Chapter 1
Toward Computing an Optimal Trajectory for an Environment-Oriented Unmanned Aerial Vehicle (UAV) under Uncertainty

Jerald Brady, Octavio Lerma, Vladik Kreinovich, and Craig Tweedie

Abstract Over the past decade a few but increasing number of researchers have begun using Unmanned Aerial Vehicles (UAVs) to expand and improve upon existing remote sensing capabilities in the Arctic. Due to the limited flight time, it is important to make sure that the UAV follows an optimal trajectory – in which it cover all the points from a given area within the smallest possible trajectory length. Under the usual assumptions that we cover a rectangular area and that each on-board sensor covers all the points with a given radius r, we describe the optimal trajectory. A more complex optimal trajectory is also developed for the situations in which we need to get a more spatially detailed picture of some sub-regions of interest (in which we should have a smaller value r) and it is sufficient to get a less detailed picture (with larger r) in other sub-regions. We also describe the best ways to cover the trajectory in situations in which an UAV missed a spot – due to excess wind or to an inexact control.

1.1 Introduction

Need for environment-oriented UAVs. Arctic observing systems need to be enhanced with improved remote sensing technologies and capabilities – particularly mid-altitude remote sensing using air-borne platforms; see, e.g., [9]. Over the past decade a few but increasing number of researchers have begun using Unmanned Aerial Vehicles (UAVs) to expand and improve upon existing remote sensing capabilities in the Arctic.

Jerald Brady · Octavio Lerma · Vladik Kreinovich · Craig Tweedie
Cyber-ShARE Center, University of Texas at El Paso, El Paso, TX 79968, USA, e-mails jerald.brady@gmail.com, lolerma@episd.org, vladik@utep.edu (contact email), ctweedie@utep.edu

Need for customizable UAVs. Typically UAVs tend to be designed for a specific task or area of operation and so Unmanned Aircraft Systems (UASs) are usually not easily customizable.

It is desirable to develop UASs that allow for customizable sensor packages, reliable communications between ground and aircraft, tools to optimize flight control, real time data processing, the ability to visually ascertaining the quantity of data while the UAV is air-borne, and the ability to launch and land safely in these remote regions.

Our system. We have developed a prototype software system that allows for the customization of UAVs. This software has enhanced communication between ground and the UAV, can synthesize near real time data acquired from sensors on-board, can log operation data during flights, can visually demonstrate the amount/quality of data for a sampling area. The software has been designed to benefit an existing NSF Arctic Observing Network project that will focus on the remote sensing of landscape-scale vegetation structure and function.

Our UAS includes a paraglider UAV that has a suite of sensors suitable for characterizing hyperspectral reflectance and other surface properties. This paraglider UAV allows low and slow flying, has a limited range but a relatively large (ca. 13 kg payload). Sensors on-board relay operational flight data (airspeed, ground speed, latitude, longitude, pitch, yaw, roll, and video) as well a series of customizable sensor packages. Additional sensors can be added to an on-board laptop or a CR1000 data logger; see Fig. 1.1.

Need for coverage. The purpose of the UAV measurements is to describe the values of the environment-related physical quantities such as temperature, humidity, etc., at all possible locations within the rectangular observation area.

Of course, this "all" cannot be understood literally: the observation area has infinitely many points, and it is not possible to measure the value of the quantity is all these points. From the practical viewpoint, it is not necessary to take the measurements in all infinitely many points: usually, we know that the values at nearby points are practically indistinguishable. Specifically, a user usually provides us with a threshold r_0 such that the values of the desired quantity at points P and P' of distance $d(P,P') \le r_0$ are indistinguishable. In this sense, to make sure that we know the value at each point within the observation area, we have to make measurements in such a way that every point from the rectangle is at a distance $\le r_0$ from some point at which a measurement was made – i.e., from one of the points on the UAV's trajectory.

