Optimal route planning of an Unmanned Aerial Vehicle for data collection of agricultural sensors

Christophe Cariou, Laure Moiroux-Arvis, Fatiha Bendali, Jean Mailfert

Optimal route planning of an Unmanned Aerial Vehicle for data collection of agricultural sensors

Christophe Cariou
Université Clermont Auvergne
INRAE, UR TSCF
F-63178 Aubière, France
christophe.cariou@inrae.fr

Laure Moiroux-Arvis
Université Clermont Auvergne
INRAE, UR TSCF
F-63178 Aubière, France
laure.moiroux-arvis@inrae.fr

Fatiha Bendali
Université Clermont Auvergne
LIMOS, UMR CNRS 6158
F-63171 Aubière, France
fatiha.bendali-mailfert@limos.fr

Jean Mailfert
Université Clermont Auvergne
LIMOS, UMR CNRS 6158
F-63171 Aubière, France
jean.mailfert@uca.fr

***Abstract*—The development of the Internet of Things is essential in agriculture to meet the challenges of the agro-ecological transition. In fact, by installing communicating sensors directly in the fields, the health of ecosystems can be accurately monitored (e.g. soil conditions, crop growth, loss of biodiversity), the processes optimally controlled (e.g. irrigation systems), and the crop production better managed (e.g. optimal decision making). However, forwarding the measurements from a massive number of IoT-based sensors distributed in the fields to the internet can be a challenging task, all the more in case of energy, cost, latency, data rate or connectivity constraints. The conventional technologies, as the connection to a cellular network, a gateway or a nano-satellite at low earth orbit, may in fact not met these constraints. This paper investigates a new paradigm based on a data collector embedded on an Unmanned Aerial Vehicle (UAV). This approach has numerous advantages (e.g. no need of infrastructures, no subscription fees, operate in white areas, reduce the transmitter power of the communicating sensors). However, as the operating time of an UAV is limited, the length of the flight trajectories to visit all the sensors and collect the data must be minimized. To address this issue, the communication ranges of the sensors are first modeled as hemispheres. The Close Enough Traveling Salesman Problem (CE-TSP) is then investigated at different flying heights. To solve this problem, an algorithm based on three successive parts, a graph reduction, a partheno-genetic algorithm and heuristic rules, is developed. This algorithm is tested on data sets involving a massive number of communicating sensors with various communication ranges, as well as on a real agricultural case study. The results highlight the performances of the method proposed and open the way to future perspectives for data collection of IoT-based sensors by means of UAVs.**

***Index Terms*—Internet of Things (IoT), Unmanned Aerial Vehicle (UAV), Close Enough Traveling Salesman Problem (CE-TSP), data collection, partheno-genetic algorithm, agroecology.**

I. Introduction

With the increase of temperatures, the alteration of seasonalities and more frequent and intense extreme weather events, climate change has become a reality, impacting significantly the ecosystems, the crop growth and the agricultural production [1]–[3]. To better understand the phenomena, the development of smart environment monitoring systems based on the Internet of Things (IoT) is currently booming, whether it be to monitor the state of the soils, assess the impact of agricultural practices or initiate sustainable actions [4], [5]. By combining the data collected from various sensors distributed in the fields with other sources of information (e.g. weather forecast, agronomic models), the agricultural production intends to adapt to the climate variability, for example by optimizing the irrigation processes or developing new agricultural practices, more efficient, more sustainable and more climate resilient [6], [7]. However, forwarding the data from a massive number of IoT-based sensors to the internet can be a challenging task, all the more in case of energy, cost, latency, data rate and connectivity constraints. The conventional technologies to achieve this data transmission do not in fact always meet these constraints. For example, the connection to a cellular network, mainly based on NB-IoT and LTE-M technologies [8], [9], requires to have a cellular base station at proximity (a few kilometers in rural environment), the power consumption for the communications is also relatively high with a limited data rate, and fees must be paid to use the service, see Fig. 2a. The use and/or implementation of a gateway positioned at proximity, based on Low Power Wide Area Networks (LPWAN), as LoRa or SigFox, enables to reach long distances (several kilometers) at low power, but with a very limited data rate, see [10] and Fig. 2b. Such gateways can moreover be complex to configure and some security concerns have to be carefully addressed [11]. Alternatively, the communications with nano-satellites at Low Earth Orbit

