Path planning for UAV search using growing area algorithm and clustering

Alexander Fedorov
Higher School of Economics
Saint Petersburg, Russia
afedorov2602@gmail.com

***Abstract*—This paper presents a novel approach to planning a path for unmanned aerial vehicles (UAVs) for searching lost targets under uncertainty. A new basic approach for fast and effective solutions is developed which outperforms standard local hill climbing (LHC) approach. This can be improved using a clusterization of probabilities in the area. The idea allows us to simplify the problem of planning a UAV path to a problem of choosing several clusters. Finally, the proposed method is compared with other methods in simulated scenarios. The comparison shows its high efficiency to solve searching problems.**

***Index Terms*—unmanned aerial vehicles (UAVs), path planning, object detection, probability distribution**

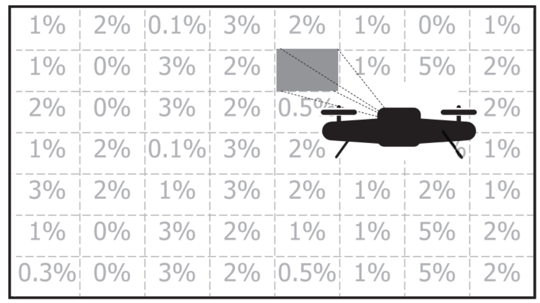

Fig. 1: Example of a probability grid [2]

I. Introduction

Unmanned aerial vehicles are gaining popularity both in civilian and military fields. They can be used for wildlife monitoring, target tracking, reconnaissance, surveillance, search and rescue, area patrolling and battle damage assessment. For many of these mission time is critical and should be minimized. Minimizing time in this paper means minimizing length of UAV paths.

Most of the problems mentioned above cannot be solved optimally in polynomial time. The complexity of a general searching problem is NP-hard [1]. This paper focuses on effective solutions which means that the proposed method can be only a suboptimal heuristic polynomial-time solution.

II. Problem formulation

There are various types of UAVs with different characteristics, but we will have the following assumptions:

- UAVs can maintain a constant height above the ground;
- UAVs have a gimballed camera;
- at some constant height, UAVs see below using the camera a square of size a on the ground;
- UAVs travel with known constant speed;
- UAVs can make a 90-degree turn if necessary.

In the paper, we supposed that the area of search is discretized into $N = W \times H$ squares of size r which means that flying above the center of a square at known constant height a UAV can see it fully. For each square (i, j) (denoted by the number of its column and row) the probability that the target is in this square is known and equal to $p_{i,j} \in [0, 1]$. See Fig. 1 for the example of such discretization with known probabilities.

A typical size of a probability grid for a search problem is tens of thousands squares [3]. The total probability in the search area should be equal to 1 and thus

$$\sum_{i,j} p_{i,j} = 1. \tag{1}$$

We denote a UAV path as $R = (R_1, \dots R_k)$, where $R_i \in \{1, \dots, W\} \times \{1, \dots, H\}$, the initial cell where the drone is located as $S \in \{1, \dots, W\} \times \{1, \dots, H\}$ and the length of the path as L.

Then the search problem then can be formulated as follows: Given probability grid with known probabilities $p_{i,j}$, the initial position of an UAV S and maximum length of the path L we should:

$$\begin{aligned} &\underset{R}{\text{maximize}} && \sum_{(x,y)\in R} p_{x,y} \\ &\text{subject to} && R_1 = S \\ &\text{and} && \sum_{i=1}^{k-1} distance(R_i, R_{i+1}) \leq L, \end{aligned}$$

where $distance(a, b)$ is Euclidean distance between centers of cells a and b.

In other words, we want to find a path that

- visits some squares and collects probabilities from them by performing a scan;
- has a length no more than L;
- maximizes probability collected.

Note that there is no benefit from visiting (scanning) the same square twice.

III. Related work

This problem is a special case of Orienteering problem [4] with a grid structure instead of a general case of a graph. Orienteering problem is a more complex version of traveling salesman problem (TSP), is known to be NP-hard and has no fully polynomial approximation scheme unless P=NP [5].

