Evolutionary Algorithm with Geometrical Heuristics for Solving the Close Enough Traveling Salesman Problem: Application to the Trajectory Planning of an Unmanned Aerial Vehicle

Christophe Cariou, Laure Moiroux-Arvis, François Pinet, Jean-Pierre Chanet

Article

Evolutionary Algorithm with Geometrical Heuristics for Solving the Close Enough Traveling Salesman Problem: Application to the Trajectory Planning of an Unmanned Aerial Vehicle

Christophe Cariou *, Laure Moiroux-Arvis, François Pinet and Jean-Pierre Chanet

INRAE, UR TSCF, University Clermont Auvergne, 9 Av. Blaise Pascal CS 20085, F-63178 Aubière, France
* Correspondence: christophe.cariou@inrae.fr

Abstract: Evolutionary algorithms have been widely studied in the literature to find sub-optimal solutions to complex problems as the Traveling Salesman Problem (TSP). In such a problem, the target positions are usually static and punctually defined. The objective is to minimize a cost function as the minimal distance, time or energy. However, in some applications, as the one addressed in this paper—namely the data collection of buried sensor nodes by means of an Unmanned Aerial Vehicle— the targets are areas with varying sizes: they are defined with respect to the radio communication range of each node, ranging from a few meters to several hundred meters according to various parameters (e.g., soil moisture, burial depth, transmit power). The Unmanned Aerial Vehicle has to enter successively in these dynamic areas to collect the data, without the need to pass at the vertical of each node. Some areas can obviously intersect. That leads to solve the Close Enough TSP. To determine a sub-optimal trajectory for the Unmanned Aerial Vehicle, this paper presents an original and efficient strategy based on an evolutionary algorithm completed with geometrical heuristics. The performances of the algorithm are highlighted through scenarios with respectively 15 and 50 target locations. The results are analyzed with respect to the total route length. Finally, conclusions and future research directions are discussed.

Keywords: Close Enough Traveling Salesman Problem; evolutionary algorithm; genetic algorithm; Wireless Underground Sensor Networks; Unmanned Aerial Vehicle; optimal trajectory planning

1. Introduction

The recent advances in Unmanned Aerial Vehicles (UAVs), such as increased payload, operating time and autonomous flight capacities, enable today to consider their use in a growing number of applications. In automatic mode, they can even autonomously take off, follow a previously defined trajectory and landing, [1]. They can act, for example, in smart farming, environmental monitoring, emergency aid and parcel delivery [2,3]. Moreover, combined with Wireless Sensor Networks (WSNs), they can collect data through radio communication, play the role of mobile gateway or communication relay in case of connectivity failure, and perform accurate operations, see [4–6].

However, planning the trajectory to be followed by an Unmanned Aerial Vehicle (UAV) in automatic mode can rapidly become a challenging task, in particular when the minimization of a cost function, as distance, time or energy, is pursued due the limited flight time of the UAV. For example, in the case of several static waypoints to be visited by the UAV in a free-obstacle working space, the search for the optimal trajectory requires to solve the Traveling Salesman Problem (TSP) [7–9]. The waypoints can be regarded as the cities to be visited. The time complexity of the TSP increases exponentially with the number of cities according to $(nb_{cities} - 1)!$, leading rapidly to prohibitive calculation costs. For example, 25 cities represent already $24! \simeq 10^{23}$ possible routes. By considering a computer able to

calculate 1 billion routes per second, it would take 10 million years to a brute force algorithm to calculate all the possible combinations ($10^{23}/(10^9 * 60 * 60 * 24 * 365) \simeq 10^7$) to next find the best solution. An alternative is to accept to not find the optimal solution and seek a sub-optimal solution. Numerous algorithms have been proposed to that end in the literature. Some of them are inspired by the natural selection (e.g., genetic algorithms [10]) or animal behavior (e.g., Ant Colony Optimization algorithms [11], Particle Swarm Optimization algorithms [12]). These approaches are non-deterministic, iteratives, and a criteria is required to stop the algorithm (e.g., number of iterations, threshold reached). Additional constraints can be considered, leading to variants of TSP. In the classical TSP problem, called Symmetric TSP (STSP), the route between two cities can be performed indifferently in one direction or the other. In the Asymmetric TSP (ATSP), the travel cost can be different to travel from city A to city B, than to travel from city B to city A—see [13]. In the Precedence Constrained TSP (TSP-PC), an additional constraint is that a city A should not be visited unless the city B was visited first—see [14]. We can also mention the Capacited Vehicle Routing Problem (CVRP) where the vehicle has limited carrying capacity, requiring to go back to the depot when the vehicle is full—see [15]. Another variant is that the cities are not represented by points, i.e., an exact location, but by areas with various shapes called neighborhoods—see [16]. In this case, the vehicle has to enter successively within the areas without the need to pass through the center. The problem is called TSP with Neighborhoods (TSPN). A similar problem is the Close Enough TSP (CETSP) where the areas are defined by circles of different radius—see [17,18].

1.1. Application and Objective

The development of Wireless Underground Sensor Networks (WUSNs) is a recent research axis based on sensor nodes buried at a few dozen centimeters deep to not impact the aboveground activities [19–22], with as main applications the environmental monitoring and precision agriculture [23–25]. The radio communication ranges of the buried sensor nodes are however limited due to the high attenuation of the electromagnetic waves in soil. Our previous work highlighted that the communication ranges of radio devices based on LoRa technology can vary from a few meters to several hundred meters with respect to various parameters (e.g., burial depth, soil moisture, transmit power, frequency—see [26]. The idea consists to use an UAV to collect the data: that approach has the advantage to avoid the installation of numerous and costly gateway in the environment, and offer the possibility to rapidly and safely cover wide areas. To that end, the trajectory of the UAV has to be determined, entering successively within the communication ranges of each node to perform the data collection task with the minimal route length.

The objective of this paper is to propose a trajectory planning algorithm enabling an UAV to successively and optimally pass through several circular areas of different radius, representing the radio communication ranges of the buried sensor nodes, see the illustration in Figure 1 and the problem statement in Figure 2.
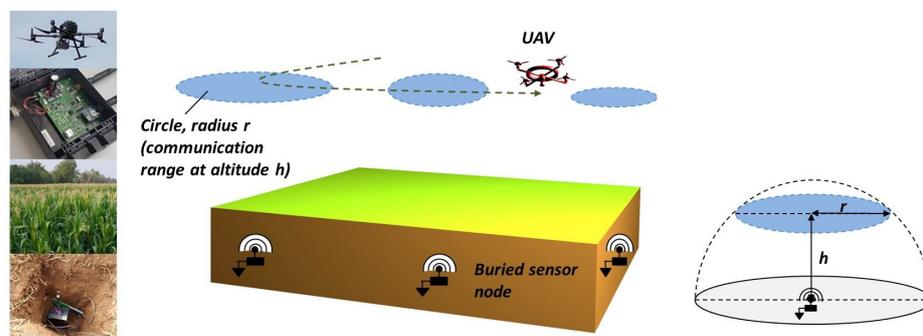