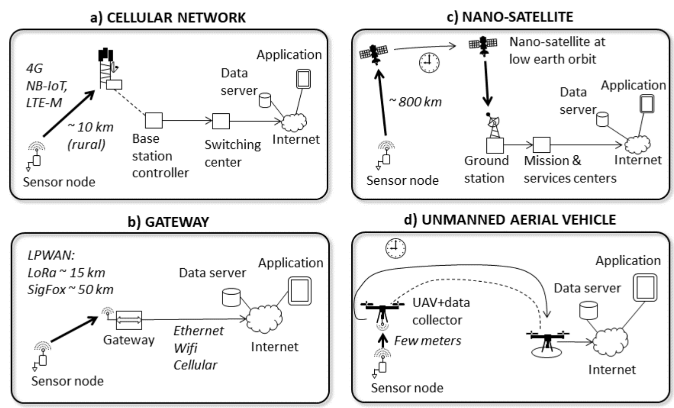


FIG. 1: Forward the data from a sensor node to the internet

This work was supported by the International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25".

(LEO) began being proposed to connect IoT-based sensors to the internet everywhere on the earth, see [12] and Fig. 2c. The data transmissions are enabled in a time window of only a few minutes per hour: the nano-satellite stores the data until it passes over a ground station and downloads the data to a web server. Unfortunately, at the time being, only messages of a few bytes per day can be transmitted and fees must also be paid to use the service.

In addition to these conventional approaches, a new one is currently emerging, based on the use of an Unmanned Aerial Vehicle (UAV) with an embedded data collector or a gateway onboard, see [13], [14] and Fig. 2d. This approach is particularly interesting as no infrastructures are required, no subscription fees have to be paid, and areas without connectivity possibilities can be instrumented (i.e. white areas). Moreover, as the UAV can come close to each sensor node, the transmission power required to transfer the data can be minimized to prolong the lifetime of the system. However, with this approach, the UAV has to visit successively all the sensor nodes to collect their data. As its operating time is limited, the minimization of the length of its flight trajectory has to be carefully investigated.

To address this issue, this paper considers a real-world scenario with a panel of real agricultural sensor nodes having limited communication ranges, see Fig. 2a, and with a data collector embedded on our experimental UAV having a limited flight duration, see Fig. 2b. The communication range of such sensors can be modeled by an hemisphere (see Fig. 3), whose center is the sensor and the radius varies with respect to different parameters as its transmit power, its buried depth, or the soil moisture, see our previous work [15]. The UAV has to enter within the hemispheres to collect the data of the sensors, without the need to pass through the centers. Obviously, the height of the UAV with respect to the ground is an important parameter to be considered as it will impact the final trajectory length. In addition, as it is depicted in Fig. 4, the communication ranges can intersect. However, as the data collector can communicate with several sensor nodes simultaneously, the passage of the UAV through the intersection areas can be investigated to reduce the total flight trajectory length.

Clearly, the trajectory planning problem of the UAV, having to pass successively through a large number of hemispheres

(a) (b)

FIG. 2: a) Buried sensor node in a pasture with a limited communication range (from 20 to 250 m). b) Experimental UAV with one hour of autonomy (Tundra, Hexadrone, payload 4 kg, span 1.83 m)

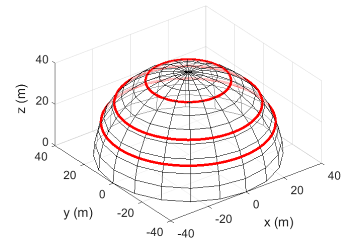


FIG. 3: Communication range of a sensor node (shape of an hemisphere). If the UAV flies at a constant height (e.g. 15, 25 or 35 m), the target area is a circle (in red) with a variable radius.

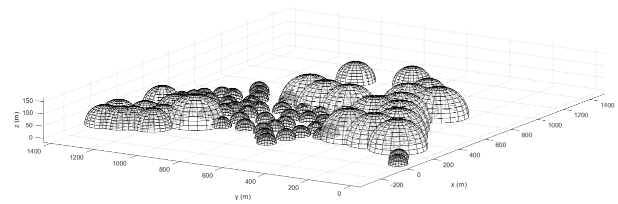


FIG. 4: Case study with 70 sensors. Some communication ranges intersect: the UAV can pass through the intersections to reduce the length of its flight trajectory to visit all the nodes.

to collect the data, can be considered as a Close Enough Traveling Salesman Problem (CE-TSP) in three dimensions, the cost function being the minimization of the total flight trajectory length. The target areas of the CE-TSP being continuous areas (hemispheres) and the number of sensor nodes being potentially important, solving this problem with an exact method is particularly complex, all the more without involving prohibitive calculation costs.

To address that problem, this paper is organized as follows. The section 2 presents a state of the art on the resolution of the CE-TSP. The section 3 presents the metaheuristic algorithm developed to solve this problem based on three successive parts, namely a graph reduction, a partheno-genetic algorithm and heuristic rules. The section 4 tests the algorithm on several data sets and a real agricultural case study, and analyzes the obtained results. Finally, conclusion and future research directions are presented in section 5.