Need to take uncertainty into account. In practice, it is not possible to maintain the exact trajectory of a UAV, we can only maintain the desired trajectory with a certain accuracy r_1. In view of this uncertainty, if we simply make sure that every point P in the area is at a distance $d(P,P') = r_0$ from some point P' on the desired trajectory, the actual trajectory point P'' may

Fig. 1.1 Our UAV in flight

be at a distance $d(P', P'') = r_1$ from P' and thus, at a distance

$$d(P, P'') = r_0 + r_1 > r_0$$

from P. Thus, to make sure that even with this uncertainty, we have the desired coverage (with a distance threshold r_0), we need to guarantee that every point P from the observation area is at such a distance $d(P, P')$ from some point P' from the trajectory that $d(P, P'') \le d(P, P') + d(P', P'') \le r_0$ even when the distance $d(P', P'')$ attains the largest possible value r_1. In other words, we need to make sure that $d(P, P') + r_1 \le r_0$ For this inequality to be satisfied, we must make sure that $d(P, P') \le r \stackrel{\text{def}}{=} r_0 - r_1$.

Thus, to provide the desired coverage under this uncertainty, we need to make sure that every point P from the observation area is at a distance $\le r$ from some trajectory point.

Need for trajectory optimization. Due to the limited flight time, it is important to make sure that the UAV follows an optimal trajectory – in which it cover all the points from a given area within the smallest possible flight time – i.e., at the smallest possible trajectory length. Such a trajectory is described in this paper.

Comment. Most of our results were first announced in [3, 4].

1.2 Towards an Optimal Trajectory

The problem: reminder. We operate under the usual assumptions that we cover a rectangular area and that each on-board sensor covers all the points with a given radius r (see discussion above).

We are looking for trajectories that provide the desired coverage of the area – i.e., for which every point from the area is located at a distance $\le r$ from some point on this trajectory. Our objective is to come up with the trajectory that is "optimal" in the sense that it is the shortest among the trajectories that provide the desired coverage.

Analysis of the problem. Each trajectory piece of length ΔL_i covers the area $A_i \approx 2r \cdot \Delta L_i$; see Fig. 1.2.

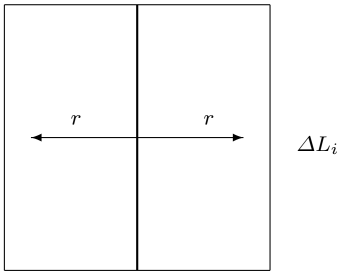

Fig. 1.2 Coverage

So, a trajectory of length $L = \sum\limits_i \Delta L_i$ covers the area

$$A \le \sum_i A_i = \sum_i (2r \cdot \Delta L_i) = 2r \cdot \sum_i \Delta L_i = 2r \cdot L.$$

Thus, to cover a region of area A_0, we need a trajectory of length $L \ge \dfrac{A_0}{2r}$.

Asymptotically optimal trajectory. The following natural trajectory is therefore asymptotically optimal, see Fig. 1.3. Indeed, in the region of area $A_0 = L_1 \cdot L_2$, we have $\dfrac{L_1}{2r}$ pieces of length $\approx L_2$ each. The total length is $L \approx \dfrac{L_1}{2r} \cdot L_2 = \dfrac{L_1 \cdot L_2}{2r} = \dfrac{A_0}{2r}$, i.e., this trajectory is (almost) optimal.

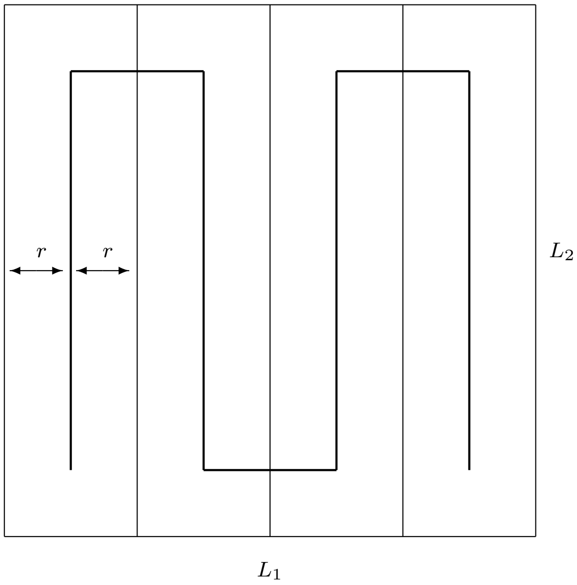

Fig. 1.3 An almost optimal trajectory

The above asymptotically optimal trajectory does not cover all the points. The minor problem with this trajectory is that the corner points (marked bold on Fig. 1.4) are not covered, because the distance from the trajectory to each corner point is $\sqrt{r^2 + r^2} = \sqrt{2} \cdot r > r$.

