ISSN 2079-0023 (print)
ISSN 2410-2857 (online)

UDK 004.827

D. E. DVUKHGLAVOV, V. E. KULYNYCH

DEVELOPMENT OF SOFTWARE SOLUTION FOR BUILDING ROUTE OF A ORDERS GROUP DELIVERY IN PRESENCE OF TIME CONSTRAINTS

The problem of determining route of visiting several points is considered. The task differs from known ones that time for arrival at each point is specified. The tasks of these class are solved in courier delivery services of various goods types. Basis of proposed algorithm for determining delivery route is algorithm for forming the route tree used matrix, which specify distances between delivery points, which is supplemented by checking conditions for possibility of visiting points, according to defined delivery time vector. Various criteria for selecting vertices for inclusion in route are considered. During developing software that implements proposed algorithm, used parallel computation methods that allow to obtain a solution for problem of sufficiently large dimension at acceptable time.

Keywords: route planning in presence of time constraints, Hamiltonian contour, parallel computing.

Розглядається задача визначення маршруту відвідування декількох пунктів за наявності заданого часу прибуття в кожний пункт. Основою алгоритму рішення, що пропонується, складає алгоритм формування дерева маршрутів на основі визначеної матриці часу переміщення між пунктами, який був доповнений перевіркою умов можливості відвідування пунктів. При цьому пропонуються різні критерії вибору вершин для включення в маршрут. Під час розробки програмного забезпечення, яке реалізує запропонований алгоритм, використані методи паралельних обчислень.

Ключові слова: планування маршруту за наявності обмежень за часом, Гамільтонів контур, паралельні обчислення.

Рассматривается задача определения маршрута посещения нескольких пунктов при наличии заданного времени прибытия в каждый пункт. Основою предлагаемого алгоритма решения составляет алгоритм формирования дерева маршрутов на основе заданной матрицы времени перемещения между пунктами, который дополнен проверкой условий возможности посещения пунктов. При этом предлагаются различные критерии выбора вершин для включения в маршрут. При разработке программного обеспечения, реализующего предложенный алгоритм, использованы методы параллельных вычислений.

Ключевые слова: планирование маршрута при наличии временных ограничений, Гамильтонов контур, параллельные вычисления.

Introduction. Especially popular now are services "delivered to door". Now to buy something, it's enough to call phone, make your choice to manager, and receive your order after some time through from hands of delivery service employee. The same can be done if you go to website of an online store or delivery service. The list of items that can be ordered today includes mail and parcels, food and water, clothing and equipment. In any case, it is modern, it is convenient, it does not require time and effort.

In turn, in the context of an increasing flow of such orders, there is question about quality of delivery service. The quality should be understood as the absence of damage and delivery timeliness. It is really difficult to keep packaging due to randomness of influencing factors. But ensuring delivery at right time is important task for this service segment.

Timeless delivery is impossible without careful planning, which, in turn, should be ensured by high-quality information provision. Currently, route planning systems are powerful systems that receive data from geographic information services such as Google Maps and GIS2, which provide information about location of infrastructure and roads, and routing directions (or multiple) from one point to other. An example of such systems is "4logist" [1], "ABMcloud" [2], "Anthill Logistics" [3], and others.

Meanwhile, algorithmic provision of such systems can be improved. In general case, delivery manager must make delivery route plan which will be minimal. Such task in scientific world was called "The Traveling Salesman Problem". Solutions and algorithms for solving this problem are known too. Meanwhile, in a number of delivery spheres planning should take into account an extra mandatory condition – delivery of orders at specified time. Such task often arises when delivering a group of "food" orders (pizza or sushi). And in such arrangement of implemented solutions in known software products there etc.

The purpose of article. Direction of improvement of known forming routes methods is account of specified delivery deadlines. This article presents method of solving problem in formulated initial conditions and results of software implementation development of this method.

Problem formulation and formalization. In general case, there is a source from which goods or services are delivered by customers. Depending on product which will be delivered, it may be a wholesale warehouse, a production shop, a bakery or a kitchen shop, a postal terminal. Future in this article such point will be called "Logistics Center" (LC).

The meaningful resolution of this problem looks like this. There are orders in LC, each of which must be delivered to their destination. For each order there is a deadline – time not later than order must be delivered. Time required to transfer from LC to destination points and times required to transfer between points is calculated and stored in memory at stage of system prepare. It is necessary to find orders delivery sequence where number of orders with violated time term will be minimal.

For formalized statement of task, following notation and variables must be entered.

Let n – the number of orders. Then designate the delivery time for a point i as follows t_i^D, where $i=\overline{1,n}$. Matrix $\{t_{ij}\}_{i=0,j=0}^{n,n}$, will consist of the elements t_{ij} – which is the time to move from point i to point j. An example of distance matrix for three points and LC is presented in Table 1.

Table 1 - Example of displacement matrix for 3 points and LC

	LC	1	2	3
LC	0	t_{01}	t_{02}	t_{03}
1	t_{10}	0	t_{12}	t_{13}
2	t_{20}	t_{21}	0	t_{23}
3	t_{30}	t_{31}	t_{32}	0

The path matrix has following properties:

– each element of main diagonal contains zeros since path from point to itself is zero;

– path matrix is symmetric matrix, since travel time from point i to point j equals time of moving from point j to point i.