II. Related work

The CE-TSP, a variant of the well-known Traveling Salesman Problem (TSP), was initially addressed to find the shortest route for a truck having to remotely read electricity or water meters at customer's residences equipped with Radio Frequency Identification (RFID) tags, see [16]. In such a problem, the communication range of the customer nodes are represented by circles with an identical radius R. As the circles highly intersect, clustering-based and convex hull-based algorithms were investigated to regroup the targets, i.e. determine the position of supernodes for the truck to cover several customer nodes simultaneously. In the clustering-based algorithm, the working space is first tiled with hexagons with edge length R, which are next replaced by circles of radius R. The circles containing some customer nodes are kept, some are removed in case of redundancy and others merged to define at the end a limited number of supernodes. Alternatively, the hull-based algorithm is based on an iterative process: the convex hull containing all the customer nodes is first built. A supernode is next defined at the boundary of the hull. The convex hull is then updated by removing the customer nodes belonging to the previously defined supernode. At the end of the iterations, a finite number of disjoint supernodes are obtained. A TSP solver is then applied to optimally rely the centers of the supernodes.

Instead to find the optimal position of supernodes, another approach consists to consider the intersections of the communication ranges. For example, [17] considered some circles with the same radius, and called the intersections Steiner Zones of degree k (i.e. $SZ(k)$ means an intersection area formed with k circles). Once the SZ are defined, a point in each one is designated, and the TSP is solved on these points (e.g. using solvers as Concorde or Lin-Kernighan). Complex heuristics are finally proposed to adapt the chosen points in the SZ in order to reduce the final trajectory length.

A different way to consider the CE-TSP is to define a finite number of points on the circumferences and inside the circles (hitting points), leading to discretize the problem, before applying a TSP algorithm, see [18], [19]. A compromise has however to be found between the number of discretization points and the calculation costs. Alternatively, [20] investigated the exact resolution of the CE-TSP through the definition of mathematical models and linear constraints transmitted to the CPLEX solver. [21] added a new constraint by considering that each customer node is available only during a limited time window per day. Numerous metaheuristics were proposed to solve the CE-TSP, for example by [22] with a branch-and-bound algorithm, [23] with a genetic algorithm based on crossover operations, and [24] proposed a method to estimate the length of a tour from the knowledge of 14 variables (e.g. number of nodes, number of Steiner zones, minimum distance to the average node).

In view of these works, it appears that the two main approaches which have been investigated to solve the CE-TSP are the development of complex metaheuristics and the research for an exact solution through the use of a TSP solver (e.g. CPLEX, Concorde). However, real applications of the CE-TSP have been rarely addressed in the literature, all the more in three dimensions. The need to find quickly a feasible solution is also essential, for example in case of sudden changes in the communication ranges of the communicating sensors (e.g. impact of the soil moisture). To that end, we propose a method based on three successive parts, namely a graph reduction, a partheno-genetic algorithm and heuristic rules. This method is able to find rapidly a sub-optimal solution to the CE-TSP, that is required for the application of data collection of IoT-based sensors by means of UAV.

III. Method proposed

A. Step 1: graph reduction

As previously presented in Fig. 4, the communication ranges of the sensors (i.e hemispheres) can intersect. As the UAV is able to communicate with several nodes simultaneously, it is possible to reduce the size of the problem by positioning the UAV in the intersection areas. To that end, the first step of our algorithm proposes to remove the hemispheres involving overlapping areas and replacing them by a new target area.

Let consider a problem with n hemispheres. At a predetermined flying height, the intersection areas are defined as circles, see the red circles in Fig. 4. To reduce the graph, we propose to consider successively the N combinations of two circles ($N = n(n-1)/2$) in the data set. In a combination, the two circles are noted C_1 and C_2, their centers $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$, and their radius r_1 and r_2. The equations of C_1 and C_2 are (1) and (2). d_{12} is the Euclidian distance between p_1 and p_2, see (3).