Figure 1. The objective is to optimally plan the trajectory of an UAV having to collect the data through radio communication of several buried sensor nodes distributed in the environment.
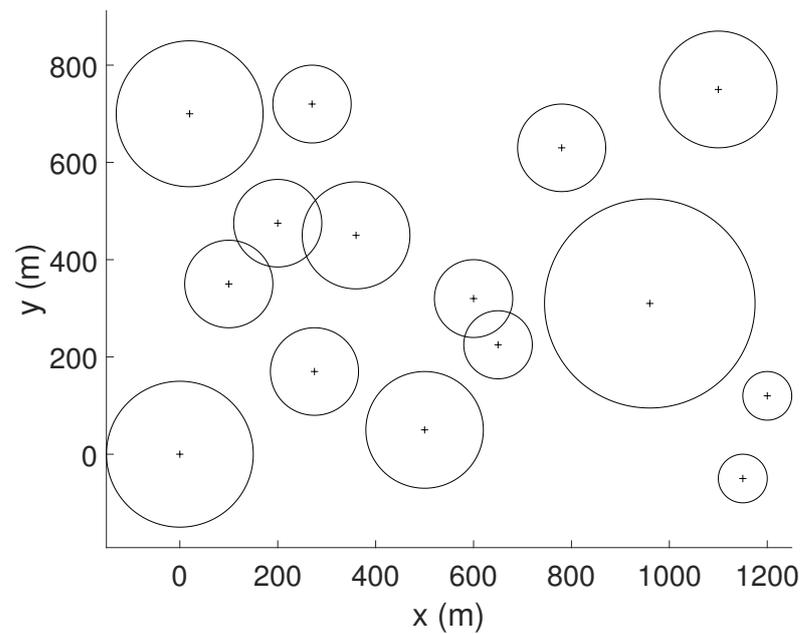
Figure 2. Problem statement in 2D: find the optimal trajectory passing through the circular disks representing the radio communication ranges of the buried sensor nodes at the altitude h.

It is assumed that the UAV flies at a constant altitude h, leading to targets in the form of circular areas of radius r. The UAV does not need to pass at the center of the circles but only within the disks to receive the data. The circles can intersect, offering the possibility to position the UAV at the intersections. We can also mention that it is assumed that the flying height is smaller than the smallest communication range of the nodes to access to all the data. That application necessitates clearly to address the Close Enough TSP.

1.2. Related Works

For several years, numerous algorithms have been developed for solving the TSP, mainly the symmetric TSP. A classification of some of these algorithms is presented in Figure 3.
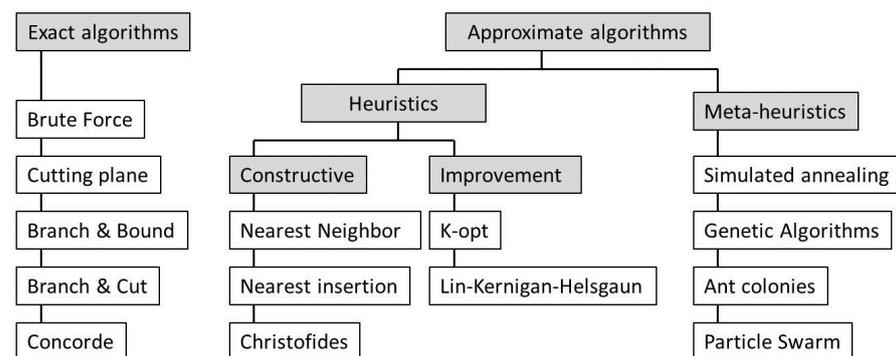
Figure 3. Algorithms for solving TSP.

The exact algorithms aim to find the optimal solution to a TSP. For example, the Brute Force algorithm calculates all the possible permutations, whereas the Cutting Plane [27], the Branch-and-Bound [28] and the Branch-and-Cut methods [29] limit the number of possibilities to reduce the calculation time. Concorde is an exact solver which combines several of these techniques, and is able to solve TSP with numerous cities (tens of thousands) [30]. The limitation of these exact algorithms is however the calculation time which

can be rapidly become excessive in case of numerous cities to be visited. A strategy can be to parallelize the algorithms and distribute the calculation on several CPU [31].

Another approach consists to develop approximate algorithms, with the objective to find a sub-optimal solution in a limited amount of time. As depicted in Figure 3, the approximate algorithms can be classified into three main groups, those based on constructive heuristics, those based on improvement heuristics and those based on meta-heuristics, see [32]. The principle of constructive heuristics is to start from an empty tour and iteratively extend the tour, for example by incorporating the nearest neighbor. The strategy of improvement heuristics is different because they start from a complete tour, which is progressively improved, for example using local permutations (e.g., K-opt [33], Lin-Kernigan-Helsgaun [34]). Meta-heuristic algorithms are generic algorithms able to deal with a broad range of problems. They have generally better performances than the heuristics algorithms, in particular by enabling to consider more complex and large TSP. They are usually based on the evolution of one or more starting populations (e.g., Genetic Algorithm [35], Ant Colony Optimization [36], Particle Swarm optimization [12]).

In the literature, the CETSP has been mainly addressed with meta-heuristic approaches. The difficulty of that problem is that the target areas are continuous, as at each point is associated a neighborhood, leading that even exact approaches are not easily applicable. The shapes of the neighborhoods are most of the time considered as discs with the same dimensions.

Initially, the CETSP was addressed by [37,38] to find the shortest route for a truck to perform automatic meter reading of customers from radio communication systems (e.g., remote electricity or water consumption measurements). Clustering-based and convex hull-based algorithms were proposed to group the customer nodes before applying a TSP solver. This approach is mainly based on the search for the adequate definition and position of groups of nodes, called supernodes, i.e., circles with a predetermined radius covering several customers. A CETSP with time window (CETSP-TW) was studied by [18] to take into account that each customer is available only for a limited period of time during the day. Ref. [39] introduced the Steiner Zones (intersection of the neighbohoods of degree k) and presented different methods to reduce the graphs. Ref. [40] proposed a branch-and-bound algorithm to address the CETSP in two and three dimensions, and [17] presented a method to estimate previously a tour length from the knowledge of several variables. Refs. [41,42] proposed discretization schemes to define first a finite number of points on the circumferences and inside the neighborhoods (hitting points), and next apply a TSP algorithm. Ref. [43] assumed disjoint regions and applied first a TSP algorithm to next modify the hitting points.

1.3. Contributions and Paper Organization

The definition of the optimal trajectory of an UAV to collect data from buried sensor nodes is an emerging application that, to the best of our knowledge, has not yet been addressed in the literature. The contribution of this paper is to propose an efficient algorithm, fast and easy programmable, to solve the CETSP applied to this application. The algorithm is decomposed of three steps, namely the definition of new target locations in case of intersections, the use of an evolutionary algorithm to find a sub-optimal order, and finally an iterative geometrical reduction to significantly reduce the trajectory length. This algorithm is programmed in a software developed in C++. In comparison to the previously presented work, this algorithm combines the resolution of a combinatorial problem (TSP) with geometrical heuristics, applied to a problem including the possibility of intersections.

The paper is organized as follows. The modeling and the methodology proposed are presented in Section 2. The algorithm is applied to two user cases. The results are reported in Section 3. The conclusion, discussions and future research directions are presented in Section 4.

2. Methodology

2.1. Modeling and Methodology Proposed

We consider a set of n circles of variable sizes, distributed in a Euclidean plane, see Figure 2. Each circle is defined by the coordinates (x_i, y_i, r_i) where x_i, y_i are the coordinates of the center of the circle and r_i its radius. The circles can intersect. The objective is to define a tour, having the minimal total distance, passing through all the circles. The trajectory will be composed of segment lines connecting the different circles.

