Exploring the Tradeoffs Between Systematic and Random Exploration in Mobile Sensors

Samuel Matloob
University of Central Florida
Orlando, USA
samuel.matloob@ucf.edu

Ayan Dutta
University of North Florida
Jacksonville, USA
a.dutta@unf.edu

Patrick Kreidl
University of North Florida
Jacksonville, Florida
patrick.kreidl@unf.edu

Damla Turgut
University of Central Florida
Orlando, USA
turgut@cs.ucf.edu

Ladislau Bölöni
University of Central Florida
Orlando, USA
Ladislau.Boloni@ucf.edu

ABSTRACT

The movement of a mobile sensor has a critical impact on the information gathered from the area of interest, as well as the quality of the estimate that a model can build from the collected information at any moment in time. Both systematic exploration models, which make the sensor move in regular patterns, and random movement models have specific advantages. There is less research concerning models that are positioned between these two extremes. In this paper, we propose Grid Limited Randomness (GLR), a family of path planning algorithms based on sampling waypoints from a grid of a specific resolution. We propose three variations differentiated by the order in which the mobile sensor visits these waypoints: new samples added to the end of the path (GLR-EOP), smallest detour (GLR-SD), and the shortest path as approximated by Christofides' algorithm. An extensive simulation study in the Waterberry Farms benchmark shows that the GLR variations offer benefits that, in specific circumstances, make them preferable to both fully random and fully systematic exploration paths.

ACM Reference Format:
Samuel Matloob, Ayan Dutta, Patrick Kreidl, Damla Turgut, and Ladislau Bölöni. 2023. Exploring the Tradeoffs Between Systematic and Random Exploration in Mobile Sensors. In *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '23), October 30-November 3, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3616388.3617524

1 INTRODUCTION

Informative path planning (IPP) aims to determine a path for a mobile sensor (for instance, a drone) that, within specified constraints, acquires the highest quality information about the geographical area of interest [14]. The primary constraint in these scenarios is a limited time, movement, or sensing budget. This means, in practice, that the mobile sensor cannot visit every location in the geographic area: for instance, a drone in a precision agriculture application checking for disease outbreaks cannot inspect in detail every single plant. Thus, the result of the sensing will be a set of observations (samples) determined by the path of the mobile sensor. An *estimation algorithm* can be used to build a full map of the area of interest. As the mobile sensor continues to move and collect observations, the estimator can obtain maps of increasing quality.

IPP algorithms range from systematic exploration techniques that maximize coverage, to techniques that aim to serendipitously discover phenomena of interest. For instance, lawnmower-type algorithms achieve a predictable coverage of the area of interest, with the observations collected from equidistant paths (Figure 1-Top Left). Given an exploration budget, a fixed-budget lawnmower (FBLM) path planner would find the densest possible pattern that would still allow the sensor to return to its home location. Such a path is uniform and predictable: no point in the area of interest will be farther from an observation than one-half of the distance between turns. However, the cost of this uniformity at the completion is the lack of uniformity *during* the exploration: halfway through the path the mobile sensor did not yet reach the top half of the area. Any kind of transient phenomena of interest in this area will not be discovered during the exploration. A second challenge is that any kind of delay in the exploration, forcing the mobile sensor to return to the base before the completion of the trajectory, would leave the top part of the area unexplored. In adversarial environments, such as intrusion detection, systematic exploration has the additional disadvantage that the movement of the sensor is predictable, and certain areas are consistently less explored than others.

The path in Figure 1 (top right) shows a random waypoint (RW) model. The advantage of this model is that the exploration is not limited to well-defined areas at the beginning of the exploration but explores larger portions of the area of interest. Typically, a high-quality estimator can provide a better estimate from the samples collected by a random waypoint explorer in the initial parts of the exploration. Furthermore, the randomness of the exploration makes it difficult for adversarial entities to anticipate the mobile sensor.

However, random waypoint models also have specific disadvantages. Due to the geometry of the model, the center of the area of interest tends to be explored more thoroughly than the outlying areas [2]. These models also often lead to large areas that are left unexplored, usually near the borders of the area of interest, as shown in the top right side of Figure 1.