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2 \tag{1}$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \tag{2}$$

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

Several situations occur with respect to d_{12}, see Fig. 5. If $d_{12} < |r_2 - r_1|$ or $d_{12} = |r_2 - r_1|$, one circle is completely contained within the other. In our algorithm, the smallest circle will be kept and the largest removed. If $|r_2 - r_1| < d_{12} < r_1 + r_2$, one circle is partially contained within the other. The two circles will be replaced by a new circle C_I of center $p_I(x_I, y_I)$ and radius r_I, created within the common area, see Fig. 6: the linear equation passing through p_1 and p_2 is calculated, and the intersection points (a, b, c, d) of this line with C_1 and C_2 are determined. b and c are respectively the points the closest to p_1 and p_2, enabling to define the coordinates of the center $p_I((x_b + x_c)/2, (y_b + y_c)/2)$ and the radius $r_I = \sqrt{(x_b - x_c)^2 + (y_b - y_c)^2}/2$ of the circle C_I. If $d_{12} = r_1 + r_2$, the circles are tangent externally. The two circles will be replaced by the intersection point. If $d_{12} > r_1 + r_2$, the circles do not overlap, they are both kept. The loop is repeated on the data set until no more intersection exists in the data set, i.e. the remaining circles are disjoint.

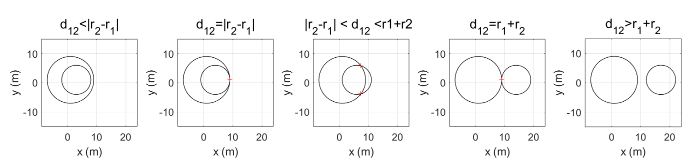


FIG. 5: Different possible configurations of two distinct circles

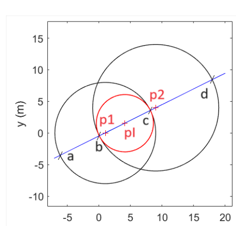


FIG. 6: A new target circle is built within the overlapping area

B. Step 2: partheno-genetic algorithm

At the end of the step 1, a set of N_b disjoint circles is obtained, numbered from 1 to N_b. The aim of the step 2 is to determine the optimal order that will minimize the Euclidian distance of the route passing through the centers of the circles. To that end, we propose a partheno-genetic algorithm based on four mutations (i.e. reverse, swap, right shift, left shift). The principle of the algorithm is depicted in Fig. 7.

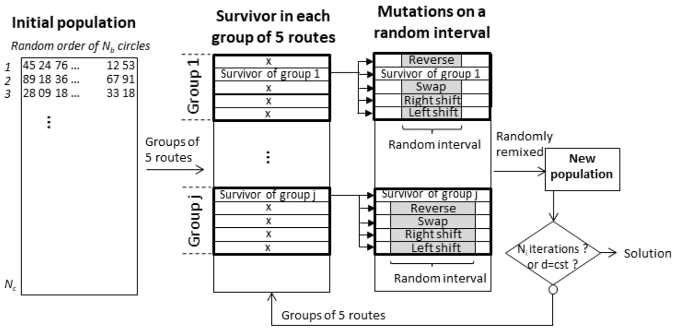


FIG. 7: Principle of the Partheno-Genetic Algorithm (PGA)

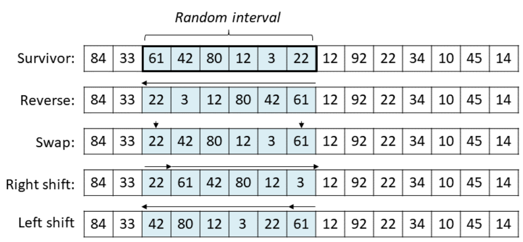


FIG. 8: Mutation operators

In this algorithm, there are no crossover operators in comparison to the Standard-Genetic Algorithm (SGA). The algorithm starts with the creation of an initial population composed of N_c possible routes randomly defined (N_c must be a multiple of 5). This population is next divided in $N_c/5$ groups of 5 routes: in each group, the Euclidian distance is calculated for each route, and the route which has the minimal distance is kept and called the survivor. The other routes of the group are removed and replaced by copies of the survivor with mutations: on a random interval, the circle orders are either reversed, swapped, right-shifted or left-shifted, see Fig. 8. Finally, all the routes are randomly remixed to form a new population. This process is repeated until a predetermined number of iterations N_i is reached or the length of the best route is no more changing during several iterations.

The advantage of this algorithm is to be simple and easy to program. The only parameters are N_c and N_i, respectively the size of the initial population and the number of iterations before stopping the algorithm. The choice of these parameters should be a compromise between the calculation cost and the possibility to find a good solution.

C. Step 3: heuristic rules

At the end of step 2, the trajectory connects the centers of the circles in a sub-optimal manner. However, in the CE-TSP, the trajectory can pass within the target areas without the need to pass through the center. We take this possibility into account by developing some heuristic rules, see Fig. 9.