To address this issue, we propose a methodology based on three successive steps. The first one determines new target locations for the nodes having some communication range intersections. The second one presents an evolutionary algorithm able to find a sub-optimal order to successively visit the target locations. Finally, from geometrical heuristics, the third one reduces significantly the length of the trajectory by passing either inside the areas (without the necessity to go by the center) or at the boundary of the areas.

2.2. First Step: New Target Locations in Case of Intersections

The communication ranges of some nodes can intersect. In this case, the UAV can be advantageously positioned at the intersections to communicate with several nodes simultaneously. The first step consists then to change the target locations by the points located at the center of the intersection areas, see Figure 4. That enables to reduce the number of target locations that will lead to reduce the tour length. For simplicity reasons, the communication ranges of these new points are defined by circles centered on the new point and whose radius is defined with respect to the boundaries of the previous circles.
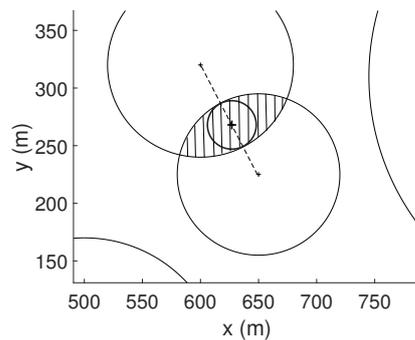
Figure 4. New target location defined by the intersection of communication ranges.

2.3. Second Step: Evolutionary Algorithm to Find a Sub Optimal Order

The second step consists to define an algorithm able to find a sub-optimal order to reach the different locations. The cost function is the shortest distance. The principle of the proposed algorithm is depicted in Figure 5. It is a genetic algorithm based on three mutations.

Figure 5. Principle of the evolutionary algorithm to generate a new population.

The algorithm starts with the creation of a set of initial solutions (list of node numbers whose order is randomly determined): that forms the initial population. This population is next divide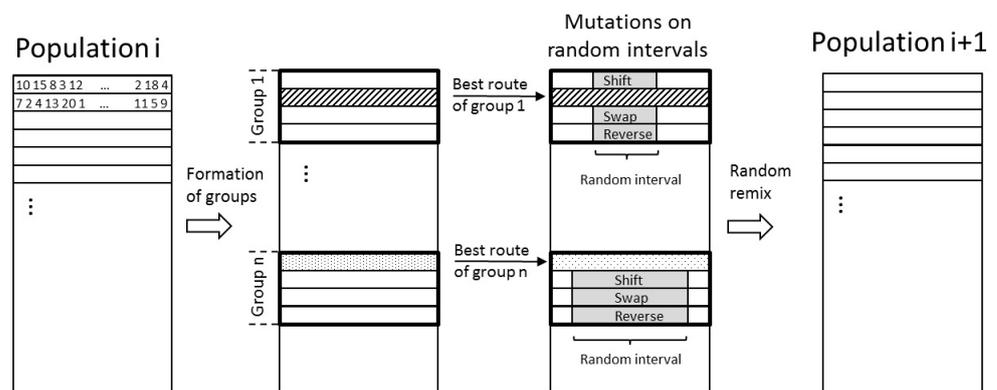d in groups: in each group, the best solution, that means the one with the route having the minimal Euclidean distance, is kept and called the survivor. The other routes of the group are replaced by copies of the survivor with mutations: on a random interval, the node orders are either reversed, shifted or swapped, see the details in Figure 6. All the routes are then randomly remixed to form the new population. This process is repeated until a stop criteria is satisfied (number of iterations). The advantage of such an algorithm is to be relatively simple and easy to program. The main parameters are the initial size chosen for the population and the number of iterations to stop the algorithm. A compromise has obviously to be reached between the calculation cost and the opportunity to find a good solution.

Survivor:	7	15	6	4	8	12	3	2	1	5	11	9	10	13	14
Random interval:	7	15	6	4	8	12	3	2	1	5	11	9	10	13	14
Shift:	7	15	2	6	4	8	12	3	1	5	11	9	10	13	14
Swap:	7	15	2	4	8	12	3	6	1	5	11	9	10	13	14
Reverse:	7	15	2	3	12	8	4	6	1	5	11	9	10	13	14

Figure 6. Mutations.

2.4. Step 3: Geometrical Reduction

An important aspect of the Close Enough TSP (CETSP) is that the trajectory can pass within the circular disks. To take this possibility into account, the strategy proposed consists, for each location, to consider the triangle formed with the previous and next locations, see Figure 7.
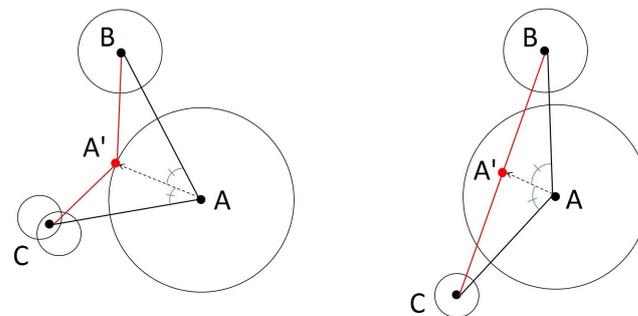

Figure 7. In the triangle ABC, relocation of the point A in A': (**left**) either on the boundary of the communication range if the limit is reached, (**right**) on the straight line (BC).

In the triangle ABC, the bisecting line is determined from the point A. If the length of the bisector is superior to the radius of the circle centered in A (left case on Figure 7), the point A is deplaced in A' (intersection of the bisector and the circle). On the contrary, the point A' is defined as being the intersection point between the bisector and the line (BC), (right case on Figure 7). This strategy is repeated on all the points in the order defined in step 2. At each iteration, the position of the previously calculated point is a vertex of the new triangle, see Figure 8. Once this procedure is applied on all the points, the trajectory length is significantly reduced. This procedure can be repeated until the shape of the trajectory stops changing.
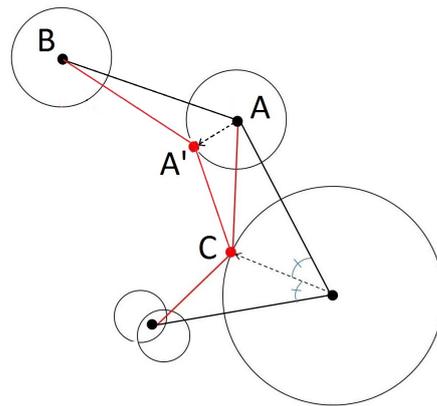
Figure 8. The previously calculated point C is the vertex of the next triangle ABC.

The algorithm is summarized in the diagram presented Figure 9.
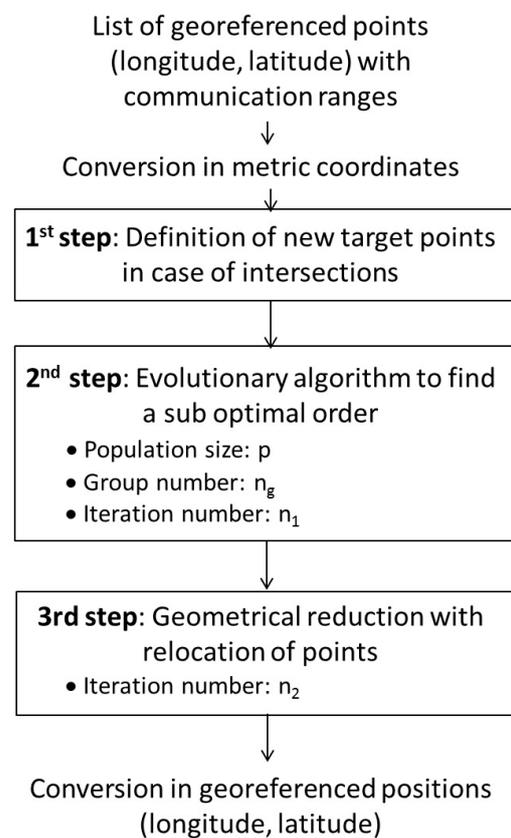
Figure 9. Diagram of the algorithm.