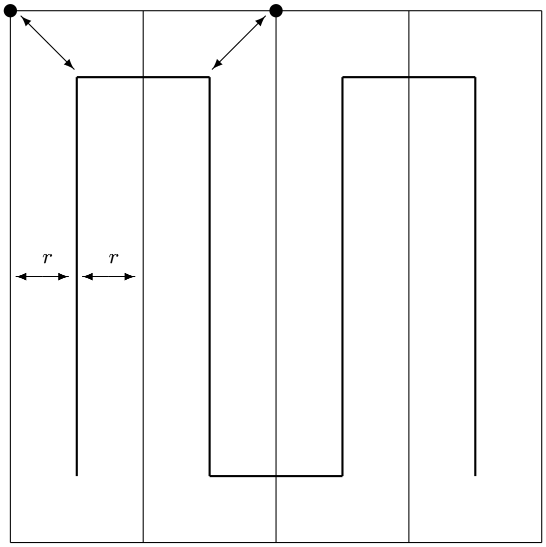

Fig. 1.4 For the asymptotically optimal trajectory, corner points are not covered

Precise formulation of our optimization problem. In this paper, we will consider trajectories which consist of two linear segments in each corner area, i.e., trajectories in which the original linear trajectory coming to the point P_1 is followed by two segments P_1P_2 and P_2P_3 that go into the next "corridor" of width $2r$; see Fig. 1.5.

Under this assumption, the question is how to select the points P_2 and P_3 in such a way that the total path $\ell = d(P_1,P_2)+d(P_2,P_3)$ is the shortest among all the paths that cover the whole corner area (i.e., for which every point from this area is at a distance of $\le r$ from some point on a trajectory).

Practical comment. The restriction to two-segment trajectories comes from *practice*: it is much easier to control the UAV along a linear trajectory, so adding too many segments would make the control difficult to execute in practice.

Mathematical comment. From the purely *mathematical* viewpoint, it may be interesting to analyze which trajectories are optimal among all possible trajectories – not necessarily the two-segment ones.

This problem is similar to the known Kolmogorov's definition of an ε-entropy of a set S (given originally in [6, 7]) as the smallest number of points for which every point in the set S is at a distance $\le \varepsilon$ from one of the selected points. What we are looking for can be viewed as a 1-D analogue of this notion: find the smallest length of a connected curve for which every point in the set S is at a distance $\le \varepsilon$ from one of the points on this curve.

Instead of considering all possible curves, we can also take into account the fact that sharp turns of a UAV are sometimes difficult to execute, and there is usually a bound on the curve's curvature. From this viewpoint, it may be interesting to consider curves whose curvature is bounded by a given value B.

An optimal trajectory: description. To cover the corner points like C, we propose a fin-like modification of the above trajectory; see Fig. 1.5. Specifically, after following the original trajectory up to the point P_1 which is located $2r$ units below the upper boundary, we then go straight to the point P_2 on the line CP' at a distance r from the corner point C. Then, we follow another straight line to the point P_3, etc.

Let us prove that this trajectory indeed covers all the points, and that, among trajectories that cover all the points, this fin-line trajectory is optimal (we will describe in what sense this trajectory is optimal). To illustrate this proof, we will use Fig. 1.6.