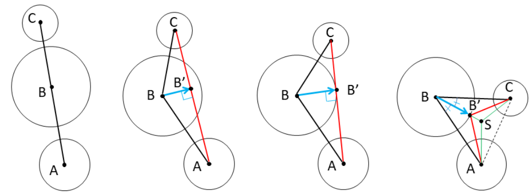


FIG. 9: Step 3: Heuristic rules for route reduction

In the route previously defined at step 2, let consider three successive circles C_A, C_B and C_C, and name their centers A, B and C, see Fig. 9. Four cases occur. a) if A, B and C are aligned (i.e. $det(\vec{AB}, \vec{BC}) = 0$), B remains unchanged. b) if the height $h_{BB'}$ of the triangle ABC from the vertex B is lower than the radius r_B of the circle C_B, B is replaced by B'. $h_{BB'}$ is given by (4) where p is the half perimeter of the triangle ABC ($p = (a + b + c)/2$).

$$h_{BB'} = \frac{2}{b}\sqrt{p(p-a)(p-b)(p-c)} \tag{4}$$

c) if $h_{BB'} = r_B$, this is a particularly case: B is replaced by B' which belongs also to C_B. d) if $h_{BB'} > r_B$, the bisecting line (BS) of the triangle ABC is considered: B is replaced in B' which is the intersection point of C_B and (BS), and belongs to the triangle ABC. The coordinates of the point S, the intersection point of the bisecting lines of the triangle ABC, are as follows:

$$S\left(\frac{ax_A + bx_B + cx_C}{a+b+c}, \frac{ay_A + by_B + cy_C}{a+b+c}\right) \tag{5}$$

These heuristic rules are applied on groups of three successive circles of the route determined in step 2. They are repeated until the trajectory obtained do not change.

IV. RESULTS

A. Case studies with a massive number of intersections

The method proposed is firstly experimented on a data set of 100 circles involving numerous intersections. The result obtained is given in Fig. 10. The step of graph reduction enables to pass from 100 circles to 24 circles. The route length after the genetic algorithm is 394.99 m, which is reduced to 360.93 m after the application of the heuristic rules. This final solution is obtained in 0.9 s, highlighting the low calculation cost of the method proposed.

The method is experimented on a second data set of 200 circles with a massive number of intersections. The result obtained is given in Fig. 11. The step of graph reduction enables to pass from 200 circles to 48 circles. The route length after the genetic algorithm is 525.53 m, which is reduced to 441.83 m after application of the heuristic rules. This final solution is obtained in 3.5s. These two first results clearly highlight the capacity of the method proposed to obtain rapidly a sub-optimal solution, while managing a high number of intersection areas.

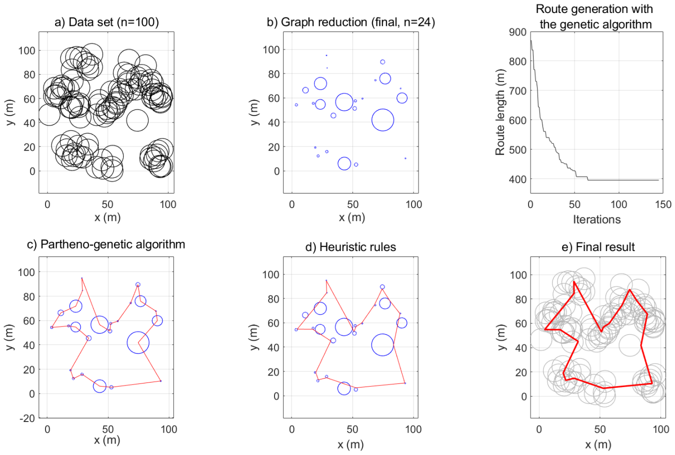


FIG. 10: Data set: TSPLIB, team1_100 (100 circles). The route length after the genetic algorithm (145 iterations) is 394.99 m. After the heuristics, the route length is 360.93 m.

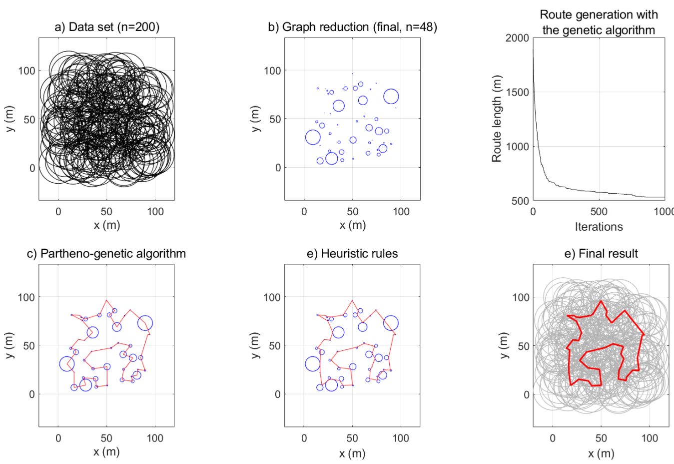
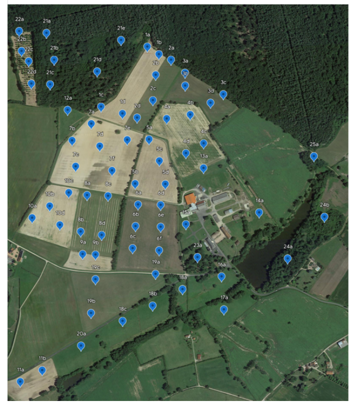