3. Results

3.1. Preliminary Test

The algorithm is first tested on a situation with a limited node numbers to easily highlight the different steps of the algorithm. To that end, 15 nodes are considered. That leads to $14! \simeq 8.10^{10}$ route possibles. The coordinates of the nodes are given in Appendix A, Table A1, and presented in Figure 10 (screenshot of our software developed in C++).
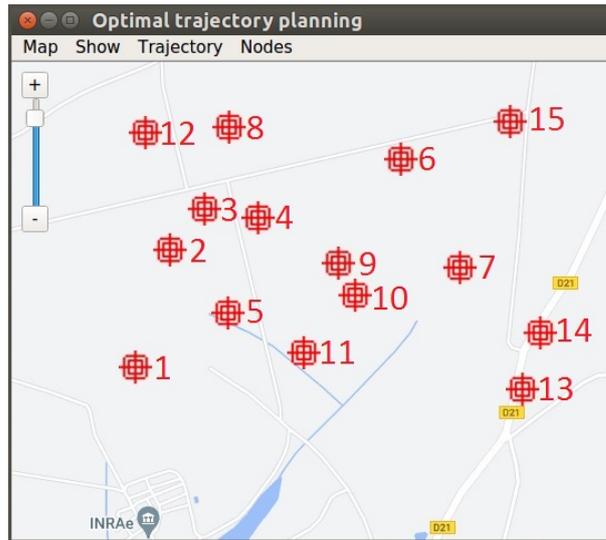
Figure 10. Display of the node locations in the developed software. At each node location is assigned a number from 1 to 15, see Appendix A, Table A1.

We first applied the evolutionary algorithm on the data without taking account the intersections and communication ranges. The parameters are a population of 100 combinations, 25 groups, and stop criteria defined as 1000 iterations. The result is the node order "15 6 4 8 12 3 2 1 5 11 9 10 13 14 7" leading to a total route length of 4635 m, see Figure 11.
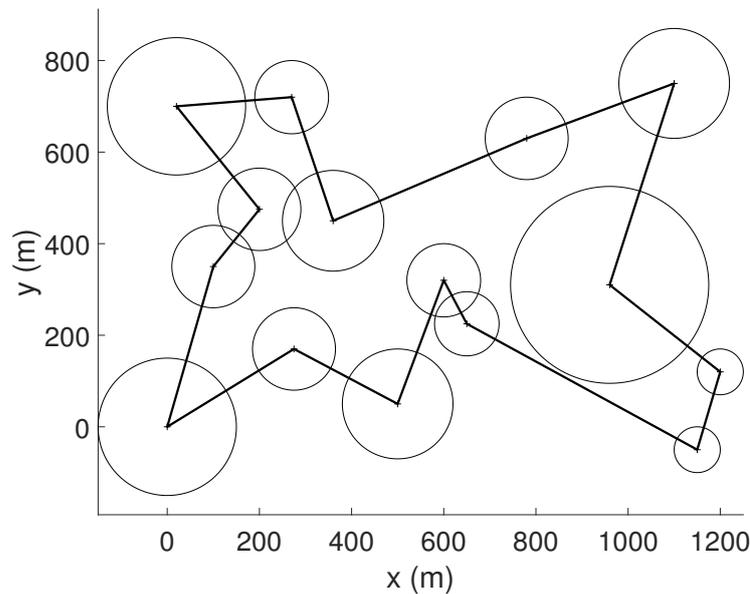
Figure 11. Initial test of the evolutionary algorithm on the 15 nodes: the route length is 4635 m.

The proposed algorithm in three steps is then applied. In step 1, the fact to take into account the intersections leads to reduce the node number to 13 as depicted on Figure 12a. Step 2 is the application of the evolutionary algorithm. The result is presented in Figure 12b. The total route length obtained is 4410 m. That means a reduction of 4.8% in comparison to the previous result. Step 3 is the geometrical reduction of the trajectory found at step 2. Two iterations are carried out. The first iteration reduces significantly the trajectory length, see Figure 12c. The second one improves further the result, see Figure 12d.
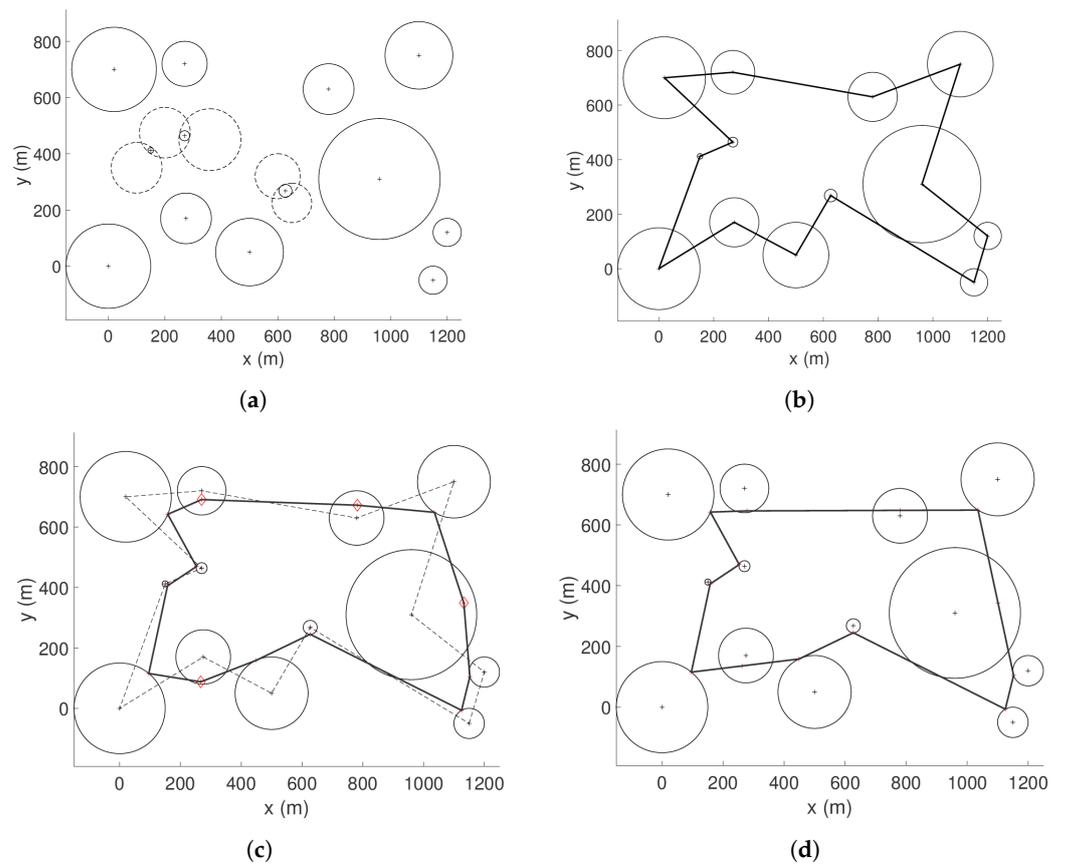
Figure 12. The three steps of the proposed algorithm. (**a**) Step 1: reduction of the node number; (**b**) Step 2: evolutionary algorithm; (**c**) Step 3: geometrical reduction (1st iteration); (**d**) Step 3: geometrical reduction (2nd iteration)