Proof that the new trajectory covers all the points. Let us first show that this trajectory indeed covers all the points. Indeed, in the given corridor of width $2r$, every point P below the line P_1P_0 is covered by the trajectory point which lies on the same horizontal line as P. Whether this point P is to the left or to the right of the trajectory, the distance is always $\le r$. A similar

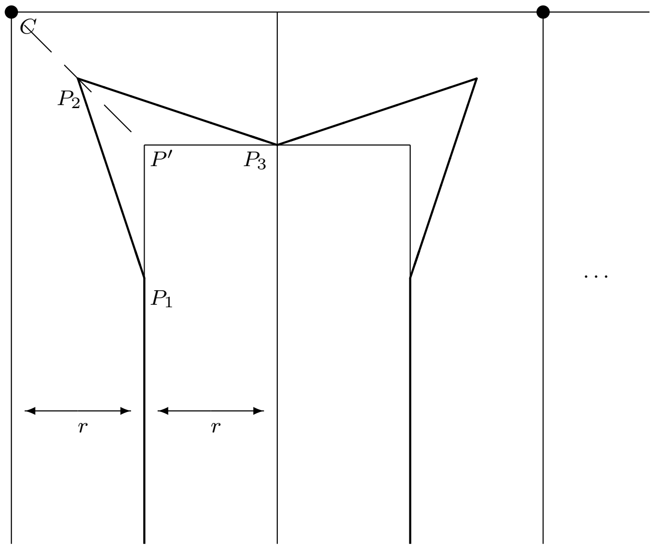

Fig. 1.5 An optimal trajectory

argument can be made about all the other points within this corridor, except for the points within the square $P_0P_1P'P_3$.

To prove the coverage for points from this square, we draw two lines through the point P_0:

- the line P_0P_1' parallel to the segment P_3P_2 of the trajectory, and
- the line P_0P_3' parallel to the segment P_1P_2 of the trajectory.

Both lines are at distance $\le r$ from the corresponding trajectory segments. Thus:

- all the points from the square $P_0P_1P'P_3$ which are above the new line P_0P_1' are covered, because they are at a distance $\le r$ from the segment P_3P_2;
- all the points from the square $P_0P_1P'P_3$ which are to the left of the new line P_0P_3' are covered, because they are at a distance $\le r$ from the segment P_1P_2.

Thus, all the points from the square $P_0P_1P'P_3$ are indeed covered. This completes the proof of coverage.

Proof that the new trajectory is indeed optimal. First, let us take into account that we need to cover points right above the point P_0 (which is $2r$ units below the upper boundary). If P_3 is less than r units away from the

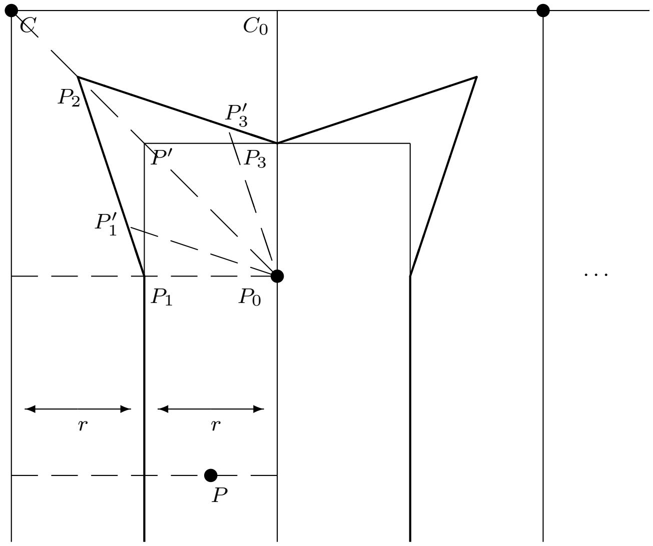

Fig. 1.6 Proof of coverage and optimality of the new trajectory

upper boundary, then these points are not covered. So, P_3 must be at least r units away from the upper boundary.

Similarly, to cover the point C_0 on the upper boundary, we need to make sure that the point P_3 is at a distance of at most r from the upper boundary. By combining these two conclusions, we thus deduce that the point P_3 must be at exactly r points from the upper boundary – i.e., at the same location as for the original trajectory. The only remaining question is where to place the turning point P_2.