FIG. 11: Data set: TSPLIB, team2_200 (200 circles). The route length after the genetic algorithm (998 iterations) is 525.53 m. After the heuristics, the route length is 441.83 m.

B. Agricultural case study

A more realistic case study is then considered with 70 sensor nodes positioned in our experimental farm. They transmit the soil moisture and temperature measurements in LoRa at 868-MHz with three different communication ranges (40 m, 80 m and 120 m), see Fig. 12 and the 3D representation in Fig. 4. The positions of the sensor nodes are in WGS84 coordinates. These coordinates are converted into metrics one before applying the algorithm. The waypoints of the final trajectory will be finally converted into WGS84 and entered into the mission planner of the UAV.

In a first case, we consider that the UAV will fly at the constant height of 25 m. The intersection of the plan at this altitude and the hemispheres leads to circles of radius of respectively 31.22 m, 75.99 m, and 117.37 m. The results are presented in Fig. 13, 14 and 15. The final route length is 5.878 km. At low speed ($v = 5m/s$), that means that the UAV will fly about 20 minutes to achieve this trajectory, that is far below its maximal autonomy (one hour).



40 m

46.345427, 3.431582
46.345171, 3.432186
46.343406, 3.429239
46.343150, 3.430326
46.344999, 3.432788
46.344458, 3.432017
46.343611, 3.431864
46.343004, 3.431065
46.344539, 3.433495
46.344098, 3.433489
46.343796, 3.435436
46.343516, 3.434771
46.342957, 3.432606
46.343121, 3.433751
46.342142, 3.434402
46.341785, 3.433444
46.342234, 3.431659
46.340706, 3.431002
46.341505, 3.432169
46.340706, 3.432497
46.339962, 3.431111
46.339236, 3.430972
46.338497, 3.430806
46.339936, 3.432251
46.339236, 3.432616
46.338377, 3.432159
46.342790, 3.431688
46.342214, 3.427767
46.341322, 3.427963
46.342036, 3.429165
46.342256, 3.430541
46.341220, 3.429772
46.340333, 3.428573
46.339097, 3.428183
46.340302, 3.429586
46.338908, 3.429290
46.338244, 3.428321
46.338214, 3.428962
46.339532, 3.425760
46.339974, 3.426667
46.340434, 3.427639
46.339298, 3.427169
46.333818, 3.425166
46.334224, 3.426231

80 m

46.345981, 3.426509
46.345036, 3.426918
46.344178, 3.426688
46.344649, 3.429019
46.345713, 3.430288
46.346007, 3.425121
46.345387, 3.425273
46.345006, 3.425609
46.344259, 3.425723
46.338146, 3.434098
46.338088, 3.438662
46.339530, 3.440566
46.341647, 3.439981

120 m

46.343253, 3.427613
46.341276, 3.434410
46.340342, 3.433764
46.339288, 3.433560
46.337515, 3.435279
46.336333, 3.435453
46.337041, 3.433358
46.336475, 3.431809
46.335940, 3.430328
46.337592, 3.431995
46.336261, 3.428738
46.337390, 3.428966
46.335066, 3.428171

FIG. 12: 70 sensor nodes positioned in our experimental farm: Google Earth, Digital Globe, 46°20'22.35"N, 3°25'44.28"E, 280 m

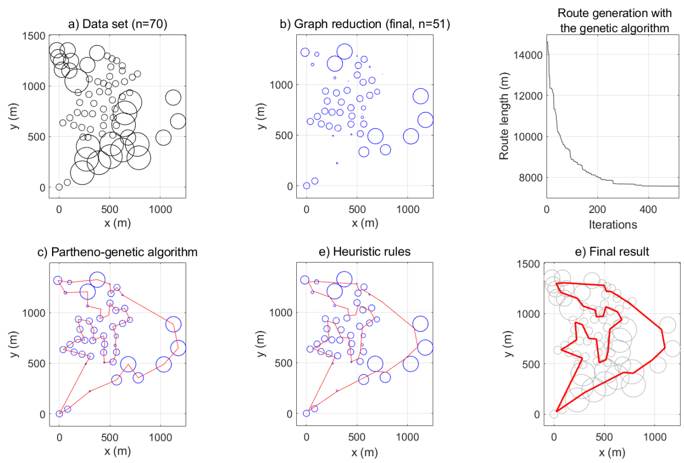


FIG. 13: The route length after the genetic algorithm is 7.128 km (519 iterations). After the heuristics, the route length is 5.878 km.

