mathematics

Article

Graph-Clustering Method for Construction of the Optimal Movement Trajectory under the Terrain Patrolling

Boris V. Rumiantsev [1], Rasul A. Kochkarov [2] and Azret A. Kochkarov [1,2,*]

[1] Federal Research Centre "Fundamentals of Biotechnology" of the Russian Academy of Sciences, Leninsky Prospect, 33, Build. 2, 119071 Moscow, Russian Federation
[2] Department of Data Analysis and Machine Learning, Faculty of Information Technology and Big Data Analysis, Financial University under the Government of the Russian Federation, Leningradsky Prospekt, 49/2, 125167 Moscow, Russian Federation
* Correspondence: akochkar@fbras.ru

Abstract: The method of the optimal movement trajectory construction in the terrain patrolling tasks is proposed. The method is based on the search of the Hamiltonian circuit on the graph of the terrain map and allows automatic construction of the optimal closed path for arbitrary terrain map. The distinguishing feature of the method is the use of the modified algorithm for the Hamiltonian circuit search. The algorithm can be scaled for the maps corresponding to the graphs with a large (more than 100) number of the vertices, for which the standard brute-force algorithm of the Hamiltonian circuit search requires significantly higher execution time than the proposed algorithm. It is demonstrated that the utilized algorithm possesses 17 times less constant of the time complexity growth than the standard brute-force algorithm. It allows more than one order of magnitude (from 30 to 500 vertices, i.e., approximately to the 17 times) increase of the graph vertices that is used for the Hamiltonian circuit search in the real time (0.1–100 s) regime.

Keywords: graph theory; Hamiltonian circuit; time complexity; monitoring; patrolling

MSC: 05C90; 05C38; 05C45

Citation: Rumiantsev, B.V.; Kochkarov, R.A.; Kochkarov, A.A. Graph-Clustering Method for Construction of the Optimal Movement Trajectory under the Terrain Patrolling. *Mathematics* **2023**, *11*, 223. https://doi.org/10.3390/math11010223

Academic Editors: Emeritus Mario Gionfriddo and Andrea Scozzari

Received: 2 November 2022
Revised: 12 December 2022
Accepted: 30 December 2022
Published: 2 January 2023

1. Introduction

The task of the search for the optimal patrolling path at the present time has special relevance due to implementation of the automated moving devices, which require no presence of the operator and can be used as mobile platforms for a variety of sensor classes: optical, acoustical, chemical, etc. These devices allow solving of the wide range of tasks related to the automation of the spatial monitoring processes [1–4]. The type of sensors used with these devices is determined by the nature of the tasks that should be solved under the process of terrain patrolling. Therefore, in particular, the patrolling can be carried out with the aim of terrain mapping [5], search for minerals [6], echolocation [7], and also for local registration of the weather conditions [8]. Since effective terrain patrolling requires the construction of the optimal movement trajectory on the given map, the task of the effective patrolling almost inevitably requires the use of the graph theory [9]. The graph-based approaches for the patrolling problem solution are actively used in the modern works. The frequently considered problem is a patrolling with the use of multiple sensors [10]. Therefore, in the work in [11], the problem of efficient patrolling with use of multiple robots in a known environment is solved by assigning individual patrolling regions to each mobile agent. The work in [12] addresses the same problem and demonstrates the method of dynamical graph partitioning by moving ant-like agents using simple local interactions during the patrolling procedure. In another work [13], authors discuss the approach of complex automation of territory patrolling by use of a set of mobile robots and propose the graph-based algorithm for the partitioning of the map between the robots. The article [14] is dedicated to the stochastic graph-based strategy for the terrain patrolling with a set of agents, that can outperform the deterministic approaches.

Another approach for the graph-based solution of the patrolling problem in the frame of Green Security Games is discussed in [15]. In this work, authors propose to construct the patrolling path by the contraction of the complete terrain graph based on the importance of each individual node for the patrolling procedure, which allows reduction of the time complexity in the context of large-graph problem.

In the frame of this work, it is proposed to implement a patrolling procedure on the entire graph with the use of one sensor moving on the map with the given placement of obstacles, which is an alternative to the above-mentioned approaches based on multi-agent partitioning of the graph and its contraction. In this case, as it will be discussed below, the relevant method for the realization of effective terrain patrolling is to construct the graph of the terrain map that allows movement of the sensor over the Hamiltonian circuit [9], i.e., closed path passing through each graph vertex exactly once.

The task of the Hamiltonian circuit search at the present time is a topical mathematical problem, that possess applied nature in a variety of areas. In particular, it is the areas where it is required to realize search of the optimal movement path of an object. These areas include transport and logistics sphere [16,17], geological exploration directions [18], and as it was mentioned above, the area of the terrain patrolling [19,20]. Despite its relevance, the task of the Hamiltonian circuit search remains one of the most difficult for optimization mathematical challenges at the present time.

The problem of the Hamiltonian path search and, in particular, search of the Hamiltonian circuit, is the NP-complete problem and in the case of the brute-force search algorithm possesses exponential time complexity [9]. Under the presence of a priori information about graph that is used for the Hamiltonian path search, it is possible to decrease the time complexity of the problem to the subexponential [21] and even to the polynomial [22] level. Exponential complexity of the Hamiltonian path search on the arbitrary graph restricts the number of the graph vertices that allows solving of the problem in the real-time regime. This factor is of particular importance in applications where it is required to construct an optimal movement trajectory under fast change of the graph configuration.

The motivation of the present work is the development of the real-time (0.1–100 s) algorithm for construction of the optimal patrolling route in the form of the Hamiltonian circuit that ensures the full covering of the given terrain map. The novelty of the method is provided by the modified algorithm of the Hamiltonian circuit search that lies at the core of the proposed method and provides significant acceleration of the search procedure in comparison with the standard brute-force search algorithm while preserving the exponential time complexity. Further in the paper, the complete statement of the problem, the description of the developed theoretic-graph method for the construction of the optimal patrolling trajectory, and the results of the method effectiveness investigation are considered.

2. Problem Statement

In the frame of this work, the task of the terrain patrolling on the given two-dimensional map of the obstacles is considered (Figure 1).

The "patrolling" (or "monitoring") term in this work means the movement of one sensor over map that is accompanied by the registration of generalized targets. At the initial moment of patrolling targets are placed in the random points (x, y) of map with equal probability. During the patrolling procedure, targets randomly move across the map. The "obstacles" term means the areas of the map where targets cannot be placed and sensors cannot be located. In a real situation, this kind of obstacle could be the mountain chains, reservoirs, and elements of the urban terrain infrastructure, that can be insurmountable obstacles for both the ground- and air-based sensors.