Following restrictions determine the conditions for constructing route. Let x_i be the order number executed by i-th account, where $i=\overline{1,n}$. Next condition is imposed on these variables:

$$x_i \in \{1,2,\ldots,n\}, i=\overline{1,n} \quad (1)$$

$$x_i \neq x_j, i \neq j. \quad (2)$$

This restriction ensures that all points are bypassed, and in this case, none of items can be visited more than once. Such conditions correspond to known travel salesman problem (The Traveling Salesman Problem) [4], in which it is necessary to bypass all points from a definite list exactly once and return to start point (in our case – to LC).

The criterion of optimality of this task should ensure that timely delivery of order is taken into account. Let $y(i,x)$ be number that x_i will be executed, where $i=\overline{1,n}$. Then denote by $z_{y(i,x)}$ – time at which order will be delivered to point i. Formula for calculating $z_{y(i,x)}$ is presented as expression (3):

$$z_{y(i,x)} = t_{0x_1} + \sum_{j=1}^{y(i,x)} t_{x_j x_{j+1}}. \quad (3)$$

Finally, let denote by $w_{y(i,x)}$ a binary value equal to 0 if directive time of order is not violated and 1 in otherwise. The formula for calculating $w_{y(i,x)}$ is presented as expression (4):

$$w_{y(i,x)} = \begin{cases} 0, \text{if } z_{y(i,x)} \le t^D_{y(i,x)}; \\ 1, \end{cases}, \ i=\overline{1,n}. \quad (4)$$

Using introduced notation, it is possible to formulate an optimality criterion for the problem. This criterion is as follows (expression (5)):

$$\sum_{i=1}^{n} w_i \to min. \quad (5)$$

Thus, problem for solve consists in minimizing target function represented by expression (5) under conditions (1) and (2).

Solution of this problem is to find Hamiltonian cycles on a graph, that is closed paths that bypass all vertices in a graph exactly once. Methods for solving such problem without accounting time criterion are known (for example [5]). Most popular of them is method of branches and boundaries and method of successive improvement of solution. Meanwhile, they are not fully suited to solution of this task.

Algorithm for determining delivery route with account time constraints. Proposed approach is to consider all possible simple paths that begin at arbitrarily selected starting point, until a Hamiltonian cycle is found or all possible paths will not be explored. Such scheme of operations became basis of algorithm for determining delivery route in presence of certain delivery deadlines, idea of which is presented in [6]. Consider steps of algorithm.

Each of branch tree vertices will be represented in $v=(\alpha_1,\alpha_2,\ldots,\alpha_k)$, where k is vertex level, path length in branch tree from root to given vertex (i.e. sequence of distance from LC to items starting with α_1 and ending with α_k). When $k=0$, it is necessary to designate vertex as $v=()$. Also denote by $B(v)$ and $H(v)$ corresponding upper and lower bounds of v; by V – vertices v.

Sequence of algorithm actions is next.

Step 1. Initialize branch tree $V=\{(1),(2),\ldots,(n)\}$ and go to step 2.

Step 2 (stop condition). At this step stop check is performed like (6):

$$|V|=1 \text{ and } B(v)=H(v). \quad (6)$$

If this condition is met, then solution is found. Optimal path is path v.

Step 3 (branching). Define $v=Branching(V)$. Define the set $\beta=\{1,2,\ldots,n\}\setminus v$. Next define new set $V=V\setminus\{v\}\cup\{v,\beta_i\}, i=\overline{1,n-k}$ (i.e. excludes vertex v and add its descendants). Go to step 4.

Step 4 (clipping). For each pair of vertices $v',v''\in V, v'\neq v''$ verify condition (7):

$$B(v') \le H(v'). \quad (7)$$

When the condition is fulfilled, perform following operation (see (8)):

$$V=V\setminus\{v''\}. \quad (8)$$

Then go to step 2.

The $Branching(V)$ function allows to find among the options best vertex for fastest way to get decision result.

The $Branching(V)$ function can be implemented in several ways. During work, there are three options of branching that differ in inclusion vertices rules in ways

tree may considered. In this case, in all options, sections of way that lead to vertices that are already included in way are not included.

The search strategy "Breadth". In this variant $Branching(V)=v$ such that $v=(\alpha_1,\alpha_2,\ldots,\alpha_k)$, $v'=(\alpha_1',\alpha_2',\ldots,\alpha_k')\in V$ and condition $k\le k'$ is satisfied, i.e. choose vertex whose last element index is smallest.

Optimistic strategy. In this variant $Branching(V)=v$ such that $v=(\alpha_1,\alpha_2,\ldots,\alpha_k)$, $v'=(\alpha_1',\alpha_2',\ldots,\alpha_k')\in V$ and condition $H(v)\le H(v')$ is satisfied, i.e. choose vertex whose high estimate of delivery time is better.

Realistic strategy. In this variant $Branching(V)=v$ such that $v=(\alpha_1,\alpha_2,\ldots,\alpha_k)$, $v'=(\alpha_1',\alpha_2',\ldots,\alpha_k')\in V$ and the condition $B(v)\le B(v')$ is satisfied, i.e. choose vertex whose below estimate of delivery time is better.