For fixed points P_1 and P_3, and for a fixed total length $\ell = d(P_1, P_2) + d(P_2, P_3)$, the set of all the corresponding points P_2 forms an ellipse, with P_1 and P_3 as foci; see, e.g., [2, 5]. At least one of these points of this two-segment piece of the trajectory must cover the corner point C, i.e., it must be at a distance $\leq r$ from C. Thus, at least one point from this trajectory must be either within or at the border of the circle of radius r centered at the corner point C. Thus, the circle and the ellipse must intersect. If they are not tangent to each other at the intersection point, then we can decrease the value ℓ, and get a smaller ellipse which will still be intersecting. Thus, for the smallest value, the circle and the ellipse must be tangent to each other at the intersection point. One can show that under this condition, the point P_2 should be on the line PC – at a distance r from the corner point C, i.e., exactly where it is in our arrangement. The optimality is proven.

Comment. It is worth mentioning that the situation is symmetric with respect to reflection over the line $CP'P_0$: under this reflection, P_1 turns into P_3 and vice versa.

Practical comment: how to actually control the UAV. Once the optimal trajectory has been determined, the next question is how to control the UAV so that it follows this trajectory.

- For *manual* control, it is important to provide a good visualization of the past trajectory and of what it has already covered; such a visualization is described, e.g., in [8].
- For *automatic* control, control algorithms are described, e.g., in [1, 10].

1.3 What If We Want Different Coverage In Different Sub-Regions

Formulation of the problem. In some situations, we need to get a more spatially detailed picture of some sub-regions of interest (in which we should have a smaller value r), and it is sufficient to get a less detailed picture (with larger r) in other sub-regions.

Solution: main idea. In this case, it is reasonable to use optimal (or asymptotically optimal) arrangement in each sub-region.

Example: case of four sub-regions. For example, if we need four different values r_i in four different quarter-regions, then we should combine the corresponding optimal trajectories in four subregions as on Fig. 1.7.

In particular, if we use asymptotically optimal trajectory in each subregion, we get the following trajectory; see Fig. 1.8.

General case. The corresponding sub-division can be iterated if within each quarter-region, we have subregions with different desired coverage.

1.4 Tailwind Problem

Idealized case. In the above text, we assumed that a UAV follows the desired trajectory. In this case, we get a full coverage of the desired region.

Tailwind: a problem. In practice, an UAV can deviate from the planned trajectory. As a result, we may not cover some points in the region.

One reason why this may happen is tailwind. In the presence of a strong tailwind, the UAV flies too fast, there is not enough time for sensing.

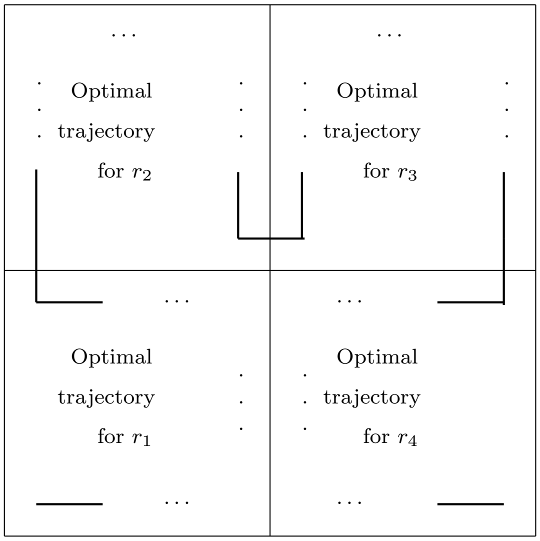
Optimal trajectory for r_2
Optimal trajectory for r_3
Optimal trajectory for r_1
Optimal trajectory for r_4