MSWiM '23, October 30-November 3, 2023, Montreal, QC, Canada

ACM ISBN 979-8-4007-0366-9/23/10…$15.00
https://doi.org/10.1145/3616388.3617524

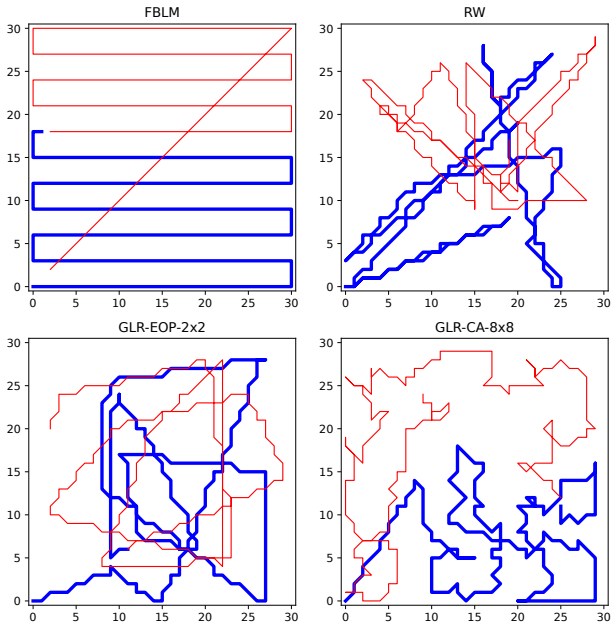

Figure 1: Four exploration techniques: Top-Left - Systematic / Lawnmower, Top-Right: Random waypoint, Bottom Left and Right: techniques trading off between randomness and systematic exploration. Thick blue lines show the trajectory in the first half of the exploration, thin red lines in the second half.

Systematic exploration and random waypoints are the two extremes of a continuum of possibilities between regularity and randomness. In this paper, we start from the conjecture that between these extremes there are approaches with features that, at least in certain circumstances, make them preferable to both the fully random and fully systematic approaches. We propose and investigate a family of exploration algorithms where the randomness of the waypoints is limited by reducing the random choice to happen within a cell of a grid. Examples of the paths generated by such an in-between approach are in Figure 1, bottom left and right. Within this algorithm family, Grid Limited Randomness (GRL), we also consider the order in which the waypoints are visited, leading to variations that affect not only the path of the robot but also the number of waypoints that can be chosen for a given budget.

The main contributions of this paper are:

- Proposes a family of exploration techniques, Grid Limited Randomness, that fill in the design space between random and systematic exploration.
- Provides practical implementations for three variations of the algorithms, based on the way new samples are appended to the path: End-of-Path, Smallest Detour, and Christofides' Algorithm.
- Through an extensive simulation study in the Waterberry Farms benchmark environment, we identify the settings in which specific variants of GRL outperform both fully random and fully systematic exploration.

2 RELATED WORK

The significant technological developments in hardware design made drones and mobile robots inexpensive and ubiquitous. While most such systems are currently remote controlled, their use as autonomous mobile sensors is a natural next step. This led to a new emphasis in the research of informative path planning algorithms.

The estimation model assumed by the algorithm can have a significant influence over the formulation of the problem.

For instance, Guestrin, Krause, and Singh [7] assume a Gaussian process estimator, which leads to the selection of observation points as the problem to maximize mutual information (a problem that is NP-complete by itself, even without considering the search for an efficient path through these points). A recursive greedy algorithm for a graph theoretical formulation was proposed by Chekuri and Pál [4]. Singh [13] proposed an innovative non-myopic method, which means the algorithm plans for possible future observations. The algorithm is an iterative one that calculates the informative path in each step to include the observation in the previous step, considering the budget and the path's cost. It also involves grouping random locations into clusters to sub-modularize the problem. Other recent path planning algorithms that assume an Gaussian process estimator include [8, 10, 12].

In general, planning for the quality of the information which might be obtained by the estimator is a very difficult problem. Thus many path planning algorithms instead optimize for a surrogate criteria, which is very often the coverage obtained by mobile sensor. Unfortunately, even the coverage problem [5], sometimes referred to as the orienteering problem [3] is difficult, because closely related to the traveling salesman problem and, as such, NP-complete. Often, practitioners accept a suboptimal algorithm from the point of view of path length, which however, solves practical problems with respect to robot control, avoidance of obstacles and so on. A survey of common coverage path planning algorithms and their two and three-dimensional applications are provided by Galceran and Carreras [6]. In certain situations, the path must be formulated to account for the topology of geographical area under consideration, which raises challenges even for the basic lawnmower pattern [1].

Waterberry Farm Benchmark [9] is a computational testbed designed to measure the performance of path planning algorithms. The model assumes a precision agriculture application, with environments of increasing size and complexity, planted with tomato and strawberry and various areas. The observations are used to create information models about two plant diseases, the Tomato Yellow Leaf Curl Virus TYLCV on tomatoes and Charcoal Rot CCR in the strawberry plants, as well as soil humidity.