In the context of the approach under consideration, the sensor is assigned by a field of view in which the sensor can register an event. The field of view is a circle of radius R with the center at the sensor location point (Figure 2).

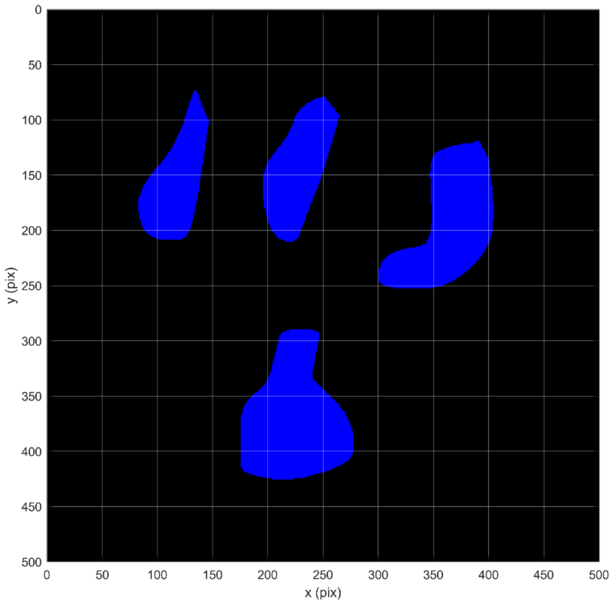


Figure 1. The example of the two-dimensional terrain map considered in this work. The dark blue color stands for the obstacles, i.e., areas where a sensor cannot be located and where there is no need for patrolling procedure. The black color stands for the area where it is required to realize patrolling procedure and a sensor can be located. The white grid that marks ticks with the step of 50 pixels is used for the visual reference and does not relate to the discussed problem.

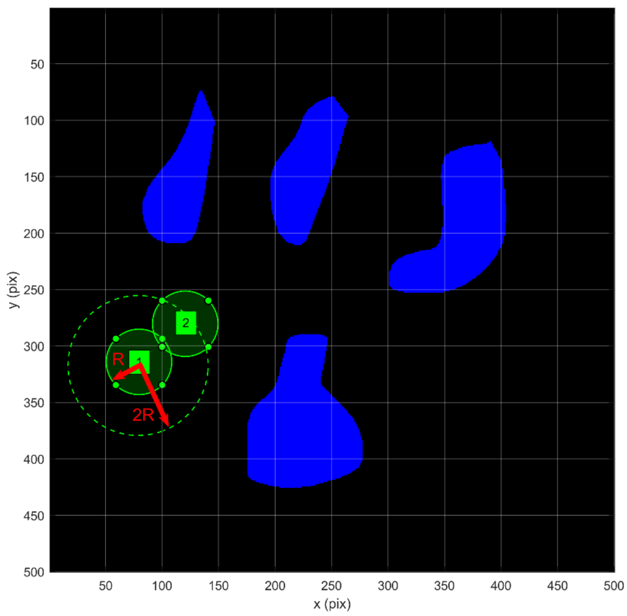


Figure 2. The sensor's fields of view (#1 and #2, green color). Being at the center of area #1, in one iteration of time the sensor can move to any point placed inside the circle of radius $2R$ (green dashed line) including the point that determines the center of the field of view #2.

In the discussed model, it is considered that under the process of monitoring, the sensor moves on the map in the discrete manner with the step $\Delta\vec{r}$ that is less or equal to the diameter $2R$ of its field of view ($1\ pix \leq \left|\Delta\vec{r}\right| \leq 2R$). This approximation corresponds to the real situation under the condition that time step Δt is much less than the complete monitoring time T ($\Delta t \ll T$). The use of this approximation allows treating of the sensor movement as a movement between centers of the fields of view placed in the monitoring area with some intersection (density) between each other. By covering of the monitoring area with fields of view, it is possible to formalize the sensor movement on the map as a movement between vertices of the graph that is constructed in correspondence with the fields of view placement.

It can be demonstrated that in order to minimize the targets detection time, the movement trajectory should represent a Hamiltonian circuit, which is discussed in previous paper concerning the considered problem [19]. The main idea is as follows.

In case of one sensor moving with the constant speed and nonmoving targets, the time of the targets registration Δt_{reg} is equal to the time Δt_{sweep} of complete sweeping (or coverage) of the monitoring area by the sensor that is inversely proportional to the sweeping rate of the sensor:

$$\Delta t_{sweep} = \frac{S_{total}}{dS_{swept}/dt} \tag{1}$$

where S_{total} is the total area of monitoring zone, dS_{swept}/dt is the sweeping rate that is area of map swept by the sensor in a unit of time. Under the given map area S_{total}, in order to minimize the sweeping time Δt_{sweep} and, consequently, the registration time Δt_{reg}, it is required to maximize the sweeping rate dS_{swept}/dt. In turn, under the given sensor movement speed, this maximization can be realized if every point of the map is covered exactly once during the period Δt_{sweep}. This is ensured by the movement of the sensor along the Hamiltonian path on the terrain graph.

In the case of moving targets, the single sweeping of the patrolling area can be insufficient for the registration of all targets and more sweeping cycles can be required. The cyclic sweeping of the patrolling zone, which ensures the coverage of every point of the map during one patrolling cycle exactly once and with equal coverage period, requires the return of the sensor to the initial point at every patrolling cycle. This can be ensured using the Hamiltonian cycle instead of the Hamiltonian path as a movement trajectory of the sensor. Thus, the optimal movement trajectory, which minimizes the detection time of moving targets in the discussed problem, is a Hamiltonian cycle.

Thus, the task of the effective terrain patrolling can be divided into two consecutive tasks. The first task is the realization of the effective covering of the target monitoring zone with the sensor fields of view and construction of the terrain graph in accordance with the obtained placement of the fields of view. The second task is the search for the Hamiltonian cycle on the constructed terrain graph that represents the optimal movement trajectory for the sensor. Further in the Section 3 "Method description", the algorithms for the solution of these two tasks are discussed.