The final result is represented with the 15 nodes on Figure 13a,b. The trajectory length is 3267 m, that means a reduction of 29.5% with respect to the initial result (4635 m).
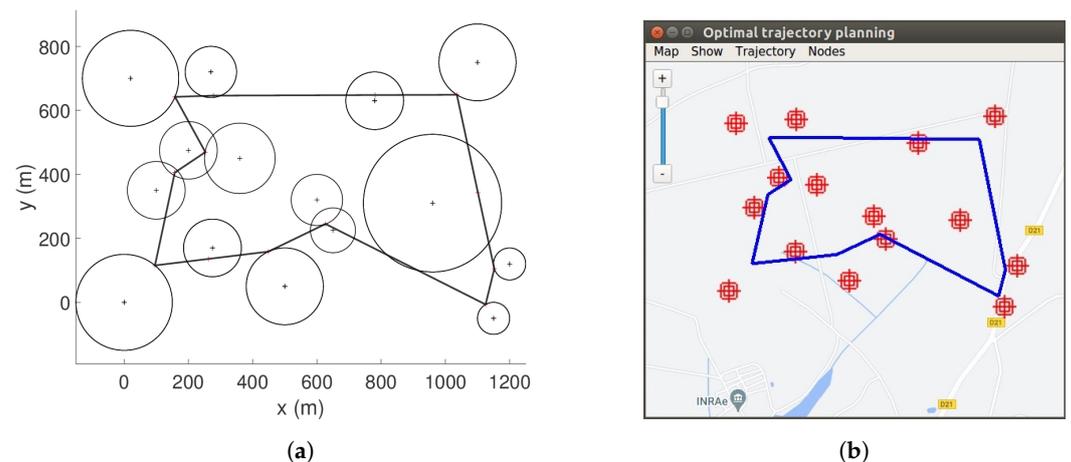
Figure 13. Display of the final result. (**a**) Final result; (**b**) Display in the software.

3.2. Test with 50 Nodes

The algorithm is then tested on a greater number of data (50 nodes). The coordinates of the nodes are given in Appendix A (Table A3) and are displayed in Figure 14. There are $49! \simeq 10^{63}$ possible routes.
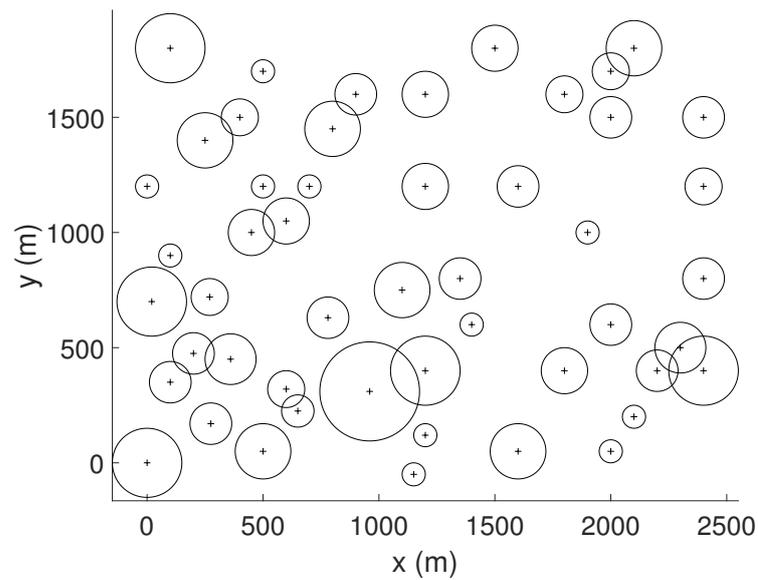
Figure 14. Node locations with their respective communication ranges.

To compare the benefit of the algorithm, we first applied the evolutionary algorithm (population size: 100, group number: 25, iteration number: 1000) on the data without taking account the intersections and communication ranges. The result is presented on Figure 15: the total route length is 15,249 m with the node order "11 5 1 2 3 4 8 12 20 19 36 37 41 42 22 21 23 24 38 43 49 39 47 46 40 44 45 48 34 25 33 50 30 29 35 28 27 26 31 32 18 17 15 6 16 14 13 7 9 10 11".
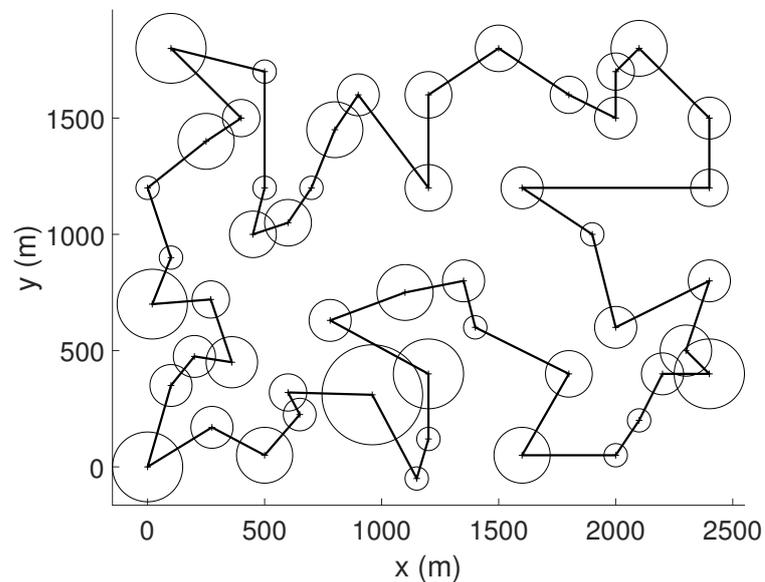
Figure 15. Initial test with the evolutionary algorithm on the 50 nodes: the route length is 15,249 m.

The algorithm in three steps is applied on these data. Step 1 found nine intersections, enabling to reduce the number of nodes. The result of the first step is presented in Figure 16a. Step 2 applies the evolutionary algorithm. A trajectory of length 13,382 m is obtained, see Figure 16b, that represents a reduction of 12.2% in comparison to the initial trajectory. Finally, step 3 is applied, based on geomtrical reductions. The result of step 3 is presented in Figure 16c. The length of the trajectory is 10,910 m. That represents a reduction of 28.4% with respect to the initial trajectory. The final result is represented with the 50 nodes in Figure 16d.

Figure 16. The proposed algorithm applied to the 50 nodes. (**a**) Step 1: reduction of the node number; (**b**) Step 2: evolutionary algorithm; (**c**) Step 3: geometrical reduction; (**d**) Final result.