The considered problem may be solved with mixed integer linear programming (MILP) [6], genetic algorithm [7] and ant colony optimization [8]. Nevertheless, these approaches are effective only for a small number of nodes (less than 100).

The basic approach for fast solutions of this problem usually is local hill climbing [9]. It is a greedy solution in which in each step a UAV flies to one of neighbours of its current cell choosing the cell with the maximum probability. However, the main problem of this approach is that the drone can't leave an area of local maximum and fly to a better area unless it covers the area fully (see Fig. 2).

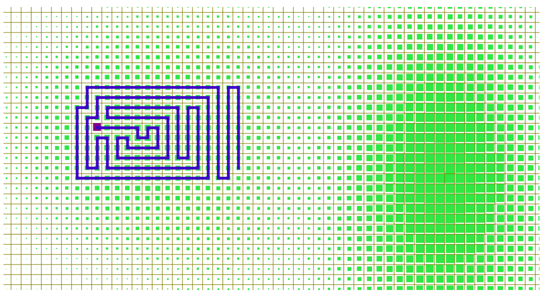

Fig. 2: Example of a problem of LHC when a drone can't get to an area of high probability. The more green a cell is the more probability it contains. The route is blue and the initial UAV position is a purple square.

Different authors tried to overcome the problem. Lanillos et al. supposed to calculate not only real probability the drone collects flying in a direction but also expected probability that it can collect there [10]. Moreover, he uses receding horizon optimization (RHC) to increase the number of steps the drone look ahead and to make routes more locally optimal. Yao et al. clusterize an area to detect areas of high probability [11]. In each of the clusters, he builds routes separately using RHC and then joins them. To use clustering the probability distribution is supposed to be good, i.e. we can clearly see subareas of high and low probability. Lin et. al proposes using of global warming effect optimization which allows the drone to ignore cells with low probability [3].

IV. Proposed algorithm

A. Issues of analogs

As it was stated before, effective (fast) solutions for the problem are usually based on local hill climbing algorithm. However, as a consequence, these algorithms ([10][11][3]) have the following issues (Fig. 3):

- Final routes visit a lot of cells twice. Visiting a cell more than once doesn't increase the probability of detecting. It doesn't always mean that the solution is non-optimal but can be usually avoided in such a way that the resulting route will have higher probability collected;
- A solution may miss cells with high probability because in some moment visiting them is locally non-optimal, but visiting them in future may be valuable. Local hill climbing algorithm is unlikely to visit these cells in future because in each step in chooses the next cell only from four adjacent cells.

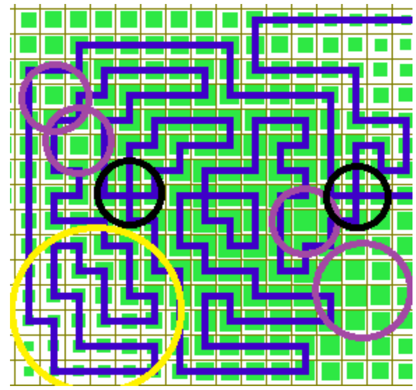

Fig. 3: Example of problems in solutions generated by algorithms based on local hill climbing. In black circles there are cells that are visited twice. In purple circles there are cells with high probability that are skipped by the solution. In the yellow circle there are cells that can be skipped instead of cells in purple circles.

B. Basic idea

Lemma. Any area of cells consisting of connected cells of double size (i.e. consisting of four small cells) may be covered with a hamiltonian cycle which goes through centers of small cells.

The proof of the lemma may be found in [12].

The algorithm to get the hamiltonian cycle is simple:

1) Build a spanning tree on cells of double size (Fig. 4). It can be built because the area is connected.
2) A path along the spanning tree is a hamiltonian cycle needed (Fig. 5)

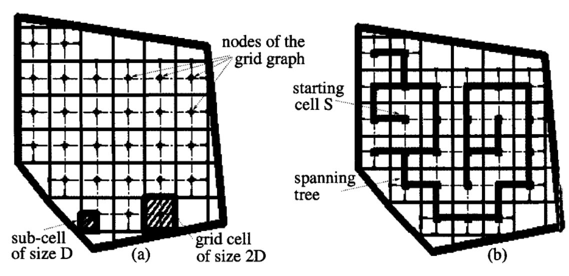

Fig. 4: (a) Area of cells of double size. (b) A spanning tree on cells of double size.[12]

This lemma brings us to the following idea: Instead of building a path we will build an area of cells of double size.