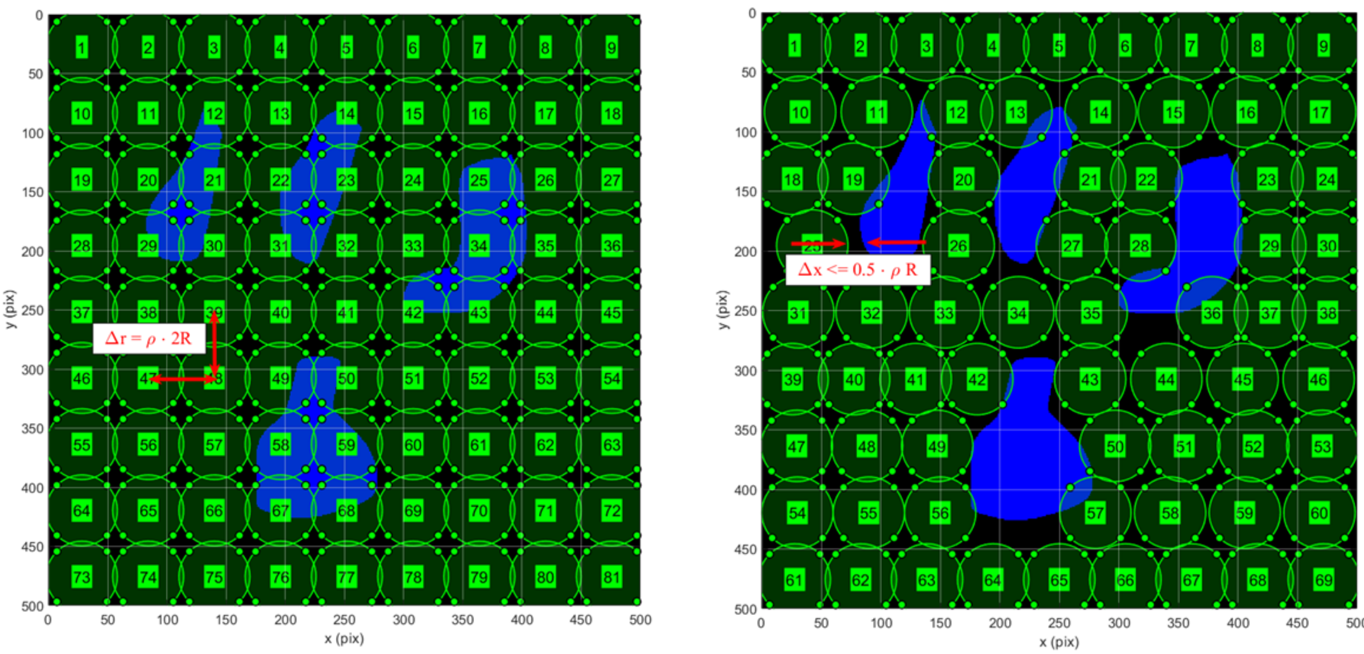
3. Method Description

3.1. The Algorithm of Target Zone Covering

In order to cover the monitoring area with the sensor fields of view, the following algorithm is used (Figure 3).

In order to ensure the effective monitoring of the terrain map that is characterized by the minimal crossing area between sensor fields of view and obstacles, the size of the field of view R should be less or equal to the minimal distance between the obstacles on the given map.

At the first step the grid with the uniform step Δr consisting of the fields of view is constructed on the map, where step Δr is set as $\Delta r = \rho \cdot 2R$, $\rho \in (0, 1]$ is the factor that determines the density of the fields of view placement. The term density is used because ρ determines the distance between the neighboring fields of view. Therefore, the smaller ρ, the more fields of view is placed on a given area, that indicates the increase of density of the fields of view placement.



(**a**) (**b**)

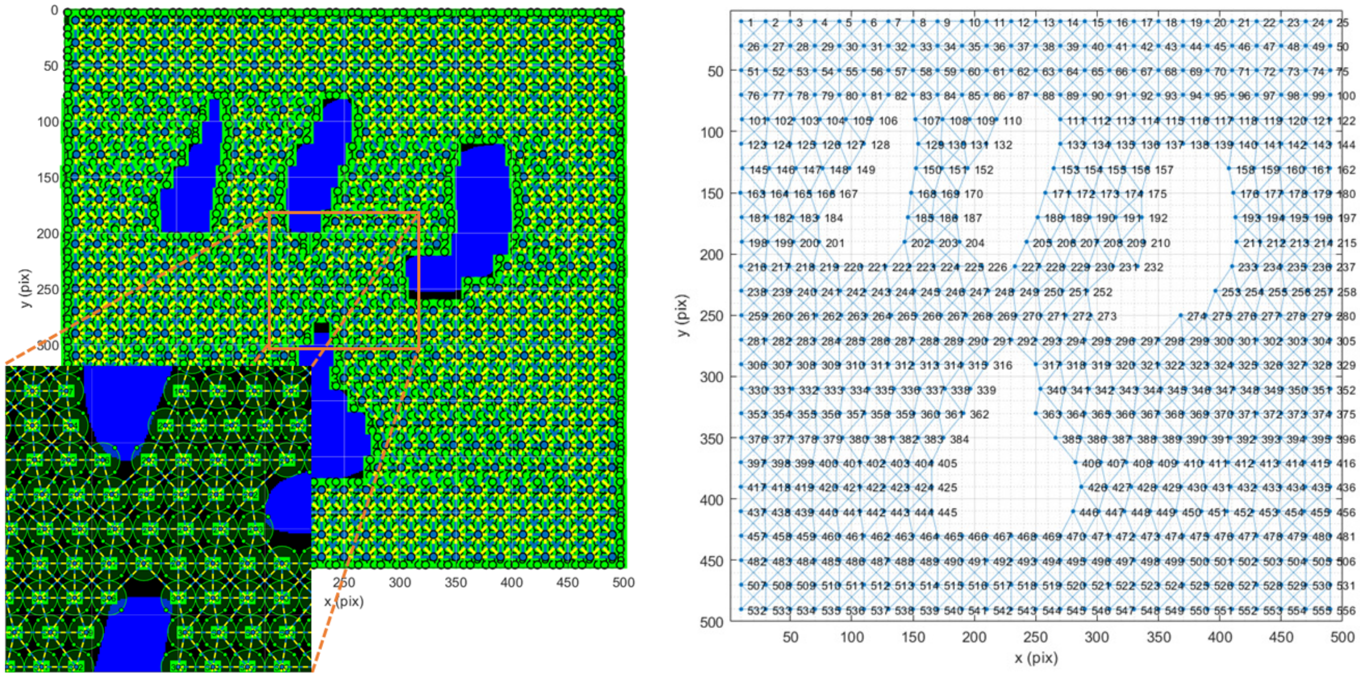
Figure 3. (**a**) Non-distorted grid of the sensor fields of view. $\Delta r = \rho \cdot 2R$ is the distance between the nodes of the grid; (**b**) The distorted grid of the sensor fields of view. The minimal size along x axis of the noncovered part of the map is $\Delta x \leq 0.5 \cdot \rho R$.

At the second step, the grid is distorted: the grid step is changed in such a way that the size of the noncovered map region along x axis is less or equal to $0.5 \cdot \rho R$ (Figure 3). The fields of view are placed in the nodes of the distorted grid lying outside the obstacles.