3 GRID LIMITED RANDOMNESS

In order to develop exploration algorithms positioned between the fully systematic and fully random techniques, we could start from two directions: randomizing a systematic model or systematizing a random one. An example of the first approach would be to add random noise to the lawnmower movement. This would solve the problem of predictability of the exploration. However, it would introduce holes in the exploration, delay the robot due to the suboptimal tracks which would make it impossible to finish the exploration at the same exploration density and would still not help

with the fact that the mobile sensor explores only a geographically limited area in the first half of the exploration.

In this paper we take the opposite approach, and start from the random exploration technique, which we will make more systematic. Our process will start by investigating what is wrong with the traditional random waypoint (RW) model. In this model, effectively, the robot visits a series of waypoints that are chosen uniformly randomly from the area of interest. Uniform random sampling is a powerful and efficient way to acquire information from a studied phenomena, with significant theoretical apparatus behind it.

What are the problems with random waypoints? First, if the number of samples is small, it is still possible that the particular set of samples leave a relatively large area uncovered. This becomes more and more unlikely as the number of samples increases. The second problem is that while the waypoints are sampled uniformly randomly in RW, the observations are not. The mobile sensor needs to continue to make observations as it is moving from waypoint to waypoint and there are several mechanisms through which this leads to a suboptimal distribution of them. First, due to geometric reasons, most of the observations will cluster into the center of the explored area. Another problem is that in the random waypoint model, there are many self-intersections on the trajectory of the mobile sensor. Each of these points corresponds to a repeated, and therefore, likely a less informative observation.

This analysis leads us to two ideas in the design of the Grid Limited Randomness model. First, we avoid the holes in the sample set by overlaying a grid on the geometric area and sequentially sampling from the individual grid cells. This ensures that there is at most a difference of one sample between the most sampled and the least sampled grid cell.

The second idea is to influence the collected observations by choosing the order in which the waypoints are traversed by the mobile sensor. To understand the intuition here, let us consider a sensor moving between two waypoints. If these two waypoints are far away from each other, the robot will have to traverse a long straight path, during which it will make observations whose location is far from the waypoints (for instance, traverse other grid cells), and might have been previously explored. Furthermore, this long traversal will take resources away from the exploration budget of the sensor, which limits our ability to choose more waypoints. On the other hand, if the waypoints are close to each other, the observations on the paths between them are well correlated with the waypoints, and their impact on the exploration cost is significantly smaller. Thus, the second idea leads us to the conclusion that the order in which the waypoints are visited is an important component of the algorithm, and the GRL algorithm's grid sampling model needs to be combined with specific algorithms for this.

Before we proceed with the specific discussions of the models, we need to make another observations, that the planning of the exploration path often depends on the exploration budget. This budget reflects constraints such as the energy available to the mobile sensor (such as drone), financial constraints, or available daylight. For the remainder of this paper, we will express the exploration budget in the form of a total distance the mobile sensor can traverse while performing observations. The exploration budget has little impact on the RW exploration: the mobile sensor will simply keep exploring with the random waypoints until it runs out of the exploration budget. In contrast, for a systematic exploration model, the budget determines how densely the exploration turns can run in such a way as to complete the exploration within the budget. For the GLR model, the implication of the exploration budget is that it is impossible to know ahead how many grid-waypoints we can sample, as the length of the path connecting them depends on the waypoint ordering component of the path planning algorithm. Thus, most variants of the GLR model will need to find a way to simultaneously determine the number of waypoints and the path through them. We propose three waypoint ordering approaches of varying computational complexity.

End-of-Path (GLR-EOP): This approach collects the random samples from the grid by traversing the grid in a back and forth line by line pattern (effectively, a grid resolution version of the lawnmower). If the grid was exhausted, the iteration over the grid cells starts from the beginning. The waypoints are added to the end of the current path, until the length of the path exceeds the exploration budget b (see Algorithm 1).

To analyze the complexity of the algorithm, we notice that successive waypoints are in successive grid cells, thus the number of waypoints will be approximately equal to the exploration budget b divided by the grid cell size. If the path is created by k complete iterations over all the grid cells plus possibly several other samples, this means that each grid cell will have k or $k + 1$ samples.

Algorithm 1: GLR-EOP

Input : starting-point, velocity, time, h_{cells}, v_{cells}
Output: path

b = velocity · time
Divide area into a grid of size $h_{cells} \times v_{cells}$
path = [starting-point]
repeat
    Choose cell c as the next cell from the grid
    Choose waypoint w randomly uniformly from c
    Append w to end of $path$
until *length(path)* < b;
return path

Shortest detour (GLR-SD): This approach collects the random samples in a similar order as the GLR-SD approach, with a different approach taken for the insertion of the new waypoint into the path. Adding a new waypoint to the path will normally lead to an increase of the lenght of the path, as the path will need to make a detour between two points, and the triangle inequality ensures that the new path will be longer. The GLR-SD approach finds the insertion point (other than the starting point, which cannot be changed) where the caused detour is the shortest.