Fig. 1.7 Case when we need different coverage in different subregions

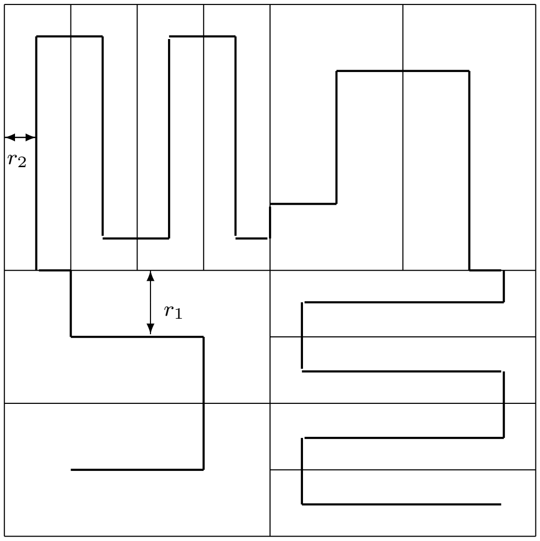

Fig. 1.8 Proof of coverage and optimality of the new trajectory: example

A solution to the tailwind problem. In the case of a tailwind, a natural solution is to change the direction of the trajectory, so that the wind would no longer be a tailwind.

This solution is illustrated below. In Fig. 1.9, we show the original plan. In Fig. 1.10, we show how this plan is disrupted by tailwind. In Fig. 1.11, we show how the change in direction changed the original trajectory.

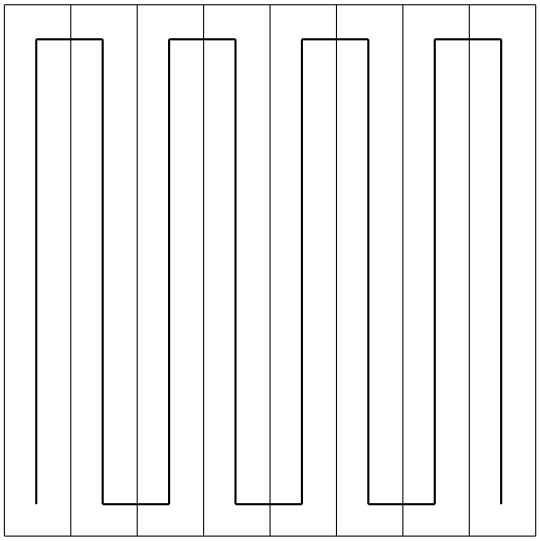

Fig. 1.9 Original plan

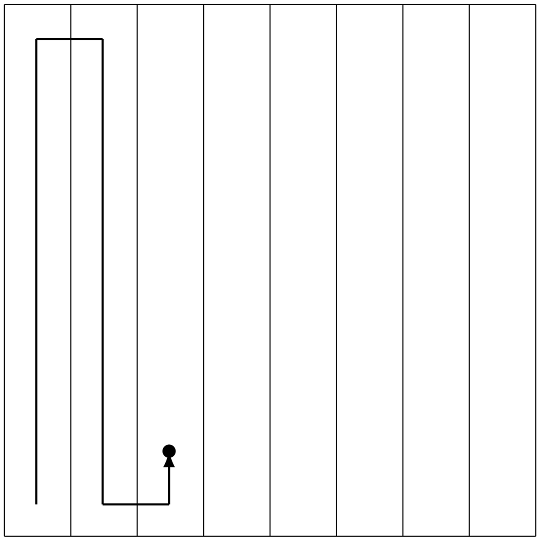

Fig. 1.10 Tailwind problem: plan disrupted by tailwind

1.5 Missed Spot Problem

Missing spot: formulation of the problem. In the ideal case, we should get a perfect coverage of the area; see Fig. 1.12. In practice, however, a sensor may malfunction when the UAV is flying over a certain area. In this case,

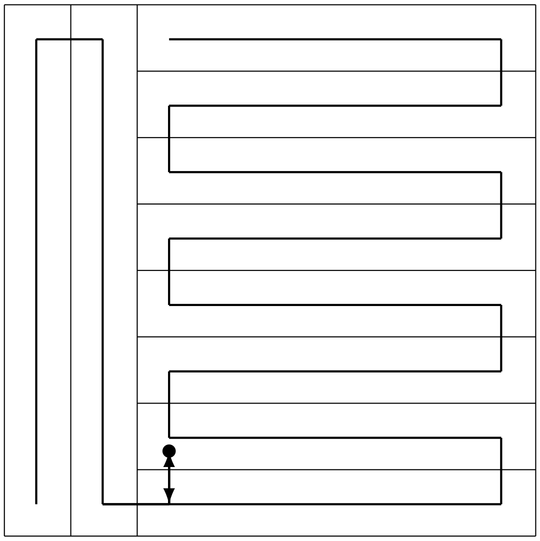

Fig. 1.11 Solution to the tailwind problem: change direction

while the trajectory is still covering the whole area, the measurement coverage misses a spot; see Fig. 1.13.