As a result of the conducted procedure, the grid consisting of the sensor fields of view is formed on the map. After passing through each of the grid nodes, the sensor will provide an almost complete scan of the target monitoring zone area. As will be discussed in Section 3.1, the fundamental limit of the fraction of covered area is 0.75 for the square map filled with the circular sensor fields of view without overlap. This means that the area of map patrolled by the sensor with the circular field of view will be covered almost completely, i.e., 75% of the map area will be scanned. This is the consequence for the application that cannot be overcome in the task of the patrolling with use of the sensor with circular shape of the field of view. The possible solution here is to overlap fields of view on the terrain map, which inevitably leads to the loss of the patrolling optimality, since in this case there will be "selected" points, which sensor monitors for more time, than the other ones, under one cycle of the terrain scan.

As mentioned above, the optimal movement trajectory, which minimizes the detection time of moving targets in the discussed problem, is a Hamiltonian cycle. In order to find the Hamiltonian circuit, the grid of the sensor fields of view is treated as a simple undirected graph $G_1(V_1, E_1)$ with the vertices located at the corresponding nodes of the grid, i.e., at the centers of the sensor fields of view (Figure 4).

The edges E_1 of the graph $G_1(V_1, E_1)$ connect those vertices V_1 that are located at the distance of the sensor step $\left|\Delta \vec{r}\right|$ from each other, $\left|\Delta \vec{r}\right| \leq 2R$. The terrain graph $G_1(V_1, E_1)$ does not contain self-loops (i, i) and parallel $(i, j)_k$ edges [9].



(**a**) (**b**)

Figure 4. (**a**) Sensor fields of view placement; (**b**) corresponding terrain graph G_1.

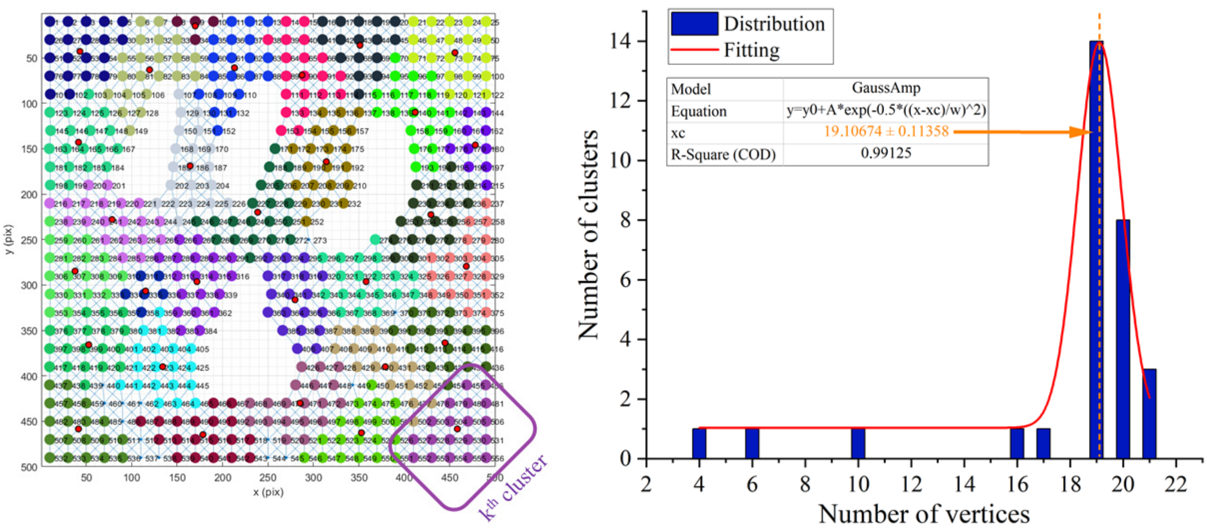
3.2. Algorithm of the Hamiltonian Circuit Search

The idea of the algorithm is as follows. In the frame of the stated patrolling problem, the number of the terrain graph vertices V_1 can be too high to implement the standard brute-force algorithm [23] for the Hamiltonian circuit search in real-time regime specified in this work as 0.1–100 seconds (3 orders of magnitude). In order to overcome the problem of the graph size, the following method is proposed. The terrain graph is decomposed into the so-called clusters, i.e., separate connected graphs $g_k(v_k, e_k)$, $k \in [1, n]$, $n = |V_1|/N$, $v_k \in V_1$ which contains low number of the vertices N ($N = 20 - 30$ in the frame of the conducted computational experiments), that allows application of the standard brute-force algorithm for the Hamiltonian path search in real-time regime in the frame of each cluster. The decomposition of the terrain graph $G_1(V_1, E_1)$ into clusters $g_k(v_k, e_k)$ allows consideration of the clusters set $\{g_k(v_k, e_k)\}$ as a single high-level graph $G_0(V_0, E_0)$, which can be treated as a terrain graph in the zero approximation, where clusters $g_k(v_k, e_k)$ acts as vertices V_0, and edges E_0 connect vertices V_0, which correspond to the neighboring clusters. In the zero approximation, it can be considered that the movement of the sensor under the patrolling procedure is realized over the graph $G_0(V_0, E_0)$, which ensures the Hamiltonian circuit search on its vertices in real-time regime due to low number N of the vertices V_0. At the next step, in order to clarify the route, the sensor movement in the frame of each cluster $g_k(v_k, e_k)$ is considered. The movement trajectory of the sensor in each cluster is set as Hamiltonian path $p_k(g_k)$ that allows movement of the sensor from initial vertex in k$^{\text{th}}$ cluster $g_k(v_k, e_k)$ to connected with the k$^{\text{th}}$ cluster vertex of the (k+1)$^{\text{th}}$ cluster $g_{k+1}(v_{k+1}, e_{k+1})$. The latter vertex becomes the initial vertex in (k+1)$^{\text{th}}$ cluster. The search of every Hamiltonian path $p_k(g_k)$ is realized with use of the standard brute-force algorithm, which due to the low ($N = 20 - 30$) number of the cluster vertices, can be implemented in real-time regime. By the connection of the Hamiltonian paths $p_k(g_k)$ in accordance with the Hamiltonian circuit over the graph $G_0(V_0, E_0)$, the total Hamiltonian circuit $C_1(G_1)$ over the graph $G_1(V_1, E_1)$ is constructed.

In the common case of arbitrary high number of the terrain graph vertices V_1, it is possible to accelerate the procedure of the Hamiltonian circuit search by the recursive multi-level repetition of the clustering process, which can be done with use of the fast

METIS algorithm [24], and search for the Hamiltonian paths in each cluster. In the frame of the present work, the case of a single-level clustering was realized, which provided the opportunity to increase the number of graph vertices in 17 times in comparison with the brute-force method while maintaining the real-time nature of the calculations.