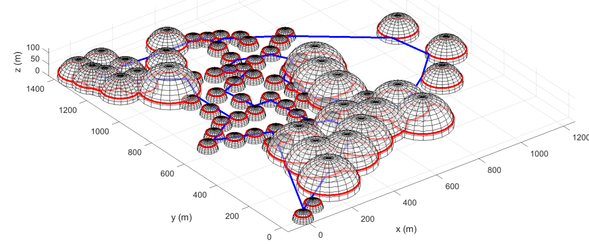


FIG. 14: Result when the UAV's height is 25 m

FIG. 15: The waypoints of the trajectory are reported in the mission planner of the UAV (Mission planner, ArduPilot).

Two other flying heights were tested, see Fig. 16 and Fig. 17. At 15 m, the circles are larger than at 25 m, the route length is reduced to 5.624 km. At 35 m, the circles are smaller, the route length is longer (6.254 km). At the speed $v = 5m/s$, the length difference represents 2 minutes more for the UAV, i.e. about 10%, that is not negligeable. In return, there are more risks of collision of the UAV with obstacles at low height (e.g. trees) than at high height.

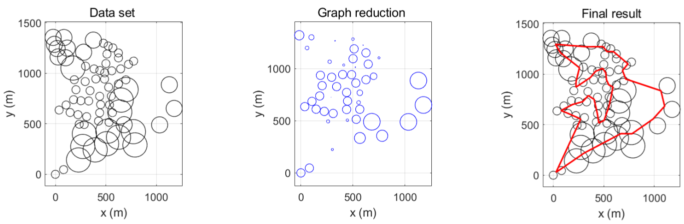


FIG. 16: UAV's height at 15 m. Tour length: 5.624 km

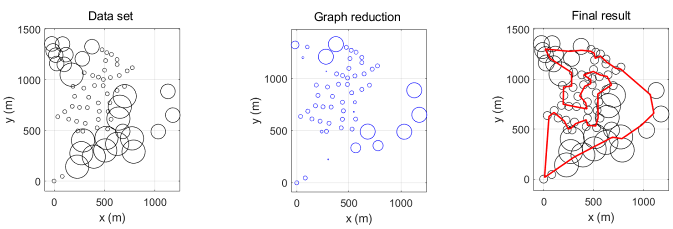


FIG. 17: UAV's height at 35 m. Tour length: 6.254 km

V. Conclusion and Future Work

This paper presents an efficient metaheuristic algorithm enabling to rapidly determine a sub-optimal trajectory for an UAV having to pass successively through target areas in the form of hemisphere, the target application being the optimal data collection of agricultural communicating sensors. The importance to take into account the height of the UAV in the algorithm is highlighted as it impacts clearly the final tour length. A flight at low altitude leads to a reduced tour length, but the presence of potential obstacles have to be carefully taken into account. To that end, the future research directions intend to add in the algorithm the knowledge of both the 3D obstacles and the ground topology from the digital terrain model, in order to improve further the UAV's route including flying height variations.

References

[1] FAO, Thinking about the future of food safety - a foresignt report, Food and Agriculture Organization of the United Nations, Rome (2022). doi:10.4060/cb8667en.

[2] J.-Y. Lee, J. Marotzke, G. Bala, L. Cao, S. Corti, J. P. Dunne, F. Engelbrecht, E. Fischer, J. C. Fyfe, C. Jones, A. Maycock, J. Mutemi, O. Ndiaye, S. Panickal, T. Zhou, Future global climate: Scenario-based projections and nearterm information, In: Climate Change 2021: The Physical Science Basis, Cambridge University Press, NY, USA, pp. 553–672 (2021). doi:10.1017/9781009157896.006.

[3] United Nations Environment Programme, Emissions gap report 2021: The heat is on – A world of climate promises not yet delivered, Nairobi (2021).

[4] A. Salam, Internet of Things for environmental sustainability and climate change, In: Internet of Things for Sustainable Community Development, Springer, Cham. pp. 33–69 (2020). doi:10.1007/978-3-030-35291-2_2.

[5] S. L. Ullo, G. R. Sinha, Advances in smart environment monitoring systems using IoT and sensors, Sensors 20, 3113 (2020). doi:10.3390/s20113113.