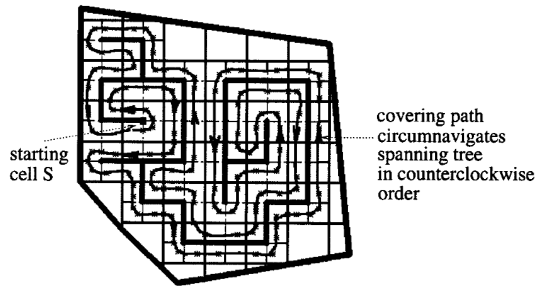

Fig. 5: A path along the spanning tree [12]

To maintain connectivity we may add to the area only cells that are neighboring to it. As in local hill climbing, we can use greedy algorithms, i.e. among all possible cells of double size add the one that contains more probability (this can be done effectively by using, e.g., a heap data structure). Note that at every step we preserve connectivity of an area of cells of double size and it is the only constraint needed to use the lemma.

After building the area we will build a hamiltonian cycle using the lemma and it will be the final route of a UAV. Note that as we will build a hamiltonian cycle we know how many cells we need to add to the area not to exceed the constraint on the length of the final route.

C. Improvement with clustering

This idea may be improved by the use of clustering.

Suppose we have detected all clusters (subareas of high probability). If we join centers of all clusters and the starting position of a drone by an area of cells of double size (e.g, using minimal spanning tree (MST) or Steiner tree, Fig. 6) and then run our greedy algorithm then the area will grow accurately in clusters (Fig. 7).

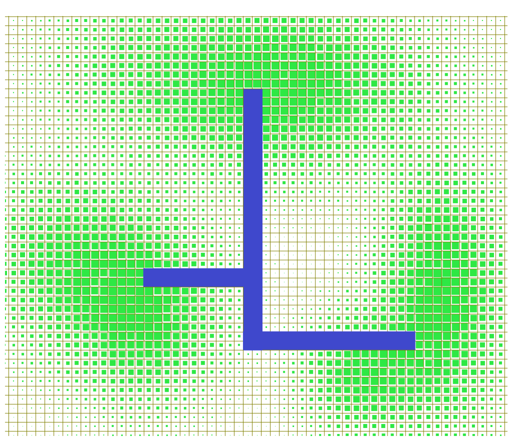

Fig. 6: Clusters joined by an area of cells of double size

In some cases (when the constraint on the length of the route is tight) it is better to join not all cluster but only a few of them. To choose what clusters to join we may iterate over all possibilities (if there are a small number of clusters, which is a pretty common case) or just try to pick the nearest cluster

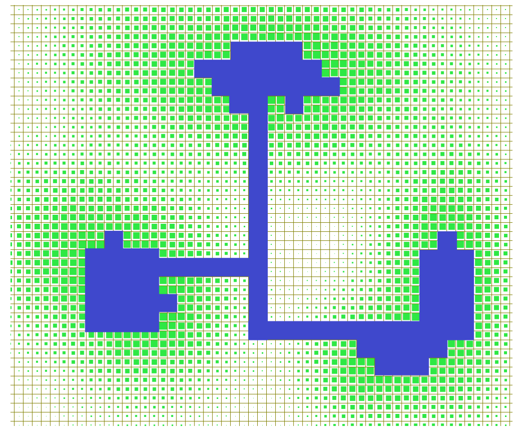

Fig. 7: Area after the end of the greedy algorithm

step by step (increasing complexity of an algorithm in a linear of the number of clusters time).

Then combining two ideas we get the following algorithm:

1) Choose what clusters to join.
2) Join them and the starting position of a UAV with an area of cells of double size.
3) Grow the area step by step while not exceeding the constraint on the route length.
4) Get the hamiltonian cycle from the area using the lemma.

The total complexity of steps 2-4 is $O(L \log N)$ if using a heap to pick the cell with the highest probability effectively.