To analyze the complexity of the algorithm, we notice that the insertion of the a waypoint has a cost proportional with the current lenght of the path. Thus, computational complexity of the algorithm grows quadratically in the length of the budget. In practice, the insertion is highly parallelizable and thus has a negligible computational cost for paths with several thousand waypoints. As the waypoints are chosen in the same order as in GLR-EOP and no waypoint is rejected, this algorithm also retains the property that

there is a difference of at most one sample between the most and the least sampled grid cell. However, it is not true any more that the successive waypoints are in different grid cells - with GLR-SD the successive waypoints will be typically close together. An implication of this is that the GLR-SD approach, for the same exploration budget will have a much larger number of waypoints compared to GLR-EOP.

Algorithm 2: GLR-SD

```
Input : starting-point, velocity, time, h_cells, v_cells
Output: path
b = velocity · time
Divide area into a grid of size h_cells × v_cells
path = [starting-point]
repeat
    Choose cell c as the next cell from the grid
    Choose waypoint w randomly uniformly from c
    d = ∞
    insert_at=-1
    for i from 1 to count(path do
        detour = dist(path[i-1], w) + dist(w, path[i]) -
          dist(path[i-1], dist(path[i])
        if detour < d then
            d = detour
            insert_at = i
        end
    end
    insert w into path at location insert_at
until length(path) < b;
return path
```

Using Christofides' Algorithm (GLR-CA): This approach also collects the waypoints from the grid in the same order as the other algorithms, and aims to find the shortest path that connects these waypoints. As this challenge is equivalent to the traveling salesman problem, the optimal solution cannot be found in tractable time. Our approach uses Christofides' algorithm, a polynomial time $O(n^3)$ algorithm that finds a solution that it at most 3/2 of the optimal length (and, in practice, most of the time a significantly better approximation).

The motivation for GLR-CA is that with this algorithm, we can add a significantly larger number of data samples, whose distribution and randomness we have more control over it through the GLR mechanism. The overall flow of the implementation, however, is slightly different from the GLR-EOP and GLR-SD algorithms. In order to find the approximation of the optimal path, we need to present the algorithm the full set of sample points. However, we do not know ahead of time the length of the path the algorithm will find. With a fixed exploration budget, the best exploration can be achieved when the budget is fully utilized. A possibility would be to present the algorithm with a larger number of points, which creates a path longer than the exploration budget, and then cut short the resulting path at the expiration of the exploration budget. This, however, would result in a suboptimal result. Let us imagine that we have a 2x2 grid, and present the algorithm a large set of nodes sampled through the iterative GLR process. If the resulting path is four times the size of the budget, and we cut it down to the budget size, it is very likely that the path which we output would only contain points that are closely clustered in the grid cell closest to the starting point.

In conclusion, we need to cut down the waypoints *before* they are presented to the algorithm, and ensure that all the waypoints presented to the algorithm will be part of the output. Luckily, the search for this number is made easier by the fact that the length of the path monotonically increases when new points are added to the existing ones. This allows us to use a binary search to find the subset of the GLR waypoints to be presented to Cristofides' algorithms (see Algorithm 3).

To analize the complexity of the algorithm, we notice that we will need to run the Cristofides' algorithm $2 \cdot log_2(l)$ times, where $l < b$ is the number of waypoints in the final path, while Christofides' algorithm has a cubic complexity in the lenght of the path.

Overall, we can state that GLR-EOP has a $O(l)$, GLR-SD an $O(l^2)$ and GLR-CA an $O(l^3 \log(l))$ complexity. This difference is compounded by the fact that the length of the output path l is largest for GLR-CA, followed by GLR-SD and GLR-EOP.

Algorithm 3: GLR-CA

```
Input : starting-point, velocity, time, h_cells, v_cells
Output: path
b = velocity · time
Divide area into a grid of size h_cells × v_cells
waypoints = [starting-point]
for i from 1 to max do
    Choose cell c as the next cell from the grid
    Choose waypoint w randomly uniformly from c
    waypoints.append(w)
end
# Fast increase to an upper limit of waypoints
upper = 1
repeat
    upper = 2 * upper
    path = Cristofides_Algorithm(waypoints[0:upper])
until length(path) < b;
# Binary search to find the exact number of waypoints
lower = upper / 2 while upper - 1 > lower do
    test = (upper + lower) / 2
    path = Cristofides_Algorithm(waypoints[0:])
    if length(path) > b then upper = test ;
    else lower = test ;
end
return path
```