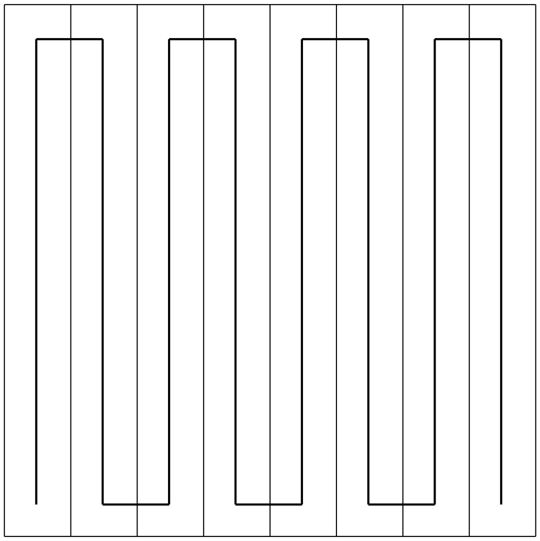

Fig. 1.12 Ideal case: perfect coverage of the area

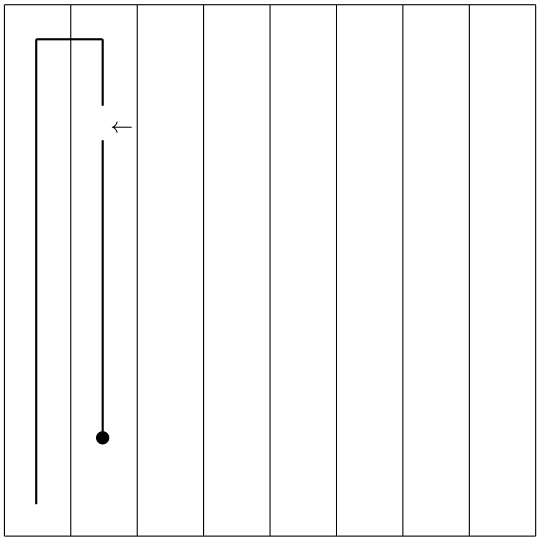

Fig. 1.13 missed Spot problem

Additional problem. The additional problem is that by the time we learn about the disruption, the plane has already moved along the planned trajectory; see Fig. 1.13.

A seemingly natural idea. In this case, if we left a missing spot, a natural idea is:

- to come back, to cover this spot, and then
- to continue along the original trajectory.

This idea is illustrated on Fig. 1.14

Limitation of this seemingly natural solution. The main disadvantage of the above (seemingly natural) solution is that we waste time by covering the same segment AB (see Fig. 1.14) three times:

- when we followed the original path,
- when we go back, from the point A t which we realized that we missed the point, to the point B that we missed; and
- when we go back, from the point B, to the point A, to resume the original trajectory.

A better idea: repair the spot on the next iteration. A better idea – an idea that avoids the above-mentioned waste – is to continue and to re-visit the missed spot on the next iteration; see Fig. 1.15.

Acknowledgements This work was supported in part by NSF grants: Cyber-ShARE Center of Excellence (HRD-0734825) and Computing Alliance of Hispanic-Serving Institutions CAHSI (CNS-0540592), and by NIH Grant 1 T36 GM078000-01.