4. Conclusions

This paper addresses the Close Enough TSP to find a sub-optimal trajectory in term of minimal distance for an UAV having to collect data from buried sensor nodes distributed in the environment. The problem is addressed in 2D with t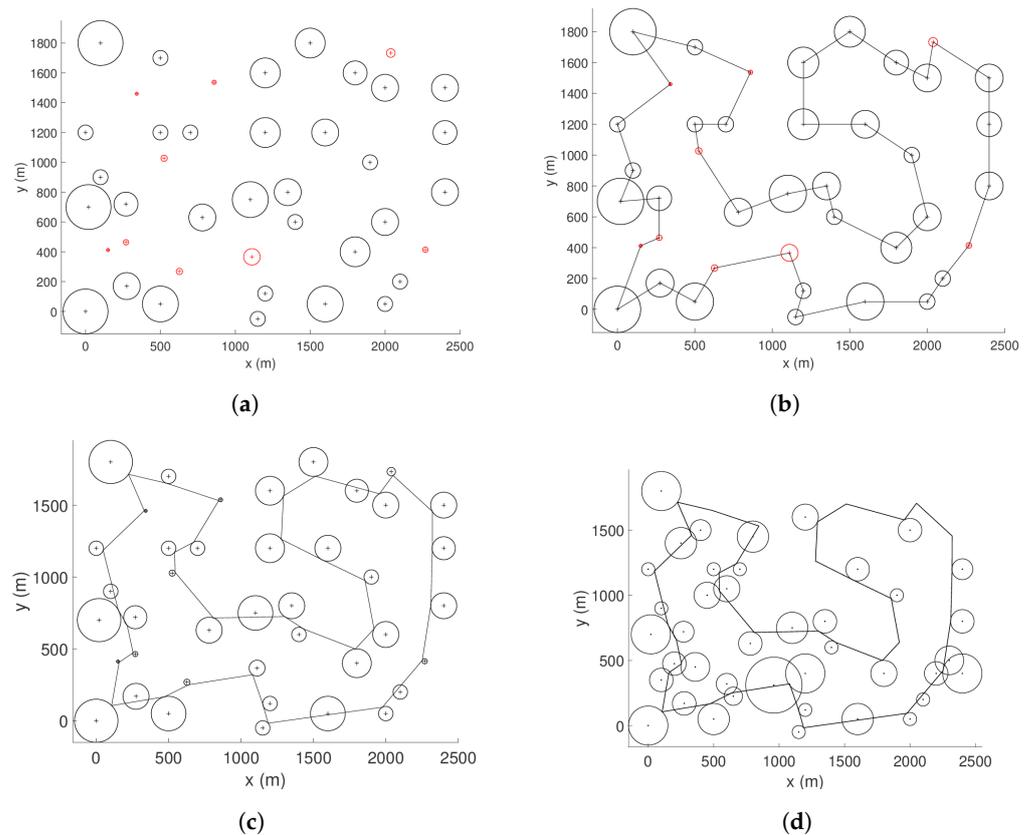he representation of the communication ranges by circular disks. An algorithm is proposed in three successive steps: the first one redefines some target points in case of intersections that enables to reduce the number of nodes. The second one is an evolutionary algorithm enables to find a sub-optimal order to visit the different nodes. Finally, the third step reduces significantly the length of the trajectory through geometrical heuristics. The algorithm is applied on two scenarios with respectively 15 and 50 nodes. The results highlight the obtention of limited trajectory length for the UAV. In particular, if the evolutionary algorithm is applied directly without geometrical heuristics on the 50 nodes, the obtained route length was 15,249 m. That distance was reduced by 28.4% (10,910 m) if the complete algorithm is applied.

This algorithm has been implemented in a software developed in C++. The advantage of this algorithm is to have a low implementation cost. It is easily programmable without the need of external libraries. The inputs, displayed in a geo-referenced map, are the GPS coordinates of the target locations with their respective communication ranges. These coordinates are first converted into Cartesian ones, to apply next the algorithm. The evolutionary algorithm proposed in this paper (genetic algorithm), as well as the geometrical heuristics, are easy to program and deliver a sub-optimal solution in a few seconds. The output is the trajectory to be followed by the UAV, with the successive waypoints.

5. Discussions and Future Work

5.1. Comparison with Other Works

To evaluate the performance of the algorithm proposed in this paper, we tested our algorithm on the "eil5" benchmark from TSPLIB composed of 51 nodes. The coordinates of the nodes and the corresponding radius are given in Appendix A (Table A3). We obtained

the result presented in Figure 17, with a trajectory length of 320.6 m. In comparison with other works on this same benchmark, the approximation algorithm developed by [41] led to a trajectory with a length of 354 m, see [43]. The algorithm proposed by [43], based on the combination of a TSP solver and heuristics, found a very similar trajectory in comparison to the one depicted in Figure 17, with a length around 320 m. That solution seems to be the optimal one. That result clearly highlights the performance of our algorithm having the capacity to deliver pertinent results for such Close Enough TSP.
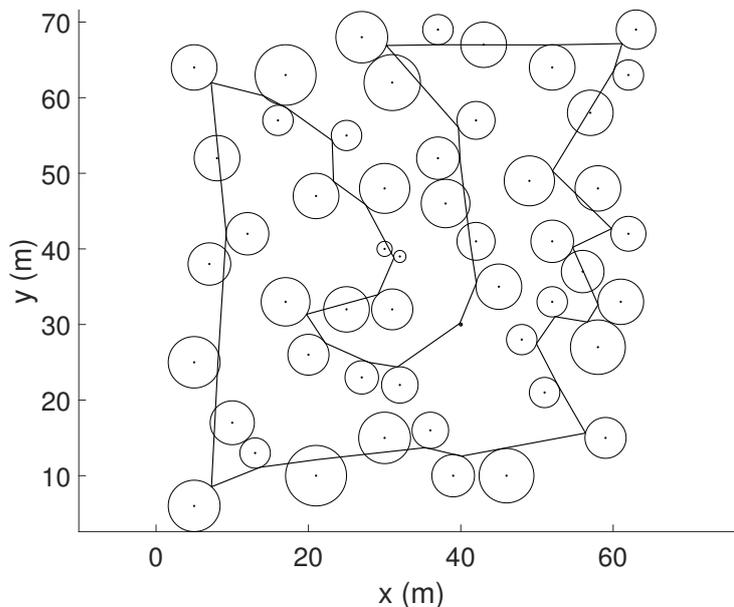

Figure 17. Test of our algorithm on the "eil5" benchmark from TSPLIB composed of 51 nodes. The length of the obtained trajectory is 320.6 m.

5.2. Future Work

The CETSP was evaluated in this paper in 2D. The assumption was made that the soil can be considered as a plane and that the UAV flies at a constant altitude (in a free-obstacle workspace). A first interesting perspective is to determine the optimal altitude for the UAV. In fact, the configuration of the problem changes with respect to the flight altitude as highlighted with the illustration on Figure 18. A flight at low altitude is more dangerous for the UAV with the possibility to encounter numerous obstacles (e.g., trees), but better in terms of quality of communication with the buried sensor nodes. On the contrary, a flight at high altitude is safer but the areas of the communication ranges are reduced.
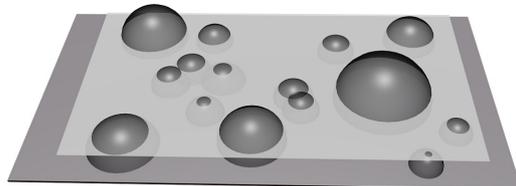
Figure 18. The communication ranges of the buried sensor nodes are represented by hemispherical shapes. The problem consists to find the optimal altitude of the UAV.

The problem could also be approached directly in three dimensions to take into account the altitude variations of the ground. The approach presented in this paper will be completed to that end: the georeferenced points will be defined by 3D coordinates (longitude, latitude, altitude). In the first step, the intersections of the communication range will be determined by spheres. In the second step, the calculation of the Euclidean distance in the evolutionary algorithm will be calculated in 3D. Finally, in the third step, the geometrical reduction will be carried out by considering 3D shapes.