We will call the algorithm "cluster area-growth algorithm".

V. Experiments

The proposed cluster area-growth algorithm was implemented and compared with local hill climbing algorithm and lhc-gw-conv algorithm from [3] (with global warming effect).

The comparison was performed on several test areas. Some of the areas were generated using several Gaussian distributions and some of these areas were obtained from height maps of real locations (because the form of those maps and real probability maps are similar). For every area, there were several measurements with different constraints on the length of the route (L). In the cells, there are total probabilities collected by solutions in given constraints on a particular test.

As it can be seen in Table I the proposed algorithm outperformed other algorithms almost in every case. Basic local hill climbing approach gives dramatically worse results in all tests. The proposed solution gained worse result only when constraint on the length of the route is tight. It is because other solutions were not obliged to return the drone to its starting position (i.e. to form a cycle). As a result, their routes got the opportunity to cover more clusters or to cover more distant clusters (see Fig. 8 and Fig. 9). However, as the constraint increases the proposed solution becomes to get better and better results in comparison with other solutions (see Fig. 10 and Fig. 11), because of the absence of those issues mentioned above.

TABLE I: Results of testing

	Test #1			Test #2		
	L=3000	L=7000	L=15000	L=3000	L=7000	L=15000
proposed	14%	**31.2%**	**60.1%**	8.1%	**19.3%**	**40%**
lhc	13.3%	23.7%	43.3%	6.2%	16.5%	35.2%
lhc-gw-conv	**14.1%**	30.8%	55.8%	**8.6%**	19.1%	38.2%

	Test #3			Test #4		
	L=3000	L=7000	L=15000	L=3000	L=7000	L=15000
proposed	**19.6%**	**39.9%**	**66.8%**	**24.4%**	**50.3%**	**84%**
lhc	19.2%	36.8%	62.2%	23.9%	42%	71.7%
lhc-gw-conv	19.4%	38.7%	65%	24.2%	48.7%	78.1%