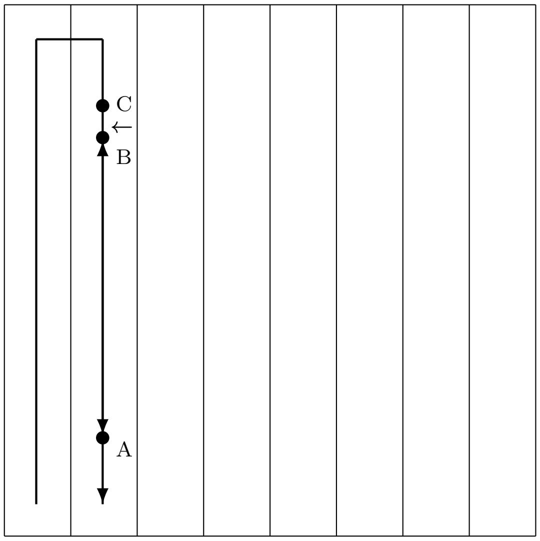

Fig. 1.14 Missed spot problem, seemingly natural idea: come back, then continue

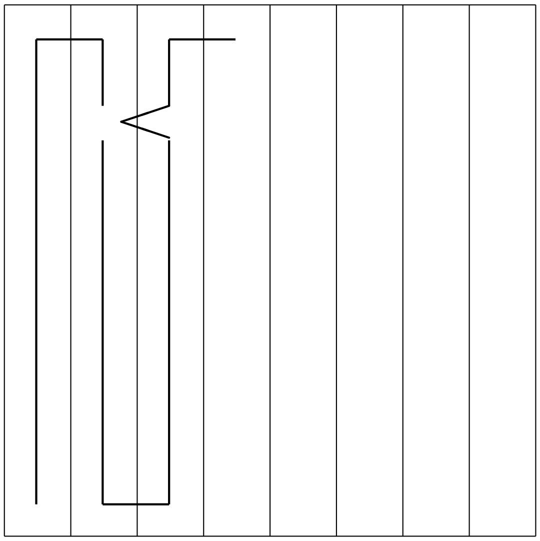

Fig. 1.15 New idea: repair the spot on the next iteration

References

1. Allaire F.C.J. et al.: FPGA implementation of genetic algorithm for UAV real-time path planning. J. Intell. Robot Syst. **54**, 495–510 (2009)
2. Berger M.: Geometry II. Springer, Berlin, Heidelberg, New York (1987)
3. Brady J. et al.: Toward computing an optimal trajectory for an environment-oriented Unmanned Aerial Vehicle (UAV). In: Abstracts of the 7th Joint UTEP/NMSU Work-

shop on Mathematics, Computer Science, and Computational Sciences, Las Cruces, New Mexico, April 3, 2010.

4. Brady J. et al.: Toward computing an optimal trajectory for an environment-oriented Unmanned Aerial Vehicle (UAV). In: Extended Abstracts of the NSF Joint Annual Meeting of the Centers of Research Excellence in Science and Technology (CREST) Historically Black Colleges and Universities – Research Infrastructure for Science and Engineering (HBCU-RISE) CREST/HBCU-RISE-JAM'2010, Washington, DC, June 6–9, 2010, pp. 69–70.
5. Coxeter H.S.M.: Introduction to Geometry. Wiley, New York (1989)
6. Kolmogorov A.N.: Theory of transmission of information. Amer. Math. Soc. Translation, Ser. 2 **33**, 291–321 (1963)
7. Kolmogorov A.N., Tikhomirov V.M.: ε-entropy and ε-capacity of sets in function space. Amer. Math. Soc. Translation, Ser. 2 **17**, 277–364 (1961)
8. Srikanth, M.B. et al.: Visibility volumes for interactive path optimization. Visual Comput. **24** 635–647 (2008)
9. Stephens, G.L. et al.: The Department of Energy's Atmospheric Radiation Measurement (ARM) Unmanned Aerospace Vehicle (UAV) Program. Bull. Amer. Meterol. Soc. **81**, 2915–2937 (2000)
10. Yang, K., Gan, S.K., Sukkareih, S.: An efficient path planning and control algorithm for RUAV's in unknown and cluttered environment. J. Intell. Robot Syst. **57**, 101–122 (2010)