Another interesting way of improvement is to add some constraints on the curvature variations of the trajectory. In fact, the kinematic and dynamic behavior of the UAV could be advantageously considered to avoid sharp movements of the UAV, all the more at high speed. The consideration of obstacles in the workspace of the UAV could also be required when the UAV flies at low altitude, e.g., in case of very short communication ranges of the buried sensor nodes. A preliminary flight could also be performed to learn the actual communication ranges of the buried sensor nodes and model the variations with respect to different environmental measurements. Finally, it would be interesting to complete the trajectory with temporal aspects, in particular to take into account the time to collect the data at each node location.

Abbreviations

The following abbreviations are used in this manuscript:

ATSP	Asymmetric Traveling Salesman Problem
CETSP	Close Enough Traveling Salesman Problem
CVRP	Capacited Vehicle Routing Problem
STSP	Symmetric Traveling Salesman Problem
TSP	Traveling Salesman Problem
TSPN	Traveling Salesman Problem with Neighborhoods
TSP-PC	Precedence Constrained Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle
WUSN	Wireless Underground Sensor Network

Appendix A

Table A1. Georeferenced and metric coordinates of 15 nodes to be visited with their respective radio communication ranges: benchmark used to test the algorithm in the Section 3.1.

n	Latitude (°)	Longitude (°)	x (m)	y (m)	Range (m)
1	46.343386	3.434335	0	0	150
2	46.346520	3.435697	100	350	90
3	46.347632	3.437019	200	475	90
4	46.347387	3.439093	360	450	110
5	46.344879	3.437938	275	170	90
6	46.348953	3.444582	780	630	90
7	46.346051	3.446861	960	310	215
8	46.349827	3.437973	270	720	80
9	46.346187	3.442187	600	320	80
10	46.345326	3.442819	650	225	70
11	46.343771	3.440838	500	50	120

Table A1. *Cont.*

n	Latitude (°)	Longitude (°)	x (m)	y (m)	Range (m)
12	46.349679	3.434721	20	700	150
13	46.342789	3.449263	1150	−50	50
14	46.344312	3.449944	1200	120	50
15	46.349992	3.448761	1100	750	120

Table A2. Benchmark of 50 nodes used to test the algorithm in the Section 3.2.

n	x (m)	y (m)	Radius (m)	n	x (m)	y (m)	Radius (m)
1	0	0	150	26	2000	50	50
2	100	350	90	27	2100	200	50
3	200	475	90	28	2200	400	90
4	360	450	110	29	2300	500	110
5	275	170	90	30	2400	800	90
6	780	630	90	31	1600	50	120
7	960	310	215	32	1800	400	100
8	270	720	80	33	1900	1000	50
9	600	320	80	34	2400	1200	80
10	650	225	70	35	2400	400	150
11	500	50	120	36	250	1400	120
12	20	700	150	37	400	1500	80
13	1150	−50	50	38	800	1450	120
14	1200	120	50	39	1200	1600	100
15	1100	750	120	40	2000	1500	90
16	1200	400	150	41	100	1800	150
17	1350	800	90	42	500	1700	50
18	1400	600	50	43	900	1600	90
19	0	1200	50	44	2000	1700	80
20	100	900	50	45	2100	1800	120
21	450	1000	100	46	1800	1600	80
22	500	1200	50	47	1500	1800	100
23	600	1050	100	48	2400	1500	90
24	700	1200	50	49	1200	1200	100
25	1600	1200	90	50	2000	600	90

Table A3. Benchmark of 51 nodes (eil5 from TSPLIB) used to test the algorithm in the Section 5.1.

n	x (m)	y (m)	Radius (m)	n	x (m)	y (m)	Radius (m)
1	37	52	2.8	27	30	48	3.3
2	49	49	3.3	28	43	67	3.0
3	52	64	3.0	29	58	48	3.0
4	20	26	2.7	30	58	27	3.6
5	40	30	0.2	31	37	69	2.0
6	21	47	3.0	32	38	46	3.2
7	17	63	4.0	33	46	10	3.6
8	31	62	3.7	34	61	33	3.0
9	52	33	2.0	35	62	63	2.0
10	51	21	2.0	36	63	69	2.6
11	42	41	2.5	37	32	22	2.4
12	31	32	2.7	38	45	35	3.0
13	5	25	3.4	39	59	15	2.5
14	12	42	2.8	40	5	6	3.4
15	36	16	2.4	41	10	17	2.9

Table A3. *Cont.*

n	x (m)	y (m)	Radius (m)	n	x (m)	y (m)	Radius (m)
16	52	41	2.8	42	21	10	4.0
17	27	23	2.2	43	5	64	3.0
18	17	33	3.2	44	30	15	3.4
19	13	13	2.0	45	39	10	2.8
20	57	58	3.0	46	32	39	0.8
21	62	42	2.3	47	25	32	3.0
22	42	57	2.5	48	25	55	2.0
23	16	57	2.0	49	48	28	2.0
24	8	52	3.0	50	56	37	2.8
25	7	38	2.8	51	30	40	1.0
26	27	68	3.4				