Function $B(v)$ determines upper estimate of optimality criterion for route v. Algorithm for determining value $B(v)$ contains following steps.

Step 1. Put $x_i=\alpha_i, i=\overline{1,k}$, if $k=n$, go to step 3; otherwise put $j=0$ and go to step 2.

Step 2. Mark $\beta=\{1,2,\ldots,n\}\setminus\{x_1,x_2,\ldots,x_{k+j}\}$. Delivery time for this way will be denoted by z. Formula for calculation is represented by expression (9):

$$z=t_{0x_1}+\sum_{i=2}^{k}t_{x_ix_i+1}. \qquad (9)$$

Next calculate values of w_{β_i} in according (10):

$$w_\beta=\begin{cases}t^D_{x_{k+j}\beta_i}-(z+t_{x_{k+j}\beta_i})\,\Big|\,z+t_{x_{k+j}\beta_i}<t^D_{\beta_i};\\ +\infty,\end{cases}$$

$$i=\overline{1,n-(k+j)}. \qquad (10)$$

Next determine value of h in according (11):

$$h=\underset{i=\overline{1,n-(k+j)}}{\operatorname{argmin}} w_{\beta_i} \qquad (11)$$

Then put $j=j+1$, $x_{k+j}=\beta_h$. If $k+j=n$, go to step 3, otherwise - to step 2.

Step 3. Determine delivery time for each point of vertex v as follows (see (12)) and then go to step 4:

$$z_1=t_{0x_1}; \qquad (12)$$

$$z_i=z_{i-1}+t_{x_{i-1}x_i}, i=\overline{2,n}.$$

Step 4. For each z_i, make comparison with directive time and write 0, if directive is not violated, and 1, otherwise (see (13)):

$$w_i=\begin{cases}0\,\Big|\,z_i<t^D_{x_i};\\ 1.\end{cases} \qquad (13)$$

Then go to step 5.

Step 5. The estimate of $B(v)$ will be (expression (14)):

$$B(v)=\sum_{i=1}^{n}w_i. \qquad (14)$$

Function $H(v)$ determines upper estimate of optimality criterion for route v.

Algorithm for determining value $H(v)$ contains following sequence of actions.

Step 1. Put $x_i=\alpha_i, i=\overline{1,k}$. If $k=n$, go to step 5, otherwise – go to step 2.

Step 2. Mark $\beta=\{1,2,\ldots,n\}\setminus\{x_1,x_2\ldots,x_{k+j}\}$. Go to step 3.

Step 3. Determine delivery time for each point of v according to expression (15), then go to step 4:

$$z_i=t_{0\alpha_1}+\sum_{j=1}^{k-1}t_{\alpha_j\alpha_{j+1}}, i=\overline{1,k}; \qquad (15)$$

$$z_i=t_{0\alpha_1}+\sum_{j=1}^{k-1}t_{\alpha_j\alpha_{j+1}}+t_{\alpha_k\beta_i}, i=\overline{k+1,n}.$$

Step 4. For each z_i make comparison with directive time (look expression (10)), then go to step 5:

Step 5. Estimate $H(v)$ will take as follow (expression (16)):

$$H(v)=\sum_{i=1}^{n}w_i. \qquad (16)$$

This problem belongs to class of NP-difficult problems [4]. Thus, there is probability that with increase in number of delivery points, time for solving problem will not be tolerable. Therefore, to reduce running time of program implementing presented algorithm proposed to apply parallel computing.

Using parallel computations. Parallel computations are calculations that can be implemented on multiprocessor systems with possibility of simultaneous execution of many actions generated by decision process one or many tasks of one project [7].

Partitioning input data into groups can be done in different ways [8]. The first variant of data partitioning is to divide input data array into p parts, and executing on each of processors processing on data from corresponding segment. This approach take name “segment algorithm”. The other strategy involves executing on each of processors processing on data from input data array, which placed on p position apart. This approach takes name “step algorithm”.

Presented algorithm was make parallel. For reduce time of solving problem of determining delivery route, proposed to distribute initial set V to several subsets $V_s, s=\overline{1,p}$, depending on number of cluster processors or kernel of computer p. Thus, p sets of nodes will be

obtained, which will be initial for constructing delivery routes. Each subset (segment) will consist of $k = \left\lceil \frac{n}{p} \right\rceil$ elements. Thus, subsets of following composition will be formed:

$$V_1 = \{(1),(2),...(k)\};$$

$$V_2 = \{(k+1),(k+2),...(2 \cdot k)\};$$

…

$$V_s = \{((s-1) \cdot k + 1),((s-1) \cdot k + 2),...(s \cdot k)\};$$

…

$$V_p = \{((p-1) \cdot k + 1),((p-1) \cdot k + 2),...(n)\}.$$

It should be noted that last set may be less than other cases in which number of elements does not divide whole on p. Now for each subset you need to find best way to deliver orders. After finding best paths it is necessary to compare decision results of each task. Index of best solution can has defined from expression (17):

$$s\_opt = \underset{s=\overline{1,p}}{\arg\min}\, w_s. \quad (17)$$

Thus, v_{s_opt} will be best way to deliver orders.