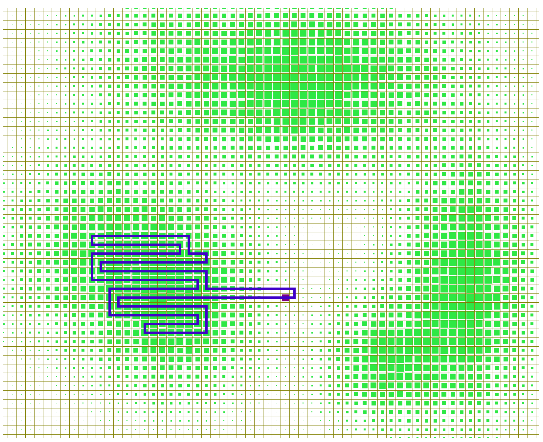

Fig. 8: Test 1 with constraint $L = 3000$, proposed solution

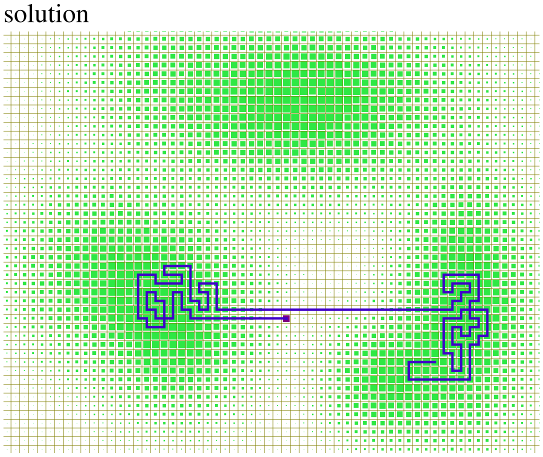

Fig. 9: Test 1 with constraint $L = 3000$, lhc-gw-conv

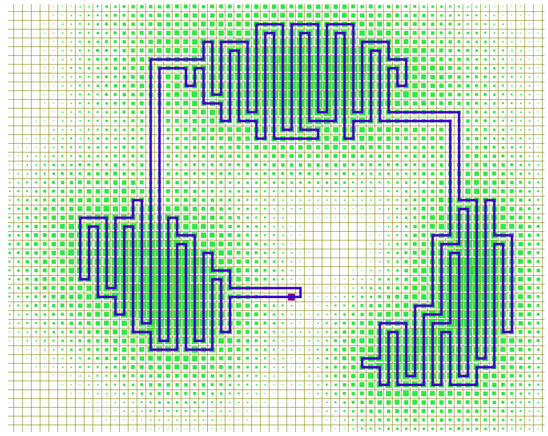

Fig. 10: Test 1 with constraint $L = 15000$, proposed solution

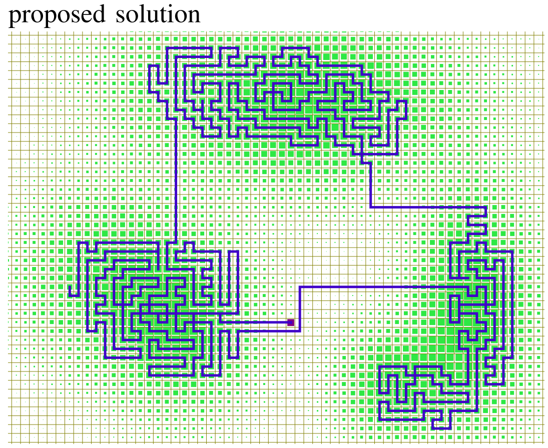

Fig. 11: Test 1 with constraint $L = 15000$, lhc-gw-conv

VI. Conclusion

The proposed solution has following next advantages:

- The idea simplifies the problem of building a path over an area to a problem of choosing several clusters (number of clusters is far less than the number of cells in the area);
- The solution can not visit the same cell twice because we have a hamiltonian cycle;
- The solution does not miss probabilities because an area can grow in all directions (not only 4 adjacent cells may be added);
- As a result, it has better efficiency (according to the tests);
- It always returns a drone to the starting position which is good when the constraint on the length of the route is due to limited battery charge of a UAV.

We compared the solution with other solutions in simulated scenarios to show its high efficiency. The ideas from the paper may be also used in other search problems to simplify them and get more practical solutions. In future work this approach can be improved by considering the number of turns in the path (which should be also minimized).

Acknowledgements

This paper was done under research work in Simlabs Inc. (sim-labs.com)

References

[1] K E. Trummel and J R. Weisinger, "The complexity of the optimal search path problem," *Operations Research*, vol. 34, pp. 324–327, Apr. 1986.

[2] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, vol. 72, pp. 411–458, Mar. 2018.

[3] L. Lin and M. Goodrich, "Uav intelligent path planning for wilderness search and rescue," Dec. 2009, pp. 709–714.

[4] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, pp. 1–10, Feb. 2011.

[5] P. Sokkappa and S. U. D. of Operations Research, *The Cost-constrained Traveling Salesman Problem*. Stanford University, 1990. [Online]. Available: https://books.google.ru/books?id=jax3JAAACAAJ.

[6] H. Oh, S. Kim, A. Tsourdos, and B. White, "Coordinated road-network search route planning by a team of uavs," *International Journal of Systems Science*, pp. 1–16, Dec. 2012.

[7] M. Tasgetiren and A. Smith, "A genetic algorithm for the orienteering problem," vol. 2, Feb. 2000, pp. 910–915.

[8] Y.-C. Liang and A. Smith, "An ant colony approach to the orienteering problem," *Journal of The Chinese Institute of Industrial Engineers*, vol. 23, pp. 403–414, Jan. 2006.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005, ISBN: 0262201623.

[10] P. Lanillos, S. K. Gan, E. Besada, G. Pajares, and S. Sukkarieh, "Multi-uav target search using decentralized gradient-based negotiation with expected observation," *Information Sciences*, vol. 282, pp. 92 –110, Oct. 2014.

[11] P. Yao, W. Honglun, and H. Ji, "Gaussian mixture model and receding horizon control for multiple uav search in complex environment," *Nonlinear Dynamics*, vol. 88, pp. 903919, Jan. 2017.

[12] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 77–98, Feb. 2001.