[6] R. A. Ahmed, E. E. Hemdan, W. El-Shafai, Z. A. Ahmed, E. M. El-Rabaie, F. E. A. El-Samie, Climate-smart agriculture using intelligent techniques, blockchain and Internet of Things: Concepts, challenges, and opportunities, Transactions on Emerging Telecommunications Technologies (2022). doi:10.1002/ett.4607.

[7] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, R. Kaliaperumal, Smart farming: Internet of Things (IoT)-based sustainable agriculture, Agriculture 12(10) (2022). doi:10.3390/agriculture12101745.

[8] H. J. Albeyboni and I. A. Ali, LPWAN technologies for IoT applications: a review, In: Journal of University of Duhok, 26(1):29-42, (2023). doi:10.26682/sjuod.2023.26.1.4.

[9] G. Valecce and P. Petruzzi and S. Strazzella and L. A. Grieco, NB-IoT for smart agriculture: Experiments from the field, 7th International Conference on Control, Decision and Information Technologies (CoDIT), Prague, Czech Republic, pp. 71-75, (2020). doi:10.1109/CoDIT49905.2020.9263860.

[10] L. Moiroux-Arvis, L. Royer, D. Sarramia, G. de Sousa, A. Claude, D. Latour, E. Roussel, O. Voldoire, P. Chardon, R. Vandaële, T. Améglio, J.-P. Chanet, Connecsens, a versatile IoT platform for environment monitoring: bring water to cloud, Sensors 23(6), 2896 (2023). doi:10.3390/s23062896.

[11] A. U. Mentsiev and T. R. Magomaev, Security threats of NB-IoT and countermeasures, In: IOP conference series, materials science and engineering, 862(5), (2020). doi:10.1109/JIOT.2021.3079567.

[12] A. Bouyedda, B. Barelaud, L. Gineste, Design and realization of an UHF frequency reconfigurable antenna for hybrid connectivity LPWAN and LEO satellite networks, Sensors 21, 5466 (2021). doi:10.3390/s21165466.

[13] C. Cariou, L. Moiroux-Arvis, F. Pinet, J.-P. Chanet, Data collection from buried sensor nodes by means of an Unmanned Aerial Vehicle, Sensors 22(15), 5926 (2022). doi:10.3390/s22155926.

[14] A. Andreadis and G. Giambene and R. Zambon, Low-power IoT for monitoring unconnected remote areas, In: Sensors, 23, 4481, (2023). doi:10.3390/s23094481.

[15] L. Moiroux-Arvis, C. Cariou, J.-P. Chanet, Evaluation of LoRa technology in 433-MHz and 868-MHz for underground to aboveground data transmission, Computers and Electronics in Agriculture 194 (2022). doi:10.1016/j.compag.2022.106770.

[16] J. Dong and N. Yang and M. Chen, Heuristic approaches for a TSP variant: The automatic meter reading shortest tour problem, In: Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies, NY, USA, (2007).

[17] W. Mennell and B. Golden and E. Wasil, 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, (2011).

[18] F. Carrabs and C. Cerrone and R. Cerulli and M. Gaudioso, A novel discretization scheme for the close enough traveling salesman problem, Comput. Oper. Res, 78, 163–171, (2007). doi:10.1016/j.cor.2016.09.003.

[19] B. Yuan and M. Orlowska and S. Sadiq, On the optimal robot routing problem in wireless sensor networks, In: IEEE Trans. Knowl. Data Eng., 19, 1252–1261, (2007).

[20] M. Hoang Ha and N. Bostel and A. Langevin and L. Rousseau, An exact algorithm for the close enough traveling salesman problem with arc covering constraints, Proceedings of the 1st International Conference on Operations Research and Enterprise Systems, 233–238, (2012).

[21] S. Semami and H. Toulni and A. Elbyed, The close enough traveling salesman problem with time window, Int. J. Circuits Syst. Signal Process. 13:579–584, (2019).

[22] W.P. Coutinho and R.Q. Nascimento and A.A. Pessoa and A. Subramanian, A branch-and-bound algorithm for the close-enough traveling salesman problem,, Informs J. Comput., 28, 752–765, (2016). doi:10.1287/ijoc.2016.0711.

[23] A.D. Placido and C. Archetti and C. Cerrone, A genetic algorithm for the close-enough traveling salesman problem with application to solar panels diagnostic reconnaissance,, Computers and Operations Research, Vol. 145, (2022). doi:10.1016/j.cor.2022.105831.

[24] D. Sinha Roy and B. Golden and X. Wang and E. Wasil, Estimating the tour length for the close enough traveling salesman problem,, Algorithms, 14, 123, (2021). doi:10.3390/a14040123.