Developed algorithms became the basis for developing software implementation.

Developing software implementation of algorithm. Program development is carried out according to modern approaches to software design [9].

At initial stages, some UML diagrams were developed, use of which allowed to significantly accelerate development process [10]. The most interesting of them is a classes diagram of, which is presented in Fig. 1.

In development of this diagram were thought-out classes that allow to search solution using different branching strategy and different variants of calculations organization.

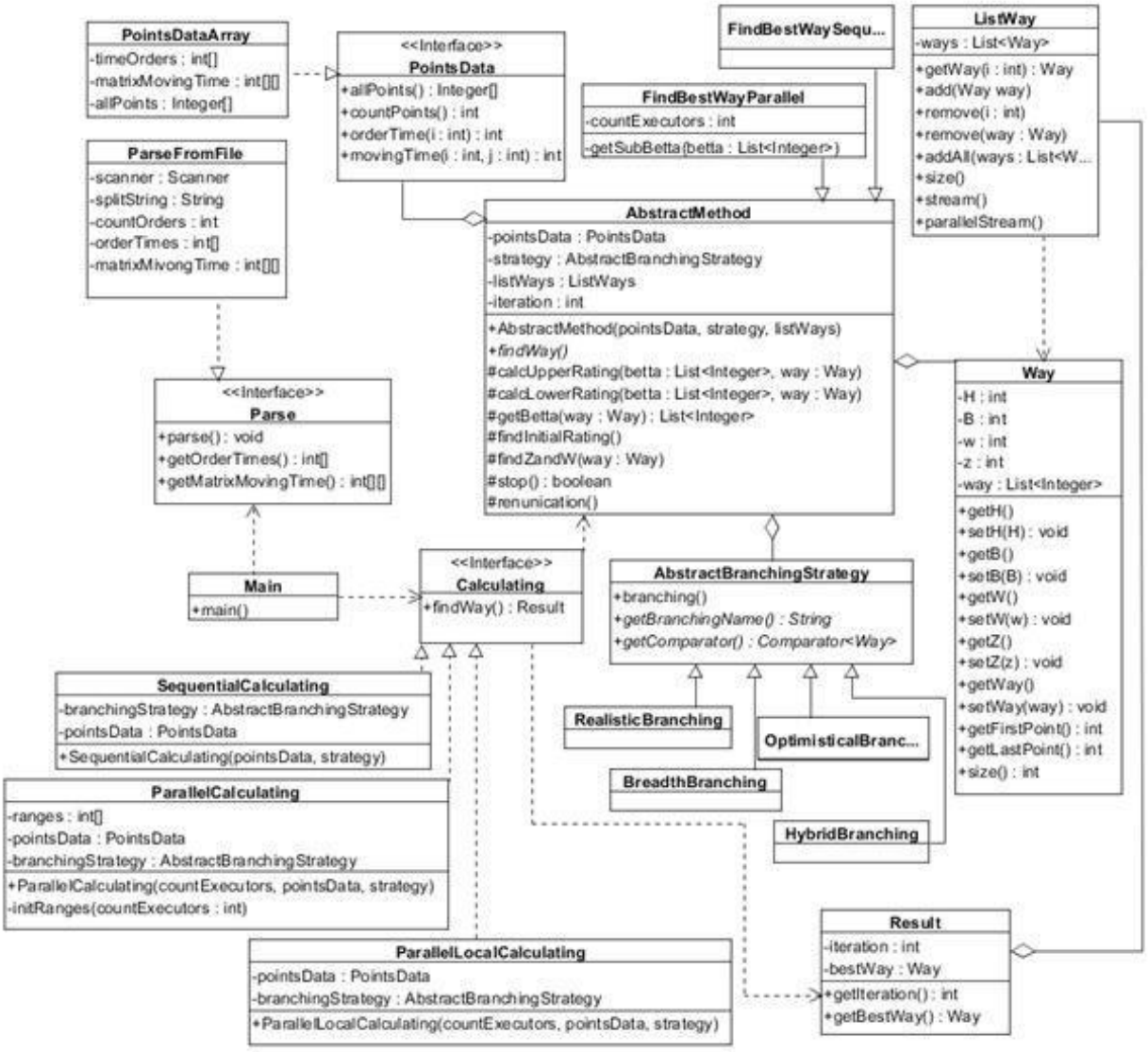


Fig. 1 – Classes diagram

ISSN 2079-0023 (print)
ISSN 2410-2857 (online)

This approach to structuring the program meets the best techniques for realizing principles of object-oriented design. The advantage of this variant of building program structure is in getting possibility of increasing number of methods of branching implementing and number of approach to organization of calculations without changing the basic algorithm for determining the route delivery orders. For this purpose it serve abstract classes AbstractBranchingStrategy and Calculating. Also on fiq. 1 you can see class HybridBranching which in future will allowed realize hybrid strategy of branching.

To study algorithm properties, software implementation of algorithm was developed that represents console application in Java programming language created in integrated InteliJ Idea environment.

Research of time characteristics of program implementation. For testing files with several delivery examples with different number of items were created (based on examples from [6]). Research of program implementation carried out in two directions:

– research dependence time of program executing from strategy of branching in defining route process;

– research dependence time of program executing from approach to organization of computations.