The algorithm of the method is as follows (Figure 5). First, the clustering of the terrain graph vertices V_1 is performed. In the frame of clustering procedure at the first step, the ascending sorting of the graph vertices V_1 in accordance with vertices degree d is conducted. The formation of the first cluster $g_1(v_1, e_1)$ is performed starting from the vertex with the lowest degree d. The neighboring vertices with the degree $d \geq 2$ in relation to the cluster $g_1(v_1, e_1)$ are added to this cluster until either number of vertices in this cluster is equal to limiting number of vertices N or vertices with the degree $d \geq 2$ in relation to this cluster are absent. The formation of the next cluster $g_2(v_2, e_2)$ is conducted by starting with the non-clustered vertex with the lowest degree d among remaining ones. The repeating of this procedure provides the clustering of almost all vertices of the terrain graph $G_1(V_1, E_1)$. The clustering process is, in fact, the operation that is reversed in relation to the operation of the replacing a graph vertex with a seed [25]. The algorithm of clustering itself can be treated as a solution of the graph burning problem [26]. The number of the remaining non-clustered vertices is relatively low and in the frame of the conducted experiments was no more than 5% of the total number of the terrain graph vertices (Figure 5).

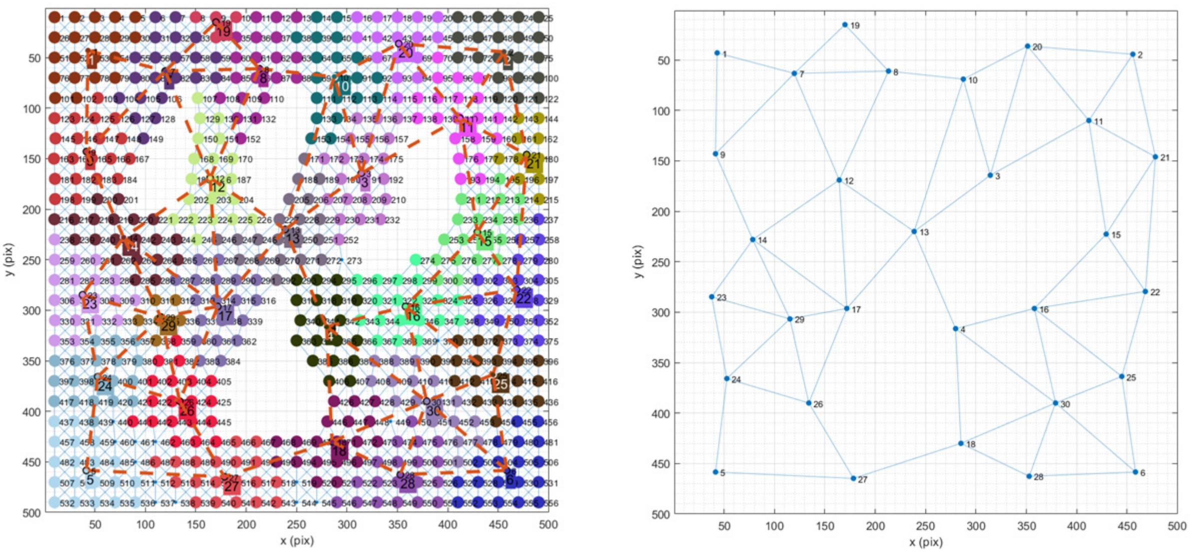


(**a**) (**b**)

Figure 5. (**a**) The clustering of the vertices V_1 of the terrain graph G_1. The colors stand for the clusters g_k, red points are the "centers of mass" of each cluster g_k. (**b**) The distribution of the clusters g_k by number of vertices $|v_k|$. This indicates that the average number of vertices in clusters g_k is close to the selected limiting number $N = 21$.

After the clustering procedure, the search of the Hamiltonian circuit $C_0(G_0)$ on the graph of clusters $G_0(V_0, E_0)$ is started (Figure 6).

If the Hamiltonian circuit $C_0(G_0)$ on the graph of clusters is absent, the density of the fields of view placement is increased by the decrease of the factor ρ. Then, the clustering starts again and the search of the Hamiltonian cycle $C_0(G_0)$ is performed on the updated graph $G_0(V_0, E_0)$. This procedure is repeated until the Hamiltonian cycle $C_0(G_0)$ is found. Discussing this in more detail, if the algorithm does not find Hamiltonian cycle at the fixed density coefficient ρ, the ρ is decreased by small step (0.05 in our realization), the terrain graph is updated in correspondence with the updated vertices placement, and the algorithm tries to find Hamiltonian cycle on the updated terrain graph. Since the increase of the density of the vertex placement under decrease of ρ adds new vertices to the terrain graph and, therefore, opens new possible routes for the sensor movement, then under certain small ρ, the possibility of Hamiltonian cycle construction on the terrain graph will be guaranteed. Hamiltonian cycle found under sufficiently small ρ, which results in significant overlap of sensor fields of view, will not be optimal, since in this case, there will be "selected" points in the overlapped regions, which sensor monitors more time under one cycle of the terrain scan than the other ones.



(**a**) (**b**)

Figure 6. (**a**) The clusters g_k connected by vertices V_0. (**b**) The corresponding graph of clusters G_0.

After the Hamiltonian cycle $C_0(G_0)$ is found, the search of the Hamiltonian paths $p_k(g_k)$ in each cluster $g_k(v_k, e_k)$ is performed. As it was mentioned above, the search is conducted in such a way that every Hamiltonian path $p_k(g_k)$ is connected with the Hamiltonian path $p_{k+1}(g_{k+1})$, where order k of the clusters g_k is fixed by the Hamiltonian cycle $C_0(G_0)$. The subsequent connection of the Hamiltonian paths $p_k(g_k)$ gives the total Hamiltonian cycle $C_1(G_1)$ over the full terrain graph $G_1(V_1, E_1)$ (Figure 7).

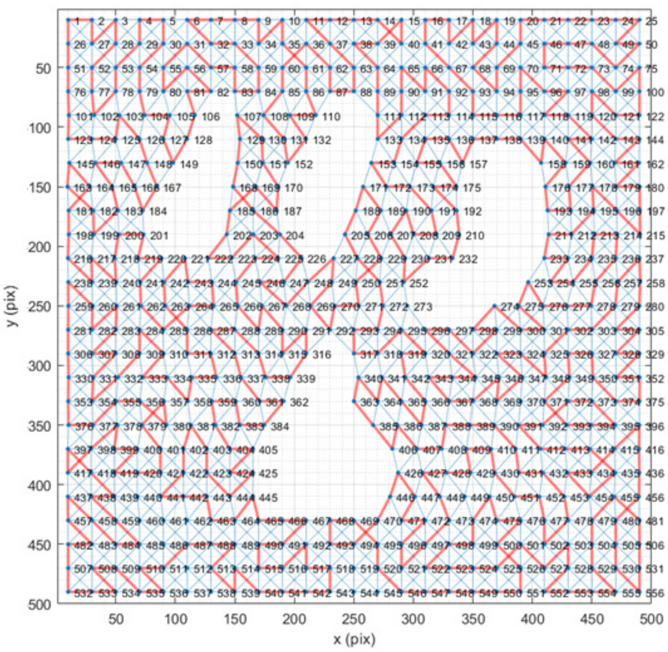


Figure 7. The constructed Hamiltonian cycle $C_1(G_1)$ over the terrain graph G_1.