4 EXPERIMENTAL STUDY

4.1 Experimental setup

In order to investigate the properties of the proposed algorithms, we run a series of experiments using the Waterberry Farms (WBF)

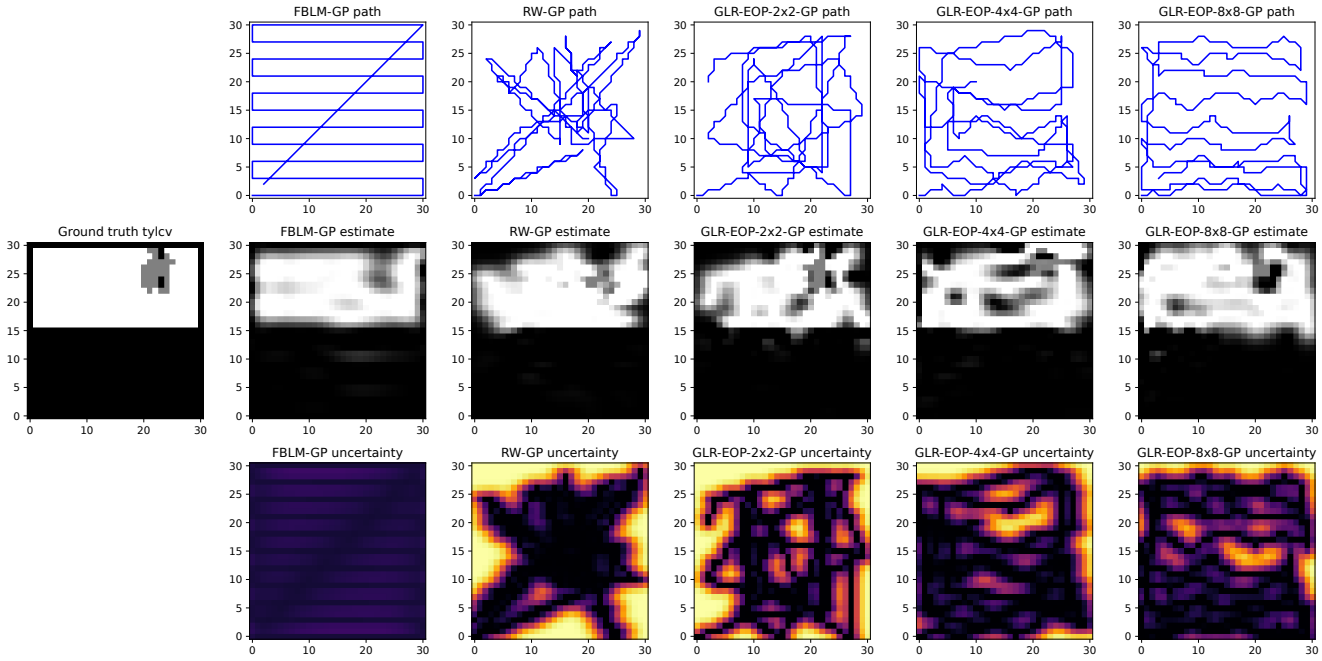

Figure 2: The path and end-of-experiment model performance of the GLR-EOP algorithm compared to FLDM and RW using the Gaussian Process estimator. Top row: the path of the mobile sensor. Center row: the ground truth for the TYLCV observations (leftmost figure), and the estimate provided by the estimator at the end of the exploration. Bottom row: the uncertainty metric of the estimator (the standard deviation of the Gaussian Process estimator at the specific point), at the end of the exploration.

benchmark environment. WBF realistically simulates the data collection in the context of a precision agriculture application, modeling a Florida farm growing tomatoes and strawberries. The environment creates models of the dynamic spreading of the tomato and strawberry diseases and the soil humidity. WBF allows the creation of experiments that benchmark pairs of path planning algorithms and estimators, and provides a number of scoring functions.

For the purposes of this paper, we used the following assumptions in the configuration of the WBF benchmark: we considered a medium size environment ("Miniberry-30", of size 30x30 ft), half planted with strawberry and half with tomato plants. We only considered the observations for the Tomato Yellow Leaf Curl Virus (TYLCV). We run a single-day experiment, which involved the day 10 of the model, where a relatively large infection of TYLCV exists in the environment. We assumed an exploration budget of 400seconds, with a velocity of the mobile sensor of 1ft/sec.

As we are comparing the path planning algorithms, we paired all algorithms we are considering with the same estimator, a high quality estimator using a Gaussian Process (GP) to perform reasoning about the observations collected. Each such GP derives its prior statistics under the same spatially-dependent structural assumptions, governed by a squared-exponential kernel function and fitting its hyper-parameters to past observations via standard regression techniques (e.g., via the `GaussianProcessorRegressor` within Python's `scikit-learn` toolbox, implementing Algorithm 2.1 assuming the kernel function in equation 2.31 of [11]).