The results were obtained for the three different branching strategies proposed above: breadth, optimistic or realistic.

Also various technologies of computing organization consider has been investigated:

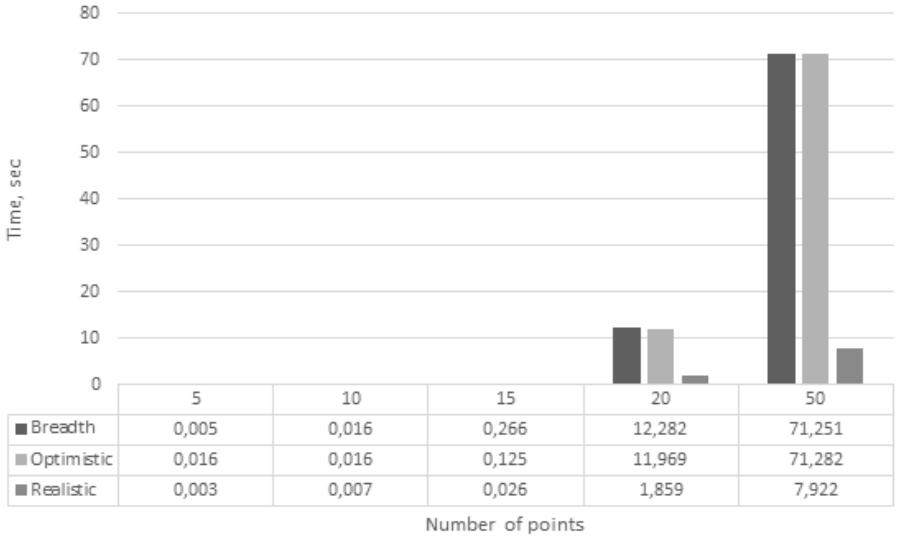
– sequential calculation;

– parallel calculations with use 2 threads;

– parallel calculations with use 4 threads.

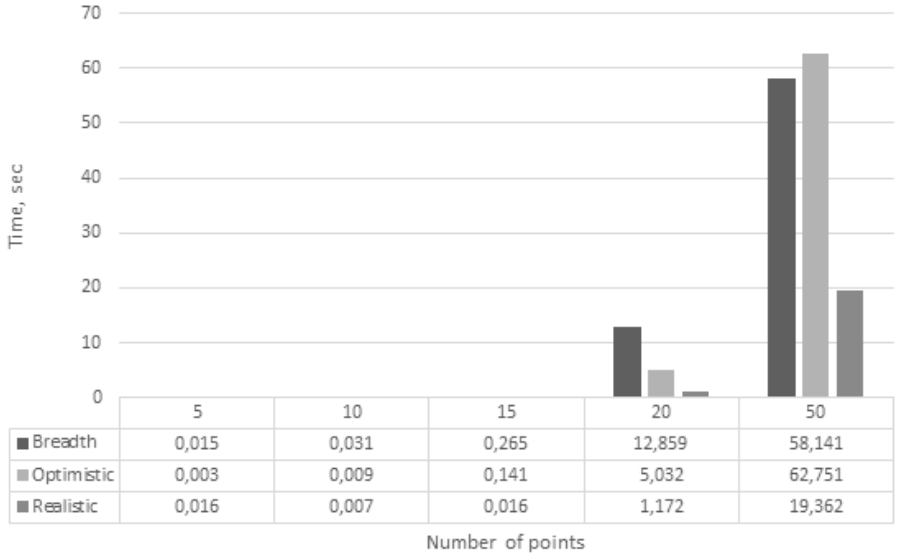
For program testing used computer, which has next configuration: processor – Intel Core i5-4440, 3.1GHz; number of kernels – 4; RAM – 8Gb. This configuration can be considered as "average modern computer", which can be used for automating in planning of delivery orders.

Times to receive delivery routes by use developing program for different output options (different delivery points) are presented on fig. 2–4.



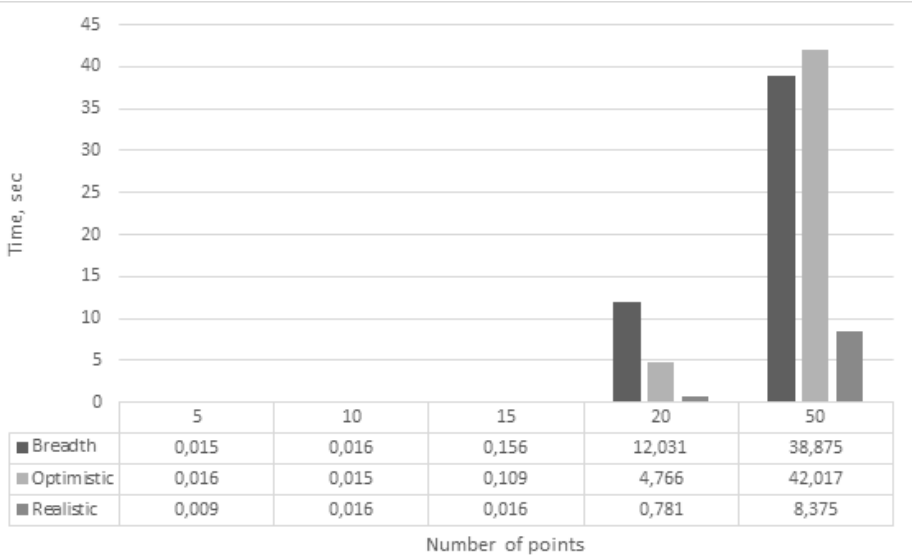
	5	10	15	20	50
Breadth	0,005	0,016	0,266	12,282	71,251
Optimistic	0,016	0,016	0,125	11,969	71,282
Realistic	0,003	0,007	0,026	1,859	7,922

Fig. 2 – Processing time for different branching strategy when using sequential calculations



	5	10	15	20	50
Breadth	0,015	0,031	0,265	12,859	58,141
Optimistic	0,003	0,009	0,141	5,032	62,751
Realistic	0,016	0,007	0,016	1,172	19,362

Fig. 3 – Processing time for different branching strategy when using parallel calculations (2 threads)



	5	10	15	20	50
Breadth	0,015	0,016	0,156	12,031	38,875
Optimistic	0,016	0,015	0,109	4,766	42,017
Realistic	0,009	0,016	0,016	0,781	8,375

Fig. 4 – Processing time for different branching strategy when using parallel calculations (4 threads)

If make conclusions only from data presented in tables then it can be argued that software implementation allows you to get a solution for 15 delivery points in less than 1 sec for any approach to computations. But implementation time grows factorially with growth of problem dimension and this thesis has been confirmed. Also in according to results best results shows using realistic strategy. When use realistic strategy of branching maximal time was near 20 sec. It can consider as good result.

On Fig. 4–6 results are presented in depends of method of calculations organization.

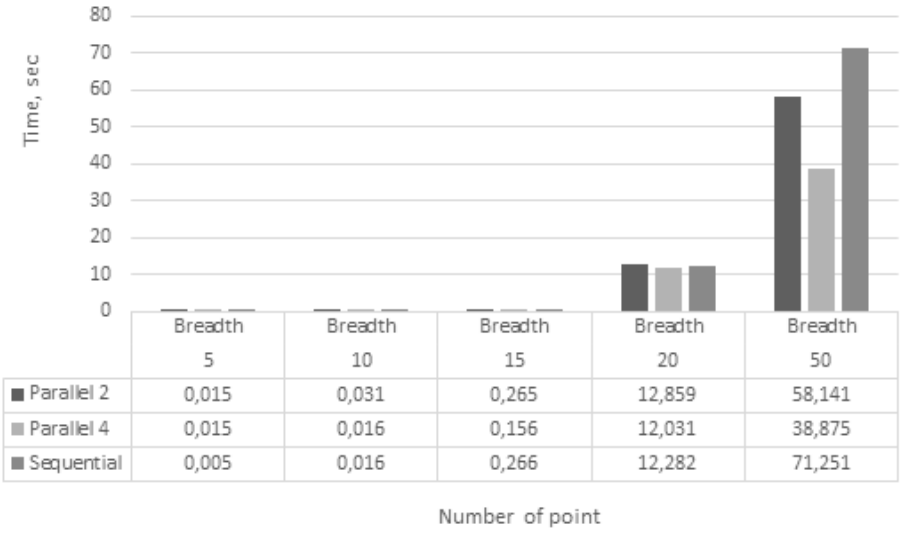
This representation should provide an idea of expediency of using parallel computing. Two variants of parallel computations are considered – using 2 or 4 threads. These are typical variants for modern computers. More computers are a dual core. In addition, in operating system Windows which installed on more computers there can create two virtual kernels.

Such representation once again confirms that solution for route is no more than 15 delivery points will be received in less than 1 sec without dependence of usage approach to computations. But such review of results did not give clear answer on expediency of parallel computing.

If look at results for 20 delivery points, then for breadth strategy results are approximately equal. But for other two strategies, the worst result was for case of using 4 threads, i.e. parallel computations slowed output almost twofold. Using 2 threads allowed to get result not much faster than sequential calculations.

When planning delivery for 50 cases, use of parallelism is more advisable. But in this case use of two threads gives results much faster. However, for a realistic strategy, it turned out to be more efficient to use 4 threads to parallelize computations.

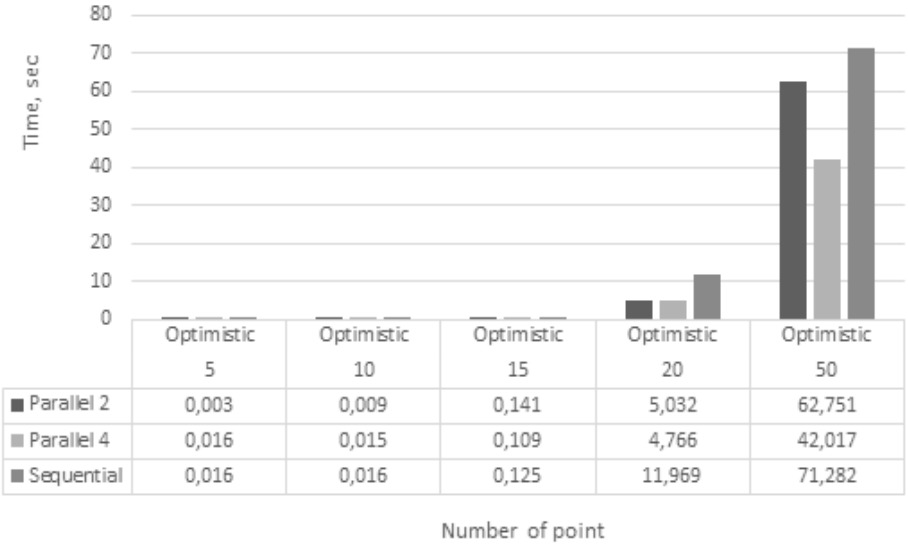
Obtained results give a possible make some conclusions on work.



	Breadth 5	Breadth 10	Breadth 15	Breadth 20	Breadth 50
Parallel 2	0,015	0,031	0,265	12,859	58,141
Parallel 4	0,015	0,016	0,156	12,031	38,875
Sequential	0,005	0,016	0,266	12,282	71,251

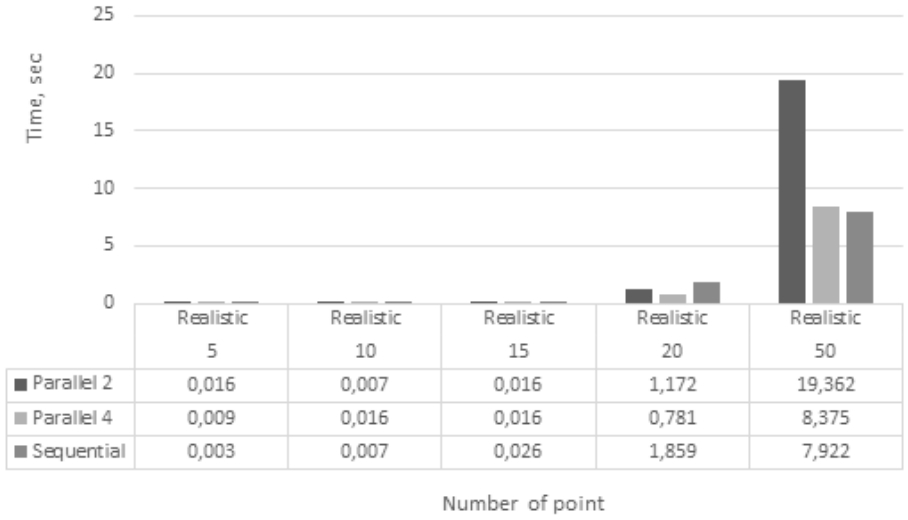
Fig. 5 – Processing time for different type of calculation when using branching strategy «Breadth»

ISSN 2079-0023 (print)
ISSN 2410-2857 (online)



	Optimistic 5	Optimistic 10	Optimistic 15	Optimistic 20	Optimistic 50
Parallel 2	0,003	0,009	0,141	5,032	62,751
Parallel 4	0,016	0,015	0,109	4,766	42,017
Sequential	0,016	0,016	0,125	11,969	71,282

Fig. 6 – Processing time for different type of calculation using branching strategy «Optimistic»



	Realistic 5	Realistic 10	Realistic 15	Realistic 20	Realistic 50
Parallel 2	0,016	0,007	0,016	1,172	19,362
Parallel 4	0,009	0,016	0,016	0,781	8,375
Sequential	0,003	0,007	0,026	1,859	7,922

Fig. 7 – Processing time for different type of calculation when using branching strategy «Realistic»

Conclusions. Presented results testify to practical significance of presented work, which consists in possibility of using this algorithm for forming of route in delivery planning systems in case of presence of defined delivery deadlines. Main strategy for branching should be considered a realistic branching strategy, which is based on worst forecast of timely delivery of orders. Concerning use of parallel computations, there is no unambiguous conclusion. For small number of delivery points, a solution at acceptable time can be obtained with traditional sequential processing. For large number of delivery points, the use of parallel computations requires a study on more examples.

References

1. CRM программа для автоматизации экспедиторов и перевозчиков «4logist». – Available at : www.4logist.com. – Accessed : 20th of October 2017.
2. Система управления транспортом «ABMcloud». – Available at : www.abmcloud.com. – Accessed : 20th of October 2017.
3. Web-сервис «Муравьиная логистика». – Available at : https://ant-logistics.com/main.html. – Accessed : 20th of October 2017.
4. *Зайченко Ю. П.* Исследование операций / *Ю. П. Зайченко* – Киев : Слово, 2003. – 688 с.
5. Balakrishnan R. A Textbook of Graph Theory / *R. Balakrishnan, K. Ranganathan.* – Springer, 2012. – 305 p.
6. *Афраймович Л. Г.* Учебно-методическая разработка «Информационные технологии в области принятия решений. Часть 1» / *Л. Г. Афраймович.* – Нижний Новгород : Нижегородский госуниверситет, 2016. – 27 с.
7. *Михалевич В. С.* Словарь по кибернетике / *В. С. Михалевич* – К. : Глав. ред. Укр. Сов. Энциклопедии им. М. П. Бажана, 1989. – 751 с.
8. *Воеводин В. В.* Параллельные вычисления / *В. В. Воеводин, Вл. В. Воеводин.* – СПб. : БХВ-Петербург, 2002. – 608 с.
9. *Орлов С. А.* Технологии разработки программного обеспечения. Современный курс по программной инженерии. 4-е изд. // *С. А. Орлов, Б. Я. Цилькер.* – СПб. : Питер, 2012. – 608 с.
10. *Буч Г.* Язык UML. Руководство пользователя: пер. с англ., 2-е изд. // *Г. Буч, Дж. Рамбо, А. Якобсон.* – М. : ДМК Пресс, 2006. – 496 с.

References (transliterated)

1. *CRM programma dlya avtomatizatsii ekspeditorov i perevozchikov «4logist»* [CRM program for the automation of forwarders and carriers «4logist»]. Available at : www.4logist.com (accessed 20.10.2017).

2. *Sistema upravleniya transportom «ABMcloud»* [Transport management system «ABMcloud»]. Available at : www.abmcloud.com (accessed 20.10.2017).
3. *Web-servis «Murav'inaya logistika»* [Web-service «Ant logistics»]. Available at : https://ant-logistics.com/main.html. (accessed 20.10.2017).
4. Zaychenko YU. P. *Issledovaniye operatsiy* [Operations research]. Kiyev, Slovo, 2003, 688 p.
5. Balakrishnan R., Ranganathan K. *A Textbook of Graph Theory*. Springer, 2012, 305 p.
6. Afraymovich L. G. *Uchebno-metodicheskaya razrabotka «Informatsionnyye tekhnologii v oblasti prinyatiya resheniy. Chast' 1»* [Educational material «Information technologies in the decision-making sphere. Part 1»]. Nizhniy Novgorod, Nizhegorodskiy gosuniversitet, 2016, 27 p.
7. Mikhalevich V. S. *Slovar' po kibernetike* [Dictionary on Cybernetics]. Kiyev, Glav. red. Ukr. Sov. Entsiklopedii im. M. P. Bazhana, 1989, 751 p.
8. Voyevodin V. V., Voyevodin Vl. V. *Parallel'nyye vychisleniya* [Parallel calculations]. St.-Petersburg, BKHV-Peterburg, 2002, 608 p.
9. Orlov S. A., Tsil'ker B. YA. *Tekhnologii razrabotki programmnogo obespecheniya. Sovremennyy kurs po programmnoy inzhenerii. 4-ye izd* [Software development technologies. Modern course in software engineering]. St.-Petersburg, Piter, 2012, 608 p.
10. Buch G., Rambo Dzh., Yakobson A. *YAzyk UML. Rukovodstvo pol'zovatelya: per. s angl. 2-ye izd* [The Unified Modeling Language. Usere Guide. Second Edition]. Moscow, DMK Press, 2006, 496 p.

Received 16.11.2017

Бібліографічні описи / Библиографические описания / Bibliographic descriptions

Разработка программного решения для построения маршрута доставки группы заказов при наличии временных ограничений / Д. Е. Двухглавов, В. Є. Кулинич // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Х. : НТУ «ХПІ», 2017. – № 55 (1276). – С. 64–71. – Бібліогр. : 10 назв. – ISSN 2079-0023.

Разработка программного решения для построения маршрута доставки группы заказов при наличии временных ограничений / Д. Э. Двухглавов, В. Е. Кулинич // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків : НТУ «ХПІ», 2017. – № 55 (1276). – С. 64–71. – Библиогр. : 10 наим. – ISSN 2079-0023.

Development of software solution for building route of a orders group delivery in presence of time constraints / D. E. Dvukhglavov, V. E. Kulynych // Bulletin of National Technical University "KhPI". Series: System analysis, control and information technology. – Kharkov : NTU "KhPI", 2017. – No. 55 (1276). – P. 64–71. – Bibliogr. : 10. – ISSN 2079-0023.

Відомості про авторів / Сведения об авторах / About the Authors

Двухглавов Дмитро Едуардович – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри «Програмна інженерія та інформаційні технології управління»; тел.: (095) 120-30-66; e-mail: ddimae72@gmail.com.

Двухглавов Дмитрий Эдуардович – кандидат технических наук, доцент, Национальный технический университет «Харьковский политехнический институт», доцент кафедры «Программная инженерия и информационные технологии управления»; тел.: (095) 120-30-66; e-mail: ddimae72@gmail.com.

Dvukhglavov Dmytro Eduardovych – Candidate of Technical Sciences (Ph. D.), Docent, National Technical University "Kharkiv Polytechnic Institute", Associate Professor at the Department «Software engineering and management information technology»; tel.: (067) 839-12-41; e-mail: ddimae72@gmail.com.

Кулинич Вадим Євгенійович – Національний технічний університет «Харківський політехнічний інститут», студент; тел.: (099) 377-18-04; e-mail: kulinichvadim29@gmail.com.

Кулинич Вадим Евгениевич – Национальный технический университет «Харьковский политехнический институт», студент; тел.: (099) 377-18-04; e-mail: kulinichvadim29@gmail.com.

Kulynych Vadim Evgenijovych – National Technical University "Kharkiv Polytechnic Institute", student; tel.: (099) 377-18-04; e-mail: kulinichvadim29@gmail.com.