Formulation of the total algorithm for terrain graph construction and search of the Hamiltonian circuit is presented in Algorithm 1 as MATLAB-like pseudo-code.

Algorithm 1 MATLAB-like pseudo-code of the total algorithm

```
hamCycle = empty; % initialization of total Hamiltonian cycle as empty constant.
N = 21; % Initialization of maximal number of nodes in cluster.
rhoCounter = 0;
while hamCycle == empty
  rho = 1 – rhoCounter*0.01; % density factor ρ
  %————————————————————————
  % Construction of the terrain graph G1(V1, E1)
        undistortedPlacement = fieldsUndistortedPlacement(Map, R rho);
        distortedPlacement = fieldsDistortedPlacement(Map, R, rho, undistortedPlacement);
        terrainGraph = createTerrainGraph(distortedPlacement, R, rho);
        numGk = round(numNodes(terrainGraph)/N); % Calculation of number of clusters
        % ————————————————————————-
        % Execution of clustering procedure:
        for i = 1:1: numGk
        Gk(i) = "node with maximal degree d among non-clustered nodes";
        n = 0;
          while n < N or "no new nodes with d ≥ 2 for current cluster Gk(i)"
              Gk(i) = [Gk(i) "node with d ≥ 2 for Gk(i)"];
          end
        end
        %————————————————————————————————
        % Construction of the graph of clusters G0(V0, E0)
        clustersGraph = createClustersGraph(Gk, R, rho);
        %————————————————————————————————
        % % Search of Hamiltonian cycle C0(G0) over graph of clusters G0(V0, E0)
        hamCycleClusters = hamiltonianCycle(clustersGraph);
        %————————————————————————————————
        % Search of Hamiltonian paths pk(gk) inside every cluster-subgraph gk(vk, ek)
        if hamCycleClusters != empty
          for i = 1:1:length(Gk)
          hamPath(i) = hamiltonianPath(Gk(hamCycleClusters(i)), Gk(hamCycleClusters(i + 1)));
            if hamPath(i) == empty
                break;
            end
          end
          %——————————————————————————————
% Connecting of Hamiltonian paths pk(gk) to the total Hamiltonian cycle C1(G1) over the terrain
% graph G1(V1, E1)
          hamCycle = connect(hamPath);
        end
end
```

4. Results and Discussion

In order to characterize the efficiency of the discussed approach for the search of optimal patrolling route, it is proposed, first of all, to characterize the efficiency of the terrain map covering algorithm and, secondly, to compare the time complexity of the proposed algorithm for the Hamiltonian circuit search with the time complexity of the standard brute-force algorithm.

The realization of the brute-force method, used in the frame of the discussed work, is presented in [27]. This recursive algorithm checks at every iteration if the neighboring vertex can be added to the Hamiltonian circuit or it has already been added. Thus, the algorithm iterates through the path options and finally finds the Hamiltonian cycle in case it exists for the given graph.

The execution of discussed algorithms was performed in MATLAB R2021b (Mathworks, Natick, MA, USA) with the use of CPU 11th Gen Intel Core i7-11800H, 2.30 GHz.

4.1. Efficiency of the Terrain Map Covering Algorithm

For the assessment of the efficiency of the terrain map covering algorithm the ratio S_Σ/S_Ω can be used, where S_Σ is a total area of the monitoring zone covered by the sensor fields of view, S_Ω is a total area of the monitoring zone. As a reference value of this ratio there can be used the value that characterizes the covering of the square monitoring zone by the non-overlapping sensor fields of view (Figure 8). The value of ratio S_Σ/S_Ω in this case is equal to $\pi/4 \approx 0.75$ for every radius of the sensor fields of view inscribed in the monitoring area.

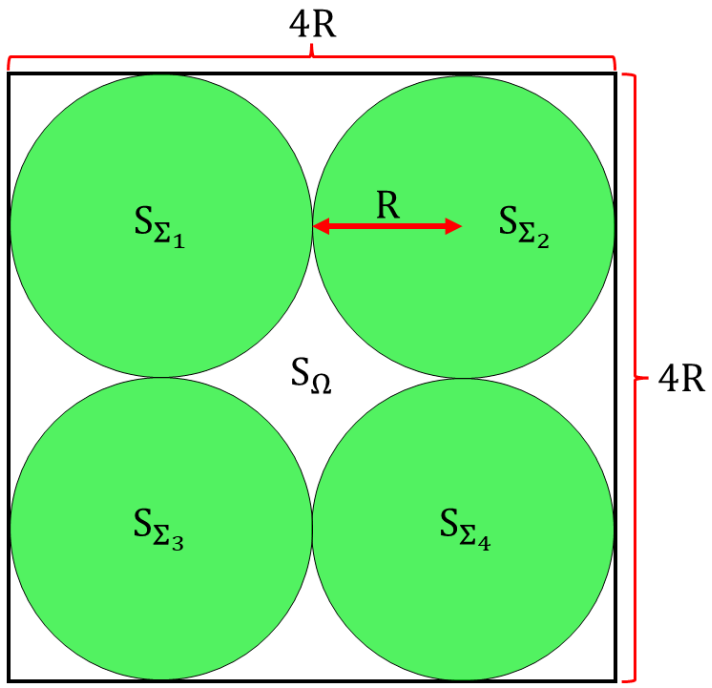


Figure 8. The scheme of the reference monitoring configuration. In the case of four fields of view ($n = 4$), which is depicted in the scheme, the effectiveness of the covering is $\frac{S_\Sigma}{S_\Omega} = \frac{4 \cdot \pi R^2}{4R \cdot 4R} = \frac{\pi}{4}$. In the case of arbitrary number of the fields of view inscribed in the monitoring area, the effectiveness is also equal to the $\pi/4$, since $\frac{S_\Sigma}{S_\Omega} = \frac{n \cdot \pi R^2}{(2R \cdot \sqrt{n})^2} = \frac{\pi}{4}$.

Based on this value, it is possible to characterize the coverage efficiency as follows: $S_\Sigma/S_\Omega < 0.75$ corresponds to the insufficient coverage of the target monitoring zone, $S_\Sigma/S_\Omega = 0.75$ corresponds to the maximum coverage of the target zone provided there is no overlap of the fields of view areas, $S_\Sigma/S_\Omega > 0.75$ corresponds to the increase of the coverage area of the target zone due to the overlapping of the fields of view. Thus, one has to choose between maximization of the covered area of target zone and minimization of the overlap area of the fields of view. The discussed algorithm allows reaching of values S_Σ/S_Ω at the level of 0.75–0.85, which indicates the relatively low (the ratio of overlap area to the area of target zone is $S_\cap/S_\Sigma \approx 0.01 - 0.03$ on the map in Figure 2) overlap of the fields of view, which provides increase of the covered area in comparison with the case of $S_\Sigma/S_\Omega = 0.75$. A possible improvement of this coverage algorithm for the increase of the coverage area while maintaining the overlap area of the fields of view is to realize distortion procedure of the covering grid relatively to both x and y axes.

4.2. Efficiency of the Algorithm for the Hamiltonian Circuit Search

In order to characterize the efficiency of the proposed algorithm for the Hamiltonian circuit search, the concept of time complexity is used [28]. The strict mathematical derivation of the computational complexity of the proposed algorithm seems to be hardly retrieved, since the performance depends on the map configuration. However, the statistical approach, which is realized by measurement of the algorithm execution time on the different maps, allows experimental determination of the complexity asymptotic law. It is the approach that has been used in the frame of the present work for the establishment of the time complexity growth law. The comparison of the experimentally determined time complexity for modified algorithm and standard brute-force search method is presented in Figure 9.

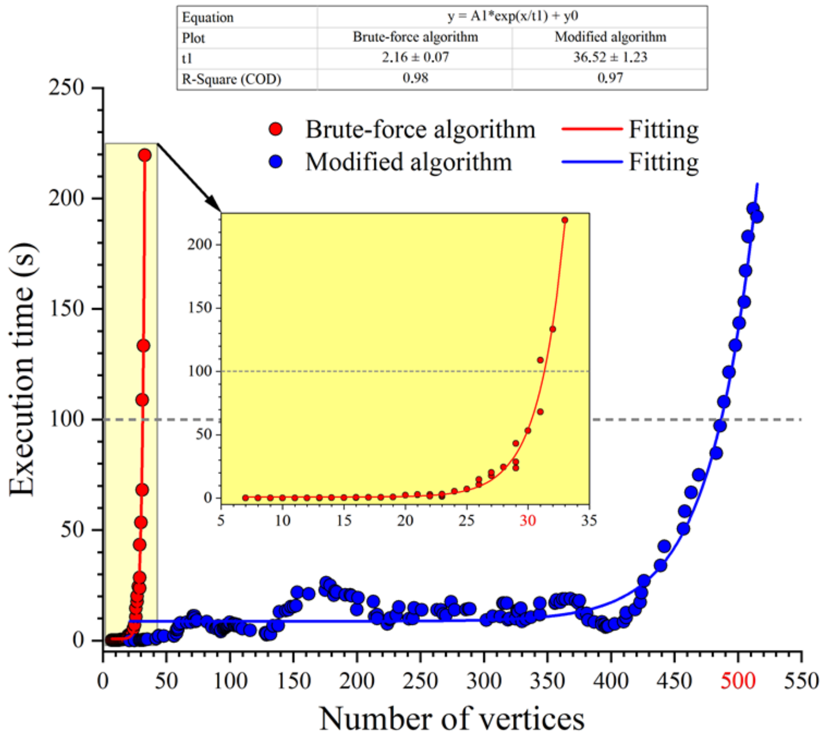


Figure 9. The comparison of the experimentally determined time complexity for modified algorithm and standard brute-force search method.

To obtain these dependencies, the algorithms were run for various maps with random location of obstacles and different sizes of visibility areas, that allowed achievement of a different number of terrain graph vertices.

As it is indicated in Figure 9, the standard brute-force algorithm is performed in a real-time regime (0.1–100 seconds) in the case of graphs with the number of vertices up to 30, while the proposed modified method allows enhancement of this range up to 500 vertices. It should be mentioned that values of 30 and 500 vertices determine the number of vertices corresponding to the increase of the execution time by three orders of magnitude for standard brute-force and modified algorithms respectively.

Both discussed algorithms possess the exponential law of the time complexity growth under the increase of the input data volume, that is represented by the number of terrain graph vertices V_1. However, the growth constant t_1 (Figure 9) of the time complexity curve for the modified algorithm is significantly lower (in $36.52/2.16 \approx 17$ times) than the one for the brute-force algorithm. Due to this fact, it is possible to increase by more than an order of magnitude (from 30 to 500 vertices, approximately in $500/30 \approx 17$ times) the number of the terrain graph vertices while keeping unchanged the real-time nature of the search procedure. Thus, in comparison with the standard brute-force algorithm, the use of the proposed modified algorithm makes it possible to significantly increase the number of graph vertices which are used for search of the Hamiltonian circuit in real-time (0.1–100 s) regime.

5. Conclusions

In the frame of the present work the method for the construction of the optimal path for arbitrary terrain map patrolling is developed. The method is based on the use of the modified algorithm for the search of Hamiltonian circuit on the terrain graph. It is demonstrated that the proposed method is effective both in the coverage of the monitoring zone, since it provides almost total coverage of the terrain area, and in the search of the Hamiltonian circuit over the terrain graph, since it ensures the increase of the search rate in comparison with the standard brute-force method. Moreover, in comparison with the brute-force method the proposed algorithm for the Hamiltonian circuit search allows more than an order of magnitude increase of the number of graph vertices (from 30 to 500 vertices, in $500/30 \approx 17$ times), which are used for the real-time (0.1–100 seconds) search of Hamiltonian circuit. The proposed approach of the vertices clustering that is used in the frame of the developed Hamiltonian circuit search procedure can be scaled to the graphs with the higher number of vertices by the recursive multi-level clustering of the terrain graph vertices.

The implementation of the proposed method in practical cases could be as follows. A user uploads the 2D terrain map to the computer program, which uses described algorithm for the patrolling path construction. The constructed path (i.e., the sequential set of 2D coordinates) is uploaded to the automated moving device which carries a sensor. Finally, moving device starts to patrol the terrain according to the uploaded path while the current position of device is monitored via, for example, GPS.

Thus, the discussed method can be applied both in the frame of the terrain patrolling problems and in the common area of tasks which require the use of the fast algorithm for the Hamiltonian circuit search on the large graphs.

Author Contributions: Formal analysis, A.A.K. and B.V.R.; investigation, B.V.R.; methodology, B.V.R.; project administration, R.A.K. and A.A.K.; software, B.V.R.; supervision, R.A.K. and A.A.K.; visualization, B.V.R.; writing—original draft, B.V.R.; writing—review and editing, A.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data from this work are available upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hammad, A.W.A.; da Costa, B.B.F.; Soares, C.A.P.; Haddad, A.N. The Use of Unmanned Aerial Vehicles for Dynamic Site Layout Planning in Large-Scale Construction Projects. *Buildings* **2021**, *11*, 602. [CrossRef]
2. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
3. Eun, J.; Song, B.D.; Lee, S.; Lim, D.-E. Mathematical Investigation on the Sustainability of UAV Logistics. *Sustainability* **2019**, *11*, 5932. [CrossRef]
4. Joshi, G.P.; Alenezi, F.; Thirumoorthy, G.; Dutta, A.K.; You, J. Ensemble of Deep Learning-Based Multimodal Remote Sensing Image Classification Model on Unmanned Aerial Vehicle Networks. *Mathematics* **2021**, *9*, 2984. [CrossRef]
5. Lindner, G.; Schraml, K.; Mansberger, R.; Hübl, J. UAV monitoring and documentation of a large landslide. *Appl. Geomat.* **2016**, *8*, 1–11. [CrossRef]
6. Pérez-Álvarez, R.; Sedano-Cibrián, J.; de Luis-Ruiz, J.M.; Fernández-Maroto, G.; Pereda-García, R. Mining Exploration with UAV, Low-Cost Thermal Cameras and GIS Tools—Application to the Specific Case of the Complex Sulfides Hosted in Carbonates of Udías (Cantabria, Spain). *Minerals* **2022**, *12*, 140. [CrossRef]
7. Kloepper, L.N.; Kinniry, M. Recording animal vocalizations from a UAV: Bat echolocation during roost re-entry. *Sci. Rep.* **2018**, *8*, 7779. [CrossRef]

8. Chodorek, A.; Chodorek, R.R.; Yastrebov, A. Weather Sensing in an Urban Environment with the Use of a UAV and WebRTC-Based Platform: A Pilot Study. *Sensors* **2021**, *21*, 7113. [CrossRef]
9. Chen, W.K. *Applied Graph Theory*, 1st ed.; North-Holland Publishing Company: Amsterdam, The Netherlands, 2012.
10. Pasqualetti, F.; Franchi, A.; Bullo, F. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Trans. Robot.* **2012**, *28*, 592–606. [CrossRef]
11. Portugal, D.; Rocha, R. Msp algorithm: Multi-robot patrolling based on territory allocation using balanced graph partitioning. *Proc. 2010 ACM Symp. Appl. Comput.* **2010**, *1*, 1271–1276. [CrossRef]
12. Elor, Y.; Bruckstein, A.M. Multi-a(ge)nt Graph Patrolling and Partitioning. In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Milan, Italy, 15–18 September 2009; pp. 52–57. [CrossRef]
13. Legovich, Y.S.; Diane, S.A.K.; Rusakov, K.D. Integration of Modern Technologies for Solving Territory Patroling Problems with the Use of Heterogeneous Autonomous Robotic Systems. In Proceedings of the 2018 Eleventh International Conference Management of large-scale system development MLSD, Moscow, Russia, 1–3 October 2018. [CrossRef]
14. Caraballo, L.E.; Díaz-Báñez, J.M.; Fabila-Monroy, R.; Hidalgo-Toscano, C. Stochastic strategies for patrolling a terrain with a synchronized multi-robot system. *Eur. J. Oper. Res.* **2022**, *301*, 1099–1116. [CrossRef]
15. Basak, A.; Fang, F.; Nguyen, T.H.; Kiekintveld, C. Abstraction methods for solving graph-based security games. *Int. Conf. Auton. Agents Multiagent Syst.* **2016**, *10003*, 13–33. [CrossRef]
16. Pan, J.-S.; Song, P.-C.; Chu, S.-C.; Peng, Y.-J. Improved Compact Cuckoo Search Algorithm Applied to Location of Drone Logistics Hub. *Mathematics* **2020**, *8*, 333. [CrossRef]
17. Li, Y.; Zhang, W.; Li, P.; Ning, Y.; Suo, C. A Method for Autonomous Navigation and Positioning of UAV Based on Electric Field Array Detection. *Sensors* **2021**, *21*, 1146. [CrossRef]
18. Raymond, A.S. *Stratigraphic Sedimentary Inversion Using Paths in Graphs*; M.Sc., Federal University of Rio de Janeiro: Rio de Janeiro, Brazil, 2017.
19. Goncharenko, V.I.; Kochkarov, A.A.; Yatskin, D.V.; Rumiantsev, B.V. Modeling the Detection of Moving Objects by Means of a Spatially Distributed Continuous Monitoring System with a Dynamic Structure. *Adv. Syst. Sci. Appl.* **2022**, *22*, 1–10. [CrossRef]
20. Luo, H.; Zhang, P.; Wang, J.; Wang, G.; Meng, F. Traffic Patrolling Routing Problem with Drones in an Urban Road System. *Sensors* **2019**, *19*, 5164. [CrossRef]
21. Deı, V.G.; Klinz, B.; Woeginger, G.J. Exact algorithms for the Hamiltonian cycle problem in planar graphs. *Oper. Res. Lett.* **2006**, *34*, 269–274. [CrossRef]
22. Deogun, J.S.; Steiner, G. Polynomial algorithms for hamiltonian cycle in cocomparability graphs. *SIAM J. Comput.* **1994**, *23*, 520–552. [CrossRef]
23. Sahalot, A.; Shrimali, S. A comparative study of brute force method, nearest neighbour and greedy algorithms to solve the travelling salesman problem. *Int. J. Res. Eng. Technol.* **2014**, *2*, 59–72.
24. Karypis, G.; Kumar, V. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*; University of Minnesota Digital Conservancy: Minneapolis, MN, USA, 1997.
25. Kochkarov, R.; Kochkarov, A. Introduction to the Class of Prefractal Graphs. *Mathematics* **2022**, *10*, 2500. [CrossRef]
26. García-Díaz, J.; Rodríguez-Henríquez, L.M.X.; Pérez-Sansalvador, J.C.; Pomares-Hernández, S.E. Graph Burning: Mathematical Formulations and Optimal Solutions. *Mathematics* **2022**, *10*, 2777. [CrossRef]
27. Biswas, P. Hamiltonian (Graph, Source, Destination), MATLAB Central File Exchange 2022. Available online: https://www.mathworks.com/matlabcentral/fileexchange/51610-hamiltonian-graph-source-destination (accessed on 1 January 2022).
28. Sipser, M. *Introduction to the Theory of Computation*, 1st ed.; International Thomson Publishing: Stamford, CT, USA, 1996.