The following algorithm were compared:

FBLM - Fixed Budget Lawnmower algorithm. This algorithm represents the "systematic" extreme of the randomness spectrum.

RW Random waypoint: the algorithm generates a series of waypoints chosen randomly and uniformly from the area of interest, and visits them in order. This algorithm represents the "random" extreme of the randomness spectrum.

GLR-EOP with grid sizes of 2-by-2, 4-by-4 and 8-by-8.

GLR-SD with grid sizes of 2-by-2, 4-by-4 and 8-by-8.

GRL-CA with grid sizes of 2-by-2, 4-by-4 and 8-by-8.

As the performance metric, we calculated a simple L1 score (mean absolute error) between the model created by the estimator and the ground truth for the TYLCV values.

For the purposes of real-world deployment, it is important to consider the computational costs of the various configurations. In our setting, the GLR-CA algorithm took about 5-10 seconds, while the GP estimator took about 20 seconds. The computational cost of the other algorithms was found to be negligible. While these values are acceptable for a practical deployment, we need to consider that the Miniberry-30 model is much smaller than a real farm. The Waterberry model built to match a real-world farm, has a size of 4000 by 5000 ft, representing a 10,000 times increase in size. As the GP algorithm has a cubic and the GLR-CA algorithm has a log-times-cubic complexity, these algorithms cannot be naively scaled to a real world setting.

4.2 Experimental results

Figure 2 shows the results of the experiments with the GLR-EOP model with various grid sizes, as well as the FBLM and RW algorithms. The top row shows the actual path traversed by the mobile sensor. We notice how the GLR-EOP variants on grid sizes 2x2, 4x2 and 8x8 represent a gradual transition between the RW randomness toward the systematic exploration of FBLM. Note that systematic exploration can be done in many different ways; the reason why the path for the GLR-EOP-8x8 resembles that of the lawnmower exploration is because of the way in which the random waypoints had been collected from the grid cells in a back and forth movement. These waypoints are not reordered in the EOP variant, retaining their original path.

The middle row of the figure shows, in the leftmost cell the ground truth of the observations. The bottom half of the area is not planted with tomatoes. In the part planted with tomatoes, there is an outbreak of the tomato yellow leaf curl virus (TYLCV), which endangers the crop. The reminder of the figures show the output of the estimators, at the end of the exploration path.

Some of the conclusions we can draw are as follows. The Gaussian Process estimator, in general, does a good job approximating the shape of the outbreak, if the number of observations around it are dense enough. When there are large areas where there are no observations, the GP can make significant errors, both in the positive and negative direction.

The bottom row of Figure 2 shows the uncertainty of the Gaussian Process estimator at the end of the exploration. Clearly, the systematic exploration algorithms leaves the least uncertainty in the exploration, *if it manages to finish the exploration.* The benefits of the GP model is evident from the fact that whenever the estimate has high errors (e.g. showing an outbreak where in reality there is none), it is actually also showing a high uncertainty. This is clearly visible, for instance, in the GLR-EOP-4x4 setting.

Figure 3 shows the same arrangement as Figure 3, but this time for the GLR-SD algorithm. In the following, we will focus on the specific differences between these algorithms. First, the GLR-SD algorithms still represent an intermediate level between the systematic exploration of FBML and the randomness of RW. However, as the shortest-detour algorithm does not insert the waypoints in the path in the order they are presented, the lawnmower pattern is not observable anymore in the path.

The SD algorithm greatly reduces the number of places where the path self-intersects. While the RW algorithm generates paths that self-intersect very often, and many intersections are present for GLR-EOP as well, the number of interactions for GLR-SD are very small. The algorithm also accomodates a larger number of waypoints, as shown by the fact that the GLR-SD paths are created from a larger number of short segments, in contrast with the longer segments in GLR-EOP.

Another interesting observation refers to the size of the unexplored "holes" in the path of the mobile sensor. The GLR procedure guarantees that there will be an approximately equal number of waypoints in every grid cells, but it does not guarantee that there will be trajectory paths traversing the center of the cells. It is thus possible that for grid cells with a lower resolution we have relatively large holes in the center of the cells, while the waypoints are connected through paths that a passing close to the borders of the cells. Such holes are visible in the lower left part of the area for the GLR-SD-2x2 and GLR-SD-4x4 algorithms in Figure 3.

Finally, Figure 4 shows the same setting for the GLR-CA algorithms. The overall look of the paths shows an evolution from the GLP-SD algorithm, retaining the general properties. The number of points continued to increase, making the straight segments even shorter. The GLR-CA algorithm remains prone to having large holes in the centers of the grid cells, as it optimizes for paths lenght, not for coverage. As in the case of GLR-SD, this problem gets ameliorated with the increase of the grid resolution.

The discussion, up to this point, as reflected in Figures 2, 3 and 4, reflected the state of the system after the successful conclusion of the estimates within the exploration budget b. Under these assumptions, systematic exploration is always the optimal strategy.

There are, however, many reasons for considering the quality of the estimates at other points in the estimation lifecycle. For instance, the customer of the sensing might require *anytime information*, the ability to query the current model at any point during the exploration. In a practical setting, there might be many reasons for the system to finish the exploration early, for instance due to weather events, mechanical failures or the actions of adversaries.

It is therefore useful to study the evolution of the quality of the model as the exploration proceeds. Figure 5 shows the evolution of the L1 score (negative mean absolute error). In these figures, the 0 value is the best achievable. The exploration budget in these experiments was 400, which means that the values at t=400 are the ones corresponding to the models shown in Figures 2, 3 and 4. The three figures compare the differently parameterized GLR-EOP, GLR-SD and GLR-CA variants respectively to GBLM and RW. For easier readibility, the figures had been an exponential smoothing applied with a 0.66 smoothing parameter.

Overall, from these results we can draw the following conclusions. All graphs start from score -0.4, which is the error beetween the initial model assumption (no TYLCV outbreak) and the ground truth. Overall, for all the trajectories, there is an increasing tendency in the score, as the model gets better with the additional observations. However, this increase is not monotonic, as the GP model might make errors in its reasoning. As the case of FBLM lines show, the score can actually dip below the one on the initial assumptions, if the model predicts disease in areas where it does not exist, while no disease in areas where it does. In the long run, however, all the trajectories lead to increasingly better scores.

Comparing the different approaches, we find that, as expected, the systematic exploration in FBLM ends up being either the best or very close to it. If we are considering the accuracy in the model earlier in the exploration, this is not the case in this scenario. In fact, FBLM overtakes the random waypoint RW only at around timepoint 375 out of 400. While this is due to some of the peculiarities of this scenario, in particular the fact that the TYLCV outbread happens to be relatively far from the origin, the fact remains that at halfway in the exploration budget, systematic approaches will have a hole that is of the size of half the explored area.

The GLR approaches, have a temporal evolution of performances which situate them between the early increase of RW and the slower but stead score increase of FBLM. Interestingly, for the GLR-EOP