References

1. Grlj, C.G.; Krznar, N.; Pranjic, M. A Decade of UAV Docking Stations: A Brief Overview of Mobile and Fixed Landing Platforms. *Drones* **2022**, *6*, 7. [CrossRef]
2. Aslan, M.F.; Durdu, A.; Sabanci, K.; Ropelewska, E.; Gultekin, S.S. A Comprehensive Survey of the Recent Studies with UAV for Precision Agriculture in Open Fields and Greenhouses. *Appl. Sci.* **2022**, *12*, 1047. [CrossRef]
3. Srivastava, S.K.; Seng, K.P.; Ang, L.M.; Pachas, A.N.A.; Lewis, T. Drone-Based Environmental Monitoring and Image Processing Approaches for Resource Estimates of Private Native Forest. *Sensors* **2022**, *22*, 7872. [CrossRef] [PubMed]
4. Nguyen, M.T.; Nguyen, C.V.; Do, H.T.; Hua, H.T.; Tran, T.A.; Nguyen, A.D.; Ala, G.; Viola, F. UAV-Assisted Data Collection in Wireless Sensor Networks: A Comprehensive Survey. *Electronics* **2021**, *21*, 2603. [CrossRef]
5. Faiçal, B.S.; Pessin, G.; Filho, G.P.R.; Carvalho, C.P.L.F.; Gomes, P.H.; Ueyama, J. Fine-tuning of UAV control rules for spraying pesticides on crop fields: An approach for dynamic environments. *Int. J. Artif. Intell. Tools* **2016**, *25*, 1660003. [CrossRef]
6. Ueyama, J.; Freitas, H.; Faiçal, B.S.; Filho, G.P.R.; Fini, P.; Pessin, G.; Gomes, P.H.; Villas, L.A. Exploiting the use of unmanned aerial vehicles to provide resilience in wireless sensor networks. *IEEE Commun. Mag.* **2014**, *52*, 81–87. [CrossRef]
7. Cinar, A.; Korkmaz, S.; Kiran, M. A discrete tree-seed algorithm for solving symmetric traveling salesman problem. *Eng. Sci. Technol. International J.* **2020**, *23*, 879–890. [CrossRef]
8. Tuani, A.; Keedwell, E.; Collett, M. Heterogenous adaptive ant colony optimization with 3-opt local search for the Travelling Salesman Problem. *Appl. Soft Comput.* **2020**, *97*, 106720. [CrossRef]
9. Mersmann, O.; Bischl, B.; Bossek, J.; Trautmann, H.; Wagner, M.; Neumann, F. Local search and the Traveling Salesman Problem: A feature-based characterization of problem hardness. In Proceedings of the International Conference on Learning and Intelligent Optimization, Paris, France, 16–20 January 2012; Springer: Berlin/Heidelberg, Germany, 2012. [CrossRef]
10. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
11. Qu, F.; Yu, W.; Xiao, K.; Liu, C.; Liu, W. Trajectory generation and optimization using the mutual learning and adaptive colony algorithm in uneven environments. *Appl. Sci.* **2022**, *12*, 4629. [CrossRef]
12. Emambocus, B.A.S.; Jasser, M.B.; Hamzah, M.; Mustapha, A.; Amphawan, A. An enhanced swap sequence-based particle swarm optimization algorithm to Solve TSP. *IEEE Access* **2021**, *9*, 164820–164836. [CrossRef]
13. Gambardella, L.M.; Dorigo, M. Solving symmetric and asymmetric TSPs by ant colonies. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 622–627. [CrossRef]
14. Sung, J.; Jeong, B. An adaptive evolutionary algorithm for traveling salesman problem with precedence constraints. *Sci. World J.* **2014**, *2014*, 313767. [CrossRef]
15. Carwalo, T.; Thankappan, J.; Patil, V. Capacitated vehicle routing problem. In Proceedings of the 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), Tokyo, Japan, 7–8 April 2017; pp. 17–21. [CrossRef]
16. Berg, M.; Gudmundsson, J.; Katz, M. TSP with neighborhoods of varying size. In Proceedings of the 10th Annual European Symposium, Rome, Italy, 17–21 September 2002; pp. 187–199._20. [CrossRef]
17. Sinha Roy, D.; Golden, B.; Wang, X.;Wasil, E. Estimating the tour length for the close enough traveling salesman problem. *Algorithms* **2021**, *14*, 123. [CrossRef]
18. Semami, S.; Toulni, H.; Elbyed, A. The close enough traveling salesman problem with time window. *Int. J. Circuits Syst. Signal Process.* **2019**, *13*, 579–584.
19. Huang, H.; Shi, J.; Wang, F.; Zhang, D. Theoretical and experimental studies on the signal propagation in soil for wireless underground sensor networks. *Sensors* **2020**, *20*, 2580. [CrossRef]
20. Saeed, N.; Alouini, M.S.; Al-Naffouri, T. Towards the Internet of Underground Things: A systematic survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3443–3466. [CrossRef]

21. Salam, A.; Raza, U. Current advances in Internet of Underground Things. In *Signals in the Soil*; Springer Nature Switzerland AG: Cham, Switzerland, 2020; Chapter 10; pp. 321–356. [CrossRef]

22. Vuran, M.C.; Salam, A.; Wong, R.; Irmak, S. Internet of underground things: Sensing and communications on the field for precision agriculture. In Proceedings of the IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 5–8 February 2018. [CrossRef]

23. Silva, A.R.; Vuran, M.C. (CPS)2: Integration of center pivot systems with wireless underground sensor networks for autonomous precision agriculture. In Proceedings of the IEEE/ACM International Conference on Cyber-Physical Systems, Stockholm, Sweden, 13–15 April 2010; pp. 10–13. [CrossRef]

24. Akyildiz, I.F.; Stuntebeck, E.P. Wireless underground sensor networks: Research challenges. *Ad Hoc Netw.* **2006**, *4*, 669–686. [CrossRef]

25. Ferreira, C.B.M.; Peixoto, V.F.; de Brito, J.A.G.; de Monteiro, A.F.A.; de Assis, L.S.; Henriques, F.R. UnderApp: A system for remote monitoring of landslides based on wireless underground sensor networks. In Proceedings of the WTIC, Rio de Janeiro, Brazil, 1–2 April 2019.

26. Moiroux-Arvis, L.; Cariou, C.; Chanet, J.P. Evaluation of LoRa technology in 433-MHz and 868-MHz for underground to aboveground data transmission. *Comput. Electron. Agric.* **2022**, *194*, 106770. [CrossRef]

27. Fleischmann, B. A cutting plane procedure for the travelling salesman problem on road networks. *Eur. J. Oper. Res.* **1985**, *21*, 307–317. [CrossRef]

28. Clausen, J. *Branch and Bound Algorithms-Principles and Examples*; Department of Computer Science, University of Copenhagen: Copenhagen, Denmark, 1999. [CrossRef]

29. Laporte, G. The traveling salesman problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **1992**, *59*, 231–247.

30. Sanches, D.; Whitley, D.; Tinos, R. Improving an exact solver for the traveling salesman problem using partition crossover. In Proceedings of the Genetic and Evolutionary Computation Conference, Berlin, Germany, 15–19 July 2017; pp. 337–344. [CrossRef]

31. Ozden, S.G.; Smith, A.E.; Gue, K.R. Solving large batches of traveling salesman problems with parallel and distributed computing. *Comput. Oper. Res.* **2017**, *85*, 87–96. [CrossRef]

32. Hahsler, M.; Hornik, K. TSP—Infracstructure for the traveling salesperson problem. *J. Stat. Softw.* **2007**, *23*, 1–21. [CrossRef]

33. Helsgaun, K. An Effective Implementation of K-Opt Moves for the Lin-Kernighan TSP Heuristic. Ph.D. Thesis, Roskilde University, Roskilde, Danmark, 2006. [CrossRef]

34. Helsgaun, K. *An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems*; Roskilde University: Roskilde, Denmark, 2017. [CrossRef]

35. Razali, N.M.; Geraghty, J. Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the World Congress on Engineering*; International Association of Engineers: Hong Kong, China, 2011; Volume 2. [CrossRef]

36. Yang, J.; Shi, X.; Marchese, M.; Liang, Y. An ant colony optimization method for generalized TSP problem. *Prog. Nat. Sci.* **2008**, *18*, 1417–1422.

37. Gulczynski, D.; Heath, J.; Price, C. Close enough traveling salesman problem: A discussion of several heuristics. *Perspect. Oper. Res.* **2006**, *36*, 271–283. [CrossRef]

38. Dong, J.; Yang, N.; Chen, M. Heuristic approaches for a TSP variant: The automatic meter reading shortest tour problem. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*; Springer: New York, NY, USA, 2007.

39. Mennell, W.; Golden, B.; Wasil, E. A steiner-zone heuristic for solving the close-enough traveling salesman problem. In Proceedings of the 12th INFORMS Computing Society Conference, Homeland Defense, Monterey, CA, USA, 9–11 January 2011.

40. Coutinho, W.P.; Nascimento, R.Q.; Pessoa, A.A.; Subramanian, A. A Branch-and-Bound algorithm for the Close-Enough Traveling Salesman Problem. *Informs J. Comput.* **2016**, *28*, 752–765. [CrossRef]

41. Elbassioni, K.; Fishkin, A.; Mustafa, N.; Sitters, R. Approximation Algorithms for Euclidian Group TSP. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Lisbon, Portugal, 11–15 July 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1115–1126. [CrossRef]

42. Carrabs, F.; Cerrone, C.; Cerulli, R.; Gaudioso, M. A novel discretization scheme for the Close Enough Traveling Salesman Problem. *Comput. Oper. Res.* **2007**, *78*, 163–171. [CrossRef]

43. Yuan, B.; Orlowska, M.; Sadiq, S. On the optimal robot routing problem in wireless sensor networks. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1252–1261. [CrossRef]