 April 2017]
[15] Maps API: creation of applications and analytics [Internet] / Available from: https://enterprise.google.ru/intl/ru/maps/products/mapsapi.html [12 April 2017]
[16] Home delivery products in the online store Instamart. Buy products with free delivery in Moscow. [Internet] / Available from: https://instamart.ru/ [12 April 2017]
[17] Vittich V.A., Skobelev P.O. Multi-agent interaction models for building needs networking and opportunities in open systems; Moscow: Automation and Remote Control; 2003. p. 177185
[18] G.Rzevski, P.Skobelev. Managing Complexity - WIT Press, UK, 193 pp. -http://www.amazon.com/Managing-Complexity-GRzevski/dp/1845649362/ref=sr 1 3?s=books\&ie=UTF8\&qid=1396421715\&sr=13\&keywords=managing+complexity

Proceedings of the 7th Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing (SOHOMA17), October 19-20, 2017, Nantes, France. - In: Borangiu T., Trentesaux D., Thomas A., Cardin O. (eds) Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol 762. Springer, Cham, 2018. - P. 489-500. ISBN 978-3-319-73750-8. DOI:

