UDC 004.8
MSC 68T50

Applied routing problem for a fleet of delivery drones using a modified parallel genetic algorithm*

A. Markelova, A. Allahverdyan, A. Martemyanov, I. Sokolova, O. Petrosian, M. Svirkin

St Petersburg State University, 7–9, Universitetskaya nab., St Petersburg, 199034, Russian Federation

For citation: Markelova A., Allahverdyan A., Martemyanov A., Sokolova I., Petrosian O., Svirkin M. Applied routing problem for a fleet of delivery drones using a modified parallel genetic algorithm. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2022, vol. 18, iss. 1, pp. 135–148.
https://doi.org/10.21638/11701/spbu10.2022.111

More and more experts agree that in the near future, most freight traffic will be carried out using automated systems, and of them drone delivery is considered to be the most promising. Drone delivery would benefit by independence from the limitations of transport infrastructure and road conditions and would ensure cargo delivery with rapid turnaround times, as well as a significant reduction of environmental impact. The technical capabilities of unmanned aerial vehicles improve year by year, so the task of coordinating drones and effectively planning routes is relevant and in great demand. The development of such technologies will help reduce transportation costs and improve customer service through faster delivery. This article discusses the applied routing problem for a fleet of drones with limited load capacity for the delivery of heterogeneous goods with the possibility of loading in multiple warehouses from an international optimization competition. The solution includes new approach based on a mixed dimensional parallel genetic algorithm (MDPGA) for finding rational routes for delivering goods to various customers and an assignment problem to reduce the dimension depending on the number of warehouses.

Keywords: drone delivery, scheduling, genetic algorithm, vehicle routing problem, multi-depot, multi-trip, multi-product, split-delivery.

1. Introduction. With technological progress in robotics and advanced logistics management, transport companies and trade organizations with large turnover are searching for alternative ways to transport goods in pursuit of minimizing costs, reducing delivery time, and reducing environmental emissions. Unmanned aerial vehicles (UAVs), or drones, are becoming a promising working tool. Trade and transportation companies such as Amazon Inc., UPS, Deutsche Post AG, and Google have invested in research projects to use drones to optimize their distribution network [1]. The advantages of drone delivery have prompted researchers to invest in industrial projects related to the use of drones in the transport sector [2, 3].

The goal of this research is to propose solutions that apply to the management of an unmanned aerial vehicle routing problem (UAVRP) which approximates real-world production conditions. Therefore, this tool will be useful for an industrial partner. UAVRP in our statement is formulated as a split delivery capacitated vehicle routing problem with multi-trip, multi-depot, and multi-product components. Figure 1 shows a diagram of the problem.The paper proposes a modified approach based on a parallel genetic algorithm.

* This work was carried out under the auspices of a grant of the President of the Russian Federation for state support of young Russian scientists — candidates of science (project N MK-4674.2021.1.1).

The results obtained show further prospects for research in this direction and optimize the distance traveled by drones on average by 10–30 % (compared to the greedy solution).

Many large companies are interested in the development of technologies and algorithms in this area. So competitions are organized that can unite the efforts of researchers to solve UAVRP. For the first time the developed a mixed dimensional parallel genetic algorithm was presented as a part of a competition from Google on the Kaggle contest management platform [4]. This competition provided data on a fleet of unmanned aerial vehicles, a list of customer orders that included different types of products with different weights, as well as a limited number of individual products in several warehouses. The goal was formulated as the task of planning drone operations to reduce the delivery time and distance traveled. Due to the complex mathematical formulation, a very large dimension problem arises that cannot always be solved by the conventional optimization methods [5]. Other proposals by competition participants involved solutions based on sequential optimization and greedy algorithms.

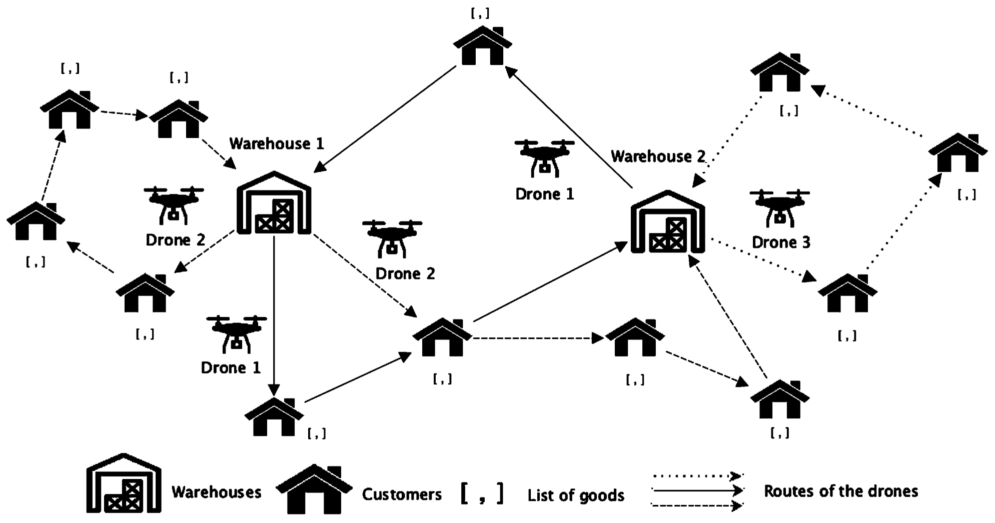

Fig. 1. A case of multi-depot UAVRP with multi-trips and multi-products

The literature presents various variants of the problem statement, as well as various approaches to its solution and algorithms. The researcher's approaches are based on problems with different variants of the classical VRP. However, there are significant differences that must be taken into account when building a mathematical model and selecting algorithms such as the limited payload capacity of drones and the energy capacity of batteries. Therefore, several approaches are used that successfully cope with these constraints. For instance, a vehicle routing problem with drones (VRPD) considers a delivery scenario based on a combination of drone delivery with a delivery truck [6]. The green vehicle routing problem (GVRP) allows vehicles to visit refueling stations when making deliveries [7]. In our problem statement, these approaches are unsatisfactory so we use an approach that reuses vehicles and is known as the multi-trip vehicle routing problem (MTVRP), first proposed by Fleishman [8]. Although there are several approaches for solving MTVRP, such as the large neighborhood search algorithm [9], the hybrid genetic algorithm with a local search operator [10], the variable neighborhood search algorithm [11], and the branching and price algorithm, they do not allow for solving problems on an industrial scale, the dimensions of which reach several thousand orders. The listed articles use similar and not only numerical optimization algorithms [12–14].

Owing to the complex structure of the problem and large dimensions, the preponderance of the literature recommends a genetic algorithm as the basis of a solution that is adapted to the proper mathematical formulation and modified to work with a very large dimension. Genetic algorithms are currently one of the most promising areas in the field of artificial intelligence as part of so-called evolutionary methods and were originally proposed by John Henry Holland in the 1970s. This technique has as its basic structure the original population, carrying out the process of selection, crossing or recombination, the stage of mutation, and the stage of substitution in the original population, resulting in a global result of the evolution of the population. For the computational approach, genetic algorithms model the process of natural selection to solve problems in the field of mathematical optimization. Our modification of the algorithm consists in the absence of crossover and the means of obtaining convergence merely by counting mutations. Meanwhile, we have added the ability to change the dimension of individuals in order to search for optimal drone loading plans. To solve the large dimension problem, a radial approach was applied to artificially reduce the dimension of the problem without a reduction in quality.

The structure of the article includes 5 sections and is organized as follows. Section 2 provides detailed information about the competition and the test case being investigated and Section 3 a detailed description of the modified genetic algorithm for solving UAVRP. Section 4 includes an experimental study with an analysis of the results obtained for a given test case. Section 5 provides conclusions and prospects for further research.

2. Competition. The competition named “Drone Delivery problem” is based on the platform of Kaggle [4], this problem was first released during Google’s 2016 Coding Competition — Hash Code. The task consisted of scheduling the drone operations so that orders were completed as soon as possible under the conditions of a particular list of customer orders and warehouse availability. The visualization of the test case is shown in Fig. 2.

Elements of simulation.

1. Map. The simulation takes place on a two-dimensional grid. Each cell is identified by a pair of integer coordinates $[r, c]$.

2. Products. There are several product types of different weights that are available in some warehouse inventory.

3. Warehouses. The warehouses have a certain number of products. During the simulation, no new product items appear in the warehouses.

4. Orders. Orders contain information about the product list and the locations of orders on the map. The order is considered fulfilled when all of the ordered product items are delivered. We can deliver product items using multiple drones, in any order.

5. Drones. The drones take the shortest path from one cell to another calculated as two-dimensional Euclidean distance. At the beginning of the simulation, all drones are at the warehouse with an id of 0.

Commands. Each drone can be given the following basic commands:

1. Load. Moves the number of items of the product type from a warehouse to the drone’s inventory. If the drone is not at the warehouse, it will fly there using the shortest path.

2. Deliver. Delivers the number of items of the product type to a customer. If the drone is not at the destination, it will fly there using the shortest path.

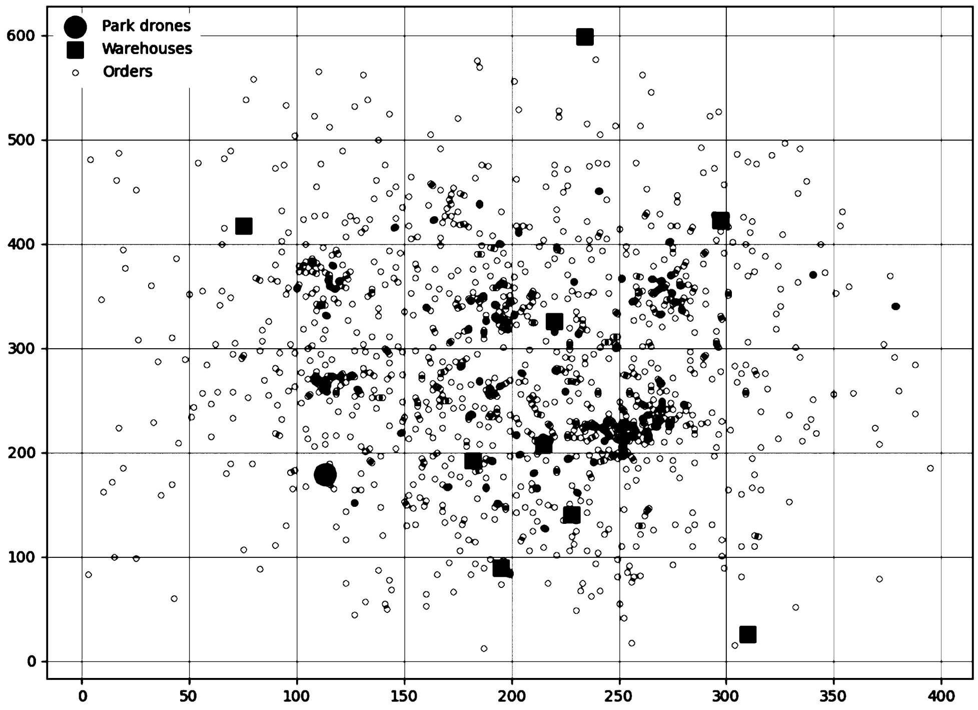

Fig. 2. Test case visualization

Simulation. The simulation proceeds in T turns, from 0 to $T-1$. A drone executes the commands issued to it in the order in which they are specified, one by one. The first command issued to the drone starts at turn 0. The Load/Deliver command takes $d+1$ turns, where d is the distance traveled by the drone to perform the requested action.

Each completed order will earn between 1 and 100 points, depending on the turn in which it is completed. For an order completed in turn t and a simulation taking T turns, the score for the order is calculated as $\frac{(T-t)}{T \cdot 100}$, rounded up to the next integer. The score for a single data set will be the sum of the scores of all completed orders.

Input data set. The input data is provided as a text file containing ASCII characters with lines terminated with UNIX-style line endings. There are 400 product types numbered from 0 to 399, 10 warehouses numbered from 0 to 9, and 1250 orders numbered from 0 to 1249. The first section of the file describes the parameters of the simulation; map dimensions: 400 by 600, number of available drones: 30, deadline of the simulation: 112 993, maximum load of a drone: 200. The second describes the weights of the products available for orders. The third describes the warehouse's location on the map and the number of product types at them. A fourth of the file describes the customer location on the map and customer orders.

3. Mathematical optimization problem.

3.1. Indices.

Sets:

C: set of clients,
W: set of warehouses,
D: set of drones,
P: set of products,
R: set of trips.

Parameters:

n_d: number of drones,
n_o: number of orders,
n_w: number of warehouses,
n_p: number of products,
n_r: number of trips,
L: drone maximum flight length,
Q: drone carrying capacity,
dm_{pj}: demand of product p in order j,
M_{p_i}: weight of product p_i,
N_{pw}: number of product p in warehouse w,
d_{ij}: length of edge (i, j).

Decision variables:

$$x_{ij}^{rk} = \begin{cases} 1, & \text{if the drone } k \text{ on the trip } r \text{ moved from } j \text{ from } i, \\ 0, & \text{otherwise,} \end{cases}$$

$$y_{pjk}^{r} = \begin{cases} 1, & \text{if the drone } k \text{ on the trip } r \text{ to deliver the } p \text{ for } j, \\ 0, & \text{otherwise.} \end{cases}$$

3.2. Mathematical model. The objective function of the competition: $\sum_{j=1}^{n_o} \frac{T-t_j}{T} \times 100 \to \max$. But in order to simplify the problem statement and convert it to a classical mathematical problem statement we assume t_j to be a turn when the order j is completed and T (the number of turns of the simulation) and 100 as positive constant, so our function is converted into: $\sum_{j=1}^{n_o} -t_j \to \max$. Multiply by -1 and get: $\sum_{j=1}^{n_o} t_j \to \min$.

The following equations are given:

$$\sum_{i=0}^{n_o+n_w} \sum_{j=0}^{n_o+n_w} d_{ij} \sum_{k=0}^{n_d} \sum_{r=0}^{n_r} x_{ij}^{rk} \to \min, \tag{1}$$

$$\sum_{p=0}^{n_p} \sum_{j=0}^{n_o} y_{pjk}^{r} \leqslant Q \qquad \forall k \in D,\ \forall r \in R, \tag{2}$$

$$\sum_{k=0}^{n_d} \sum_{i=0}^{n_o} \sum_{r=0}^{n_r} x_{ij}^{rk} \geqslant 1 \qquad \forall j \in C,\ \forall k \in D, \tag{3}$$

$$\sum_{i=0}^{n_o} x_{ji}^{rk} - \sum_{i=0}^{n_o} x_{ij}^{rk} = 0, \qquad i \neq j,\ \forall j \in C,\ \forall r \in R, \forall k \in D, \tag{4}$$

$$\sum_{j=0}^{n_o} x_{ij}^{rk} \leqslant 1 \qquad \forall i \in W,\ \forall k \in D,\ \forall r \in R, \tag{5}$$

$$\sum_{i=0}^{n_o} \sum_{j=0}^{n_o} x_{ij}^{rk} \leqslant |S| - 1 \qquad \forall S \subseteq C,\ \forall k \in D,\ \forall r \in R, \tag{6}$$

$$\sum_{k=0}^{n_d} \sum_{i=0}^{n_o+n_w} \sum_{j=0}^{n_o} y_{pjk}^{r} x_{ij}^{rk} \leqslant N_{pw} \qquad \forall w \in W,\ \forall p \in P,\ \forall r \in R, \tag{7}$$

$$\sum_{i=0}^{n_o+n_w} \sum_{j=0}^{n_o+n_w} \sum_{r=0}^{n_r} d_{ij} x_{ij}^{rk} \leqslant L \qquad \forall k \in D, \tag{8}$$

$$\sum_{k=0}^{n_d}\sum_{r=0}^{n_r} y_{pjk}^r = dm_{pj} \qquad \forall j \in C,\ \forall p \in P. \tag{9}$$

The objective function (1) is to minimize the total number of distances traveled by drones. It is important to note that we use the assumption that by minimizing the objective function (1) we get the maximum of the objective function from the competition. Constraint (2) guarantee that the current loading of the drone must be less than or equal to the payload capacity of the drone; (3) condition guarantee, that every client is visited by at least one drone. Constraint (4) saying that the number of arrivals in vertices equal to the number of departures. Constraint (5) states that the drone can depart from the warehouse at most once during the trip. Constraint (6) eliminate subtours. Constraint (7) ensure that the total amount of product taken from the warehouse by drones should not exceed the amount of products in the warehouse. Constraint (8) require the fact that the drone trip length is less than the maximum flight length. Constraint (9) state that the quantity of a product of type p placed at the customer by all drones must be equal to the demand for this product type.

4. Solution methodology for the UAVRP. This section will discuss the mixed dimensional parallel genetic algorithm in detail. The Github repository with the MDPGA implementation is attached in the list of references [15].

4.1. Classical genetic algorithm. The first stage of solving the problem will consider the application of a classical genetic algorithm to solving abstract optimization problems. Let S be the set of all possible permissible encodings. In the classical genetic algorithm, all $c \in S$ must have the same constant length Ln.

A real fitness-function f defined on S is a function of solution assessment. We are searching for its point of maximum or minimum. The convergence of the genetic algorithm is based not on the analysis of the fitness-function, but on working with some set of solutions, which is called a population. Elements of the population are called individuals.

Before the start of the algorithm we generate the initial population $P^0 \subset S$, which consist of m encodings $c_1^0, ..., c_m^0$. A genetic algorithm consists of iterative implementations of operations (i is the number of the generation). A detailed description of the genetic algorithm is given in Algorithm 1.

Algorithm 1. Genetic algorithm:

```
1. Assessment: calculate $f(c)$ for all $c \in P^i$.
2. Selection: choose a reproductive set $R^i \subset P^i$ depending the on previous assessment.
3. Reproduction:
    (a) Crossover: choose pairs from the reproductive set $R^i$ and make new individuals from them with some special algorithm. The resulting descendants form the set $N^i$;
    (b) Mutation: with some probability a number of individuals $c_k \in P^i \cup N^i$ can mutate. This means they get some independence from fitness-function changes in their encoding: $c_k \longrightarrow c'_k$ (they must still be permissible $\Leftrightarrow c'_k \in S$).
4. Updating the population: create a new population $P^{i+1}$ from $m$ individuals(usually the best) $c \in P^i \cup N^i$.
```

4.2. Parallel genetic algorithm. To reduce the influence of the initial population on the convergence of the genetic algorithm, the algorithm is improved by parallelization. Its essence lies in the fact that we create k independent populations of a predetermined size $\{P_i^0\}_1^k = (P_1^0, P_2^0, ..., P_k^0)$. Each population passes the algorithm independently, thus we iteratively obtain the sets N_j^i, P_j^i, $j \in [1, k]$. In the end, the best individual H_j in each population is selected and the best one is again selected among the set $\{H_j\}_1^k$, ultimately giving the optimal solution H'. Each population can be calculated on a separate processor thread. Thus, the quality of the final solution increases significantly for a slight increase in the computation time.

4.3. Modified genetic algorithm. For a better understanding of the modifications, consider them using the example of the Traveling Salesman Problem (TSP) (Fig. 3, *I*), which is to find the most profitable route that passes through the specified cities at least once, and then returns to the original city.

One of the easiest and most convenient ways of encoding solutions to this problem is allocating strings c^h — n-permutation of numbers $1, ..., n$, where n is the number of sites. All manner of these c^h form the set S as in Fig. 3, *I*.

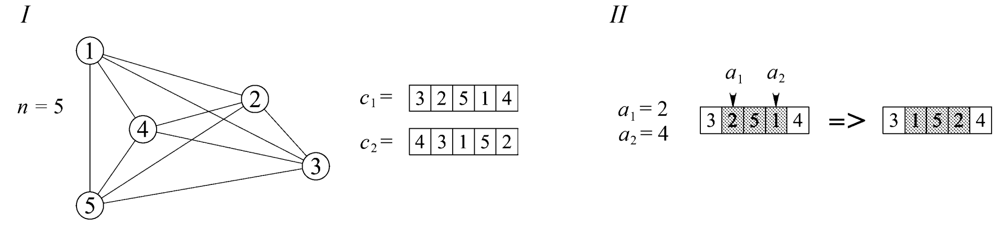

Fig 3. Modified GA using the example of the TSP
I — TSP graph and encodings example; *II* — two-points inverse mutation

Fitness-function is sum of the distances:

$$f(c^h) = \sum_{j=1}^{n-1} d(c_j^h, c_{j+1}^h) + d(c_n^h, c_1^h),$$

where $d(i, j)$ — distance between cites with numbers i and j.

Let m be the size of the population. The population is sorted by the value of function f from lower to higher. As a reproductive set R^i, we take first $m/2$ individuals.

The change in the algorithm consists in the absence of a crossover and obtaining convergence only due to mutation. As a mutation we use a two-point inverse: randomly choose two natural numbers $a_1, a_2 \in [1, n], a_1 < a_2$, part of code string between elements with numbers a_1 and a_2 reverse as in Fig. 3, *II*.

Additionally, we need to use another method of working with encodings for wrestling with local extrema. To do this, using swap mutation (i. e., choose two different elements to form the code string and swap them) to all individuals with some fixed probability.

4.4. Mixed dimension. At this point, we will proceed to the problem of delivering products using drones. The first step is to distribute drones to warehouses, uniformly in our case. So we have to bind drone $d_i \in D$ to warehouse $w_j \in W$. This means that drone d_i will deliver products and always come back to warehouse w_j. The second step is to get a sequence of orders (customers) for the drones at each trip.

Consider drone $d_i \in D$, bound to a warehouse $w_r \in W$. Let U_{w_r} be a set of orders that can be completely or partly delivered with products from w_r ($U_{w_r} = \{\, u \in O \mid P_u \cap P_{w_r} \neq \varnothing \,\}$). The set of all possible permissible encodings $S_{d_i}^t$ consists of sequences $c = (c_1, ..., c_h)$ of numbers of orders from U_{w_r}.

Code c must fulfill the condition that the sum of products has a weight lower or equal to the drone's carrying capacity:

$$\sum_{j=1}^{h} M_{p_j} \leqslant Q, \quad p_j \in P_{c_j} \cap \{P_{w_r} \setminus \cup_{g=1}^{j-1} p_g\}.$$

In this case, the encoding length need not be a fixed number because our modification of the algorithm has no crossover and there will be no collisions when trying to cross strings with a different size (Algorithm 2).

Algorithm 2. Our radial algorithm:

```
Begin
Load information from file
Make Drones(Dr), Warehouses (Wh), Products(Pr), Orders (Or)
for i in Drones do
    Dr[i] go to Wh[j] by distribution
end
for i in Wh do
    Let $r_0$ be a initial radius
    Determine $V_{w_0}$
    Make distance matrix
end

for t in Turns do
    for j in Dr do
        if $V_{w_i}$ =? then
            $r_i + 1 = r_i + st$
            if $r_i + 1 > R$ then
                Find the busiest Wh
                Dr[j] go to Wh
                Exit
            end
            else
                Determine $V_{w_i} + 1$
                Make distance matrix
                Exit
            end
        end
        else
            Start MDGA
            Dr[j] make flight on way B
            Make a command c for this move
        end
    end

end
Write Commands c in file
End
```

4.5. Using MDPGA for drones delivery. The goal of our optimization task is to complete all orders in the shortest possible time. Thus our fitness-function attempts to minimize the flight range and to maximize the count of completed orders:

$$f(c) = \frac{d(w, c_1^h) + \sum_{j=2}^{size(c)-1} d(c_j^h, c_{j+1}^h) + d(c_{size(c)}^h, w)}{h_{d_i} + 1},$$

here h_{d_i} — the order count completed over this simulated flight.

A detailed description of the mixed dimensional genetic algorithm is given in the Algorithm 3.

```
**Algorithm 3.** MDGA:
Attach all drones $d_j \in D$ to warehouses $w_r \in W$.
Implement path $c'$ for drone $d_j$, $j = \overline{1, U}$.
At the beginning of each route, we propose that $d_j$ has the carrying capacity $l_{d_j} = 0$.
Generate initial population $P^0$ with size $m$:
    1. Create empty code string $c$.
    2. Randomly choose product $p \in P_{U_{w_r}}$, and if $l_{d_j} + M_p \leqslant Q$, then the order number
       for that product adds up to $c$, and this step repeats.
The following steps implement iteratively ($i$ — number of iteration):
    1. Sort the population $P^i$ by function $f$ from lower to higher.
    2. Transform the first $m/2$ individuals with a two-point reverse mutation and write
       results replacing the last $m/2$ individuals.
    3. Apply swap mutation to $c \in P^{i+1}$ with some probability.
```

4.6. Radial optimization. The dimension of the problem was rather large for the genetic algorithm. Therefore, to increase the score of the algorithms a radial method was used to decrease this dimension. The pseudo-code radial method is described in the Algorithm 2.

Let U_{w_i} be a set of orders that can be completely or partly delivered with products from w_i. Under construction around a warehouse, is a circle Z of radius r. This circle contains orders $o_j \in U_{w_i}$, the distance to which is $d(w_i, o_j) \leqslant r$. We will call them V_{w_i}.

We get a new set $V_{w_i} \subset U_{w_i}$. The main feature of this set is that $\forall v \in V_{w_i}$, the execution time will be shorter than $\forall u \in U_{w_i} \setminus V_{w_i}$. We will call V_{w_i} the visibility of warehouse w_i. This visibility will also be the visibility for all drones attached to w_i.

Let $V_{w_i}^0$ be the visibility in the initial circle Z^0 of a radius r^0. When all orders from $V_{w_i}^0$ are complete, change r^0 on the step value δr. Thus, we get a new radius $r^1 = r^0 + \delta r$ and a new visibility $V_{w_i}^1$. Continue the iterative process while $V_{w_i}^k \neq U_{w_i}$ or no orders completed. Available drones fly to another warehouse.

Figure 4, *a–c* shows an example of a drone flying at different times in a radial version.

5. Simulation results. In this section we will present the results of the efficiency of MPDGA in practice according to the problem considered.

5.1. Competitions results. According to the results of the competition, the proposed algorithm scored 102 915 points. However, after improvements, it was possible to increase this value to 103 649 points. The score solutions of the winner amounted to 114 399 points. As the difference between the two is less than 10 %, this indicates that there are good prospects for examining the solution proposed in this article.

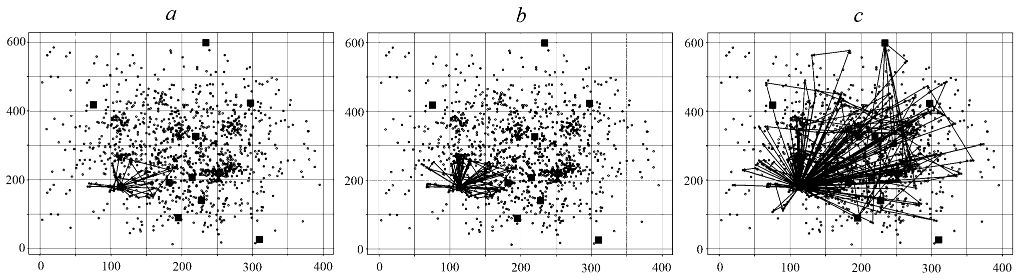

Fig. 4. Radial example

5.2. Solution steps analytic. To demonstrate the result of the algorithm, will conduct a series of comparisons of different versions of the solution. The global version is that everyone knows about all orders. The dimension of the problem is enormous. The radial version is that each input only knows about a part of the orders in the scope. The wandering version is that each drone moves randomly across the field.

5.3. Attachment profit. After a series of experiments, it was found that assigning drones to warehouses improves the results of the algorithm compared to the principle when, after finishing the trip, the drone flies to the nearest warehouse. This may be due to the similar distribution of products in warehouses and that redistribution is redundant in order to solve this problem.

5.4. Radial and global comparison. Figure 5, *a–c* show an example of one drone flying at different times in the global version. As we wrote in the previous section, we used an artificial reduction in the dimension of the problem by setting the radius and taking into account only those orders that do not go beyond it. The positive impact of this solution as compared to the global one, where all kinds of orders are taken into account, can be seen from Table 1.

Table 1. **Global version test**

Individuals	Average	Min	Max	Error	Error, %
20	89998.4	89657	90498	841	0.9344610571
250	98653.4	98556	98768	212	0.2148937594
2500	102051.2	101947	102242	295	0.2890705842

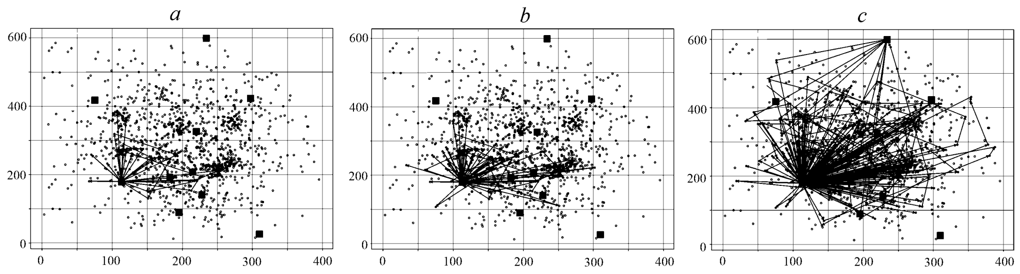

Fig. 5. Global example

5.5. MDPGA and greedy algorithm comparison. One of the most obvious solutions to this problem is a greedy algorithm. Its simplest modification is when the drone finds the order closest to the warehouse and flies to it until it is completed. This approach also gives an acceptable solution, though far from the best. The MDPGA improves the result of optimization of the target function by up to 30 % compared to the greedy algorithm as suggested in the competition.

5.6. Hyper-parameter dependencies results. Let us consider the dependence of results on hyperparameters.

Hyperparameters:

1. Initial visibility radius $r_0 \in [0, \sqrt{a^2+b^2}]$.
2. Step of increasing the radius $st \in [0, \sqrt{a^2+b^2}]$.
3. Depth of the genetic algorithm is the size of the initial population $n \in [0, \infty]$.

27 tests were conducted without using a parallel genetic algorithm in order to show the maximum, minimum, and average values of the objective function. Testing was carried out on a specific example of input data from the competition. Each test was repeated 5 times to assess the rate of error. Results are shown in Table 2.

According to the results of experiments, the initial radius r_0 was chosen to be 100. This value was used in subsequent tests:

$$\begin{gathered} 101\,236.13_{r_0=100}, \quad 101\,045.66_{r_0=150}, \quad 100\,567.40_{r_0=50} \ \text{(250 individuals)}, \\ 97\,613.66_{r_0=100}, \quad 97\,131.33_{r_0=150}, \quad 95\,717.46_{r_0=50} \ \text{(20 individuals)}, \\ 103\,125_{r_0=100}, \quad 103\,062.44_{r_0=50}, \quad 102\,779.55_{r_0=150} \ \text{(2500 individuals)}. \end{gathered}$$

This suggests that too much cutting of the initial scope leads to non-optimal solutions because of limitations. At the same time, a radius that is too large, while solving the problem of boundedness, gives too huge dimension to the problem.

There are similar comparisons for the mean for population size: $2500 > 250 > 20$ individuals of the population. This is completely predictable since GA is the most important component of success in our task. With 20 individuals, the error can reach 0.58 %, and with 2500 individuals — only 0.29 % with a much lower average error (0.31 vs 0.157 %).

In the end, Fig. 6 shows a graph of the convergence of the best results after selecting the parameters for different approaches both global and radial:

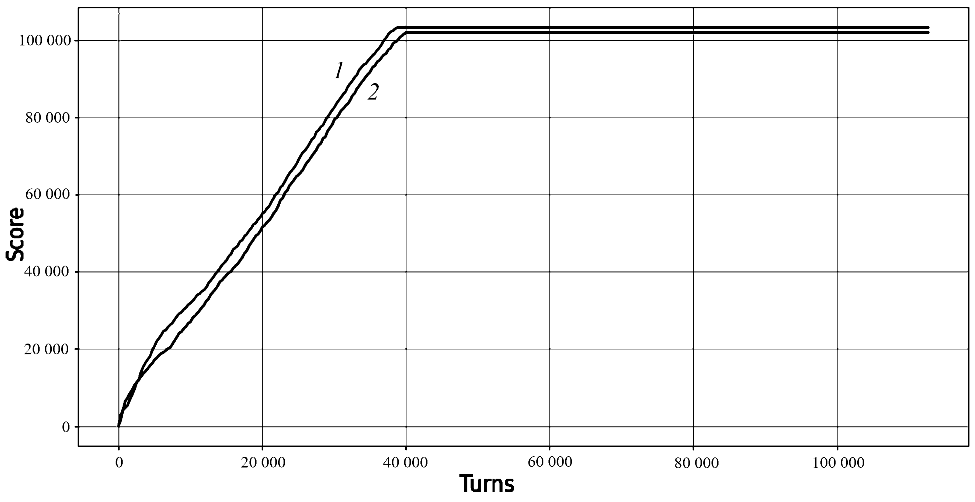

Fig. 6. Radial (*1*) and global (*2*) convergence

1) r_0 is 7.8, 15.6 and 23.4 % like our 50, 100, 150;
2) st is 7.8, 15.6 and 23.4 % like our 50, 100, 150;
3) n is 20, 250, 2500 the more the better, but the time is limited to 9 hours.

Table 2. **The results of experiments to determine hyperparameters**

r_0 – δr – ind	Average	Min	Max	Error	Error, %
50 – 50 – 20	95 692.2	95 551	95 786	235	0.2455790545
50 – 100 – 20	95 680	95 551	95 747	196	0.2048494983
50 – 150 – 20	95 780.2	95 542	96 100	558	0.5825838743
100 – 50 – 20	97 623.4	97 450	97 995	545	0.5582677924
100 – 100 – 20	97 580.8	97 437	97 692	255	0.2613218994
100 – 150 – 20	97 636.8	97 467	97 812	345	0.3533503761
150 – 50 – 20	97 023.2	96 890	97 117	227	0.23396466
150 – 100 – 20	97 197.6	97 133	97 302	169	0.1738726059
150 – 150 – 20	97 173.2	97 068	97 232	164	0.1687708134
50 – 50 – 250	100 481.6	100 415	100 621	206	0.205012659
50 – 100 – 250	100 574.6	100 403	100 717	314	0.312206064
50 – 150 – 250	100 646	100 511	100 714	203	0.2016970371
100 – 50 – 250	101 223.2	101 143	101 326	183	0.1807885939
100 – 100 – 250	101 239.8	101 100	101 465	365	0.3605301472
100 – 150 – 250	101 245.4	100 957	101 449	492	0.4859480036
150 – 50 – 250	101 007.8	100 922	101 139	217	0.2148348939
150 – 100 – 250	101 106	100 976	101 307	331	0.3273791862
150 – 150 – 250	101 023.2	100 931	101 147	216	0.2138122728
50 – 50 – 2500	103 050	102 957	103 129	172	0.1669092673
50 – 100 – 2500	103 043.3	102 968	103 088	120	0.11645586
50 – 150 – 2500	103 094	102 915	103 217	302	0.2929365433
100 – 50 – 2500	103 142.6	103 076	103 203	127	0.1231304213
100 – 100 – 2500	103 126.3	103 046	103 195	149	0.1444829804
100 – 150 – 2500	103 106	103 002	103 175	173	0.1677884895
150 – 50 – 2500	102 777.6	102 686	102 843	157	0.1527569219
150 – 100 – 2500	102 785	102 680	102 854	174	0.1692854016
150 – 150 – 2500	102 776	102 739	102 826	87	0.08465011287

6. Conclusion. This article discusses the problem of organizing a drone-based delivery service as a new version of the vehicle routing problem. The statement of the UAVRP includes the need to route a homogeneous fleet of drones with a limited capacity in order to deliver multiple products to customers from multiple heterogeneous warehouses. The dimension of the test case approximates tasks in real production conditions. As a solution method, an adaptation of the parallel genetic algorithm with a modification was proposed that consists in the absence of crossover and obtaining convergence only by counting mutations. The specifics of changing the dimension of individuals was added to find the optimal drone loading plans. In the course of further research, we would like to consider other options for settling the UAVRP problem. We also intend to develop other metaheuristics that can give a sufficient result for the algorithm in a faster time. A radial approach was also applied that allows us to artificially reduce the dimension of the problem without losing the quality of the solution. The proposed algorithm is interesting for further study and modification. Its effectiveness is confirmed by the results of the competition where there was less than a 10 % gap in results between the first place entry and our solution. In comparison to the greedy algorithm, the target function saw an improvement of up to 30 %.

References

1. Poikonen S., Wang X., Golden B. The vehicle routing problem with drones: Extended models and connections. *Networks*, 2017, no. 70(1), pp. 34–43. https://doi.org/10.1002/net.21746

2. Theverge.com. *Theverge Official Website, Drones could make Amazon's dream of free delivery profitable — The Verge.* 2015. Available at: https://www.theverge.com/2015/6/3/8719659/amazon-prime-air-drone-delivery-profit-free-shipping-small-items (accessed: June 20, 2021).

3. Radzki G., Thebbotuwawa A., Bocewick G. Uavs flight routes optimization in changing weather conditions — constraint programming approach. *Applied Computer Science*, 2019, vol. 15, no. 3, pp. 5–20. https://doi.org/10.23743/acs-2019-17

4. Kaggle.com. *Kaggle Official Website, Hash Code Archive — Drone Delivery Homepage*, 2021. Available at: https://www.kaggle.com/c/hashcode-drone-delivery (accessed: June 20, 2021).

5. Euchi J. Genetic scatter search algorithm to solve the one-commodity pickup and delivery vehicle routing problem. *J. of Modelling in Management*, 2017, vol. 12(1), pp. 2–18. https://doi.org/10.1108/JM2-10-2015-0077

6. Euchi J., Sadok A. Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication*, 2021, vol. 44. https://doi.org/10.1016/j.phycom.2020.101236

7. Erdogan S., Miller-Hooks E. A Green Vehicle Routing Problem. *Transportation Research. Pt E. Logistics and Transportation Review*, 2012, vol. 48(1), pp. 100–114. https://doi.org/10.1016/j.tre.2011.08.001

8. Fleischmann B. *The vehicle routing problem with multiple use of vehicles.* Technical report. Hamburg, Universitat Press, 1990.

9. Azi N., Gendreau M., Potvin J.-V. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers and Operations Research*, 2014, vol. 41, pp. 167–173. https://doi.org/10.1016/j.cor.2013.08.016

10. Cattaruzza D., Absi N., Feillet D., Vidal T. A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research*, 2014, vol. 236(3), pp. 833–848. https://doi.org/10.1016/j.ejor.2013.06.012

11. Cheikh M., Jarbou B. A variable neighborhood search algorithm for the vehicle routing problem with multiple trips. *Electronic Notes in Discrete Mathematics*, 2015, vol. 47, pp. 277–284. https://doi.org/10.1016/j.endm.2014.11.036

12. Ovsyannikov D. A., Mizintseva M. A., Balabanov M. Yu., Durkin A. P., Edamenko N. S., Kotina E. D. Optimization of dynamics of trajectory bundles using smooth and nonsmooth functionals. Pt 1. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2020, vol. 16, iss. 1, pp. 73–84. https://doi.org/10.21638/11701/spbu10.2020.107 (In Russian)

13. Popkov A. S. Optimal program control in the class of quadratic splines for linear systems. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2020, vol. 16, iss. 4, pp. 462–470. https://doi.org/10.21638/11701/spbu10.2020.411

14. Drivotin O. I. On numerical solution of the optimal control problem based on a method using the second variation of a trajectory. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2019, vol. 15, iss. 2, pp. 283–295. https://doi.org/10.21638/11701/spbu10.2019.211

15. Github.com. *Github Official Website*. Implementation of the MDPGA. Available at: https://github.com/LaLa-Lisa/Drones_delivery_compet (accessed: June 20, 2021).

Received: December 22, 2021.
Accepted: February 01, 2022.

Authors' information:

Anastasia Y. Markelova — Student; anastasiya_markel@mail.ru

Alexander L. Allahverdyan — Student; aleka_alexander@mail.ru

Alexey A. Martemyanov — Student; lyohich999@yandex.ru

Inga S. Sokolova — Student; Sokolova.ing@gmail.com

Ovanes L. Petrosian — PhD in Physics and Mathematics, Associate Professor; petrosian.ovanes@yandex.ru

Mikhail V. Svirkin — PhD in Physics and Mathematics, Associate Professor; msvirkin@spbu.ru

Прикладная задача маршрутизации для парка беспилотных летательных аппаратов с использованием модифицированного параллельного генетического алгоритма*

А. Маркелова, А. Аллахвердян, А. Мартемьянов, И. Соколова, О. Петросян, М. Свиркин

Санкт-Петербургский государственный университет, Российская Федерация,
199034, Санкт-Петербург, Университетская наб., 7–9

Для цитирования: *Markelova A., Allahverdyan A., Martemyanov A., Sokolova I., Petrosian O., Svirkin M.* Applied routing problem for a fleet of delivery drones using a modified parallel genetic algorithm // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2022. Т. 18. Вып. 1. С. 135–148.
https://doi.org/10.21638/11701/spbu10.2022.111

Все больше экспертов сходятся во мнении, что в ближайшем будущем бо́льшая часть грузовых перевозок будет осуществляться с использованием автоматизированных систем, и из них наиболее перспективной считается доставка с помощью дронов. Такая доставка выиграла бы благодаря независимости от ограничений транспортной инфраструктуры и дорожных условий и обеспечила бы более быструю развозку грузов, а также значительное снижение вредного воздействия на окружающую среду. Технические возможности беспилотных летательных аппаратов улучшаются, поэтому задача их координации и эффективного планирования маршрутов актуальна и пользуется большим спросом. Развитие таких технологий поможет снизить транспортные расходы и улучшить обслуживание клиентов за счет более быстрой доставки. В статье рассматривается прикладная задача маршрутизации для парка беспилотных летательных аппаратов с ограниченной грузоподъемностью для доставки разнородных товаров с возможностью загрузки на нескольких складах. Решение включает в себя новый подход, основанный на смешанном размерном параллельном генетическом алгоритме для поиска рациональных маршрутов доставки товаров различным клиентам, и задачу назначения для уменьшения размера в зависимости от количества складов.

Ключевые слова: доставка дронами, теория расписания, генетический алгоритм, задача маршрутизации транспорта, несколько депо, несколько продуктов, разделенная доставка.

Контактная информация:

Маркелова Анастасия Юрьевна — студент; anastasiya_markel@mail.ru

Аллахвердян Александр Львович — студент; aleka_alexander@mail.ru

Мартемьянов Алексей Алексеевич — студент; lyohich999@yandex.ru

Соколова Инга Сергеевна — студент; Sokolova.ing@gmail.com

Петросян Ованес Леонович — канд. физ.-мат. наук, доц.; petrosian.ovanes@yandex.ru

Свиркин Михаил Владимирович — канд. физ.-мат. наук, доц.; msvirkin@spbu.ru

* Работа выполнена при содействии гранта Президента Российской Федерации для государственной поддержки молодых российских ученых — кандидатов наук (проект № МК-4674.2021.1.1).