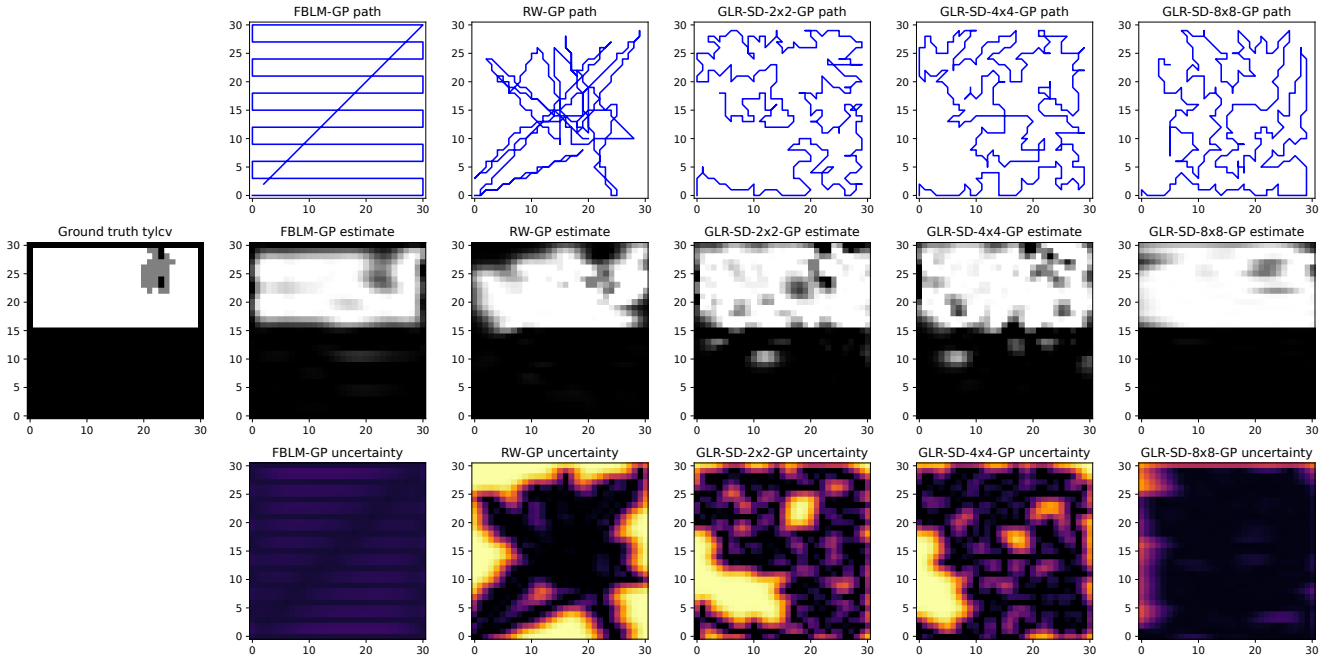

Figure 3: The path and end-of-experiment model performance of the GLR-SD algorithm compared to FBLM and RW using the Gaussian Process estimator. Top row: the path of the mobile sensor. Center row: the ground truth for the TYLCV observations (leftmost figure), and the estimate provided by the estimator at the end of the exploration. Bottom row: the uncertainty metric of the estimator (the standard deviation of the Gaussian Process estimator at the specific point), at the end of the exploration.

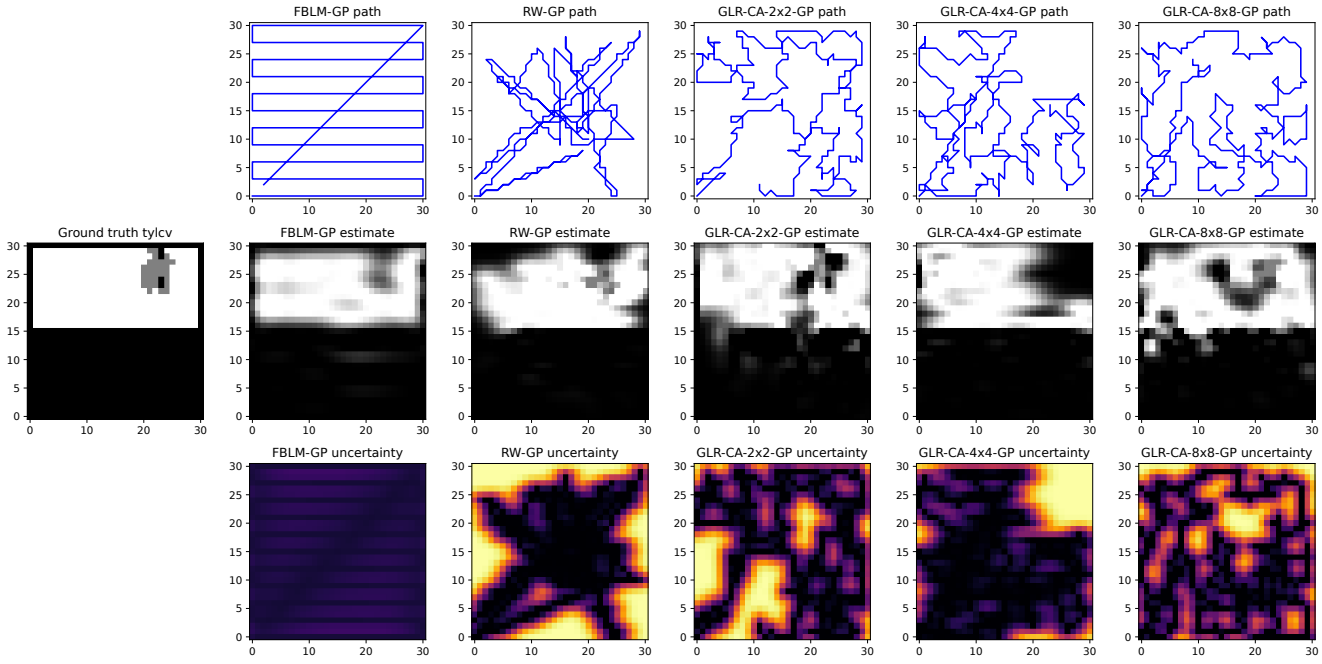

Figure 4: The path and end-of-experiment model performance of the GLR-CA algorithm compared to FBLM and RW using the Gaussian Process estimator.

approach, the coarser grids (2x2 and 4x4) perform better, while for the GLR-SD and GLR-CA approaches, the finer grids (8x8) is better.

As an overall evaluation, the best performance at the end of the exploration budget was achieved by the GLR-SD-8x8 variant, however the differences between most algorithms were sufficiently small to assign them primarily to random circumstances.

However, the differences were significantly larger earlier in the process. We can conclude that, if anytime performance is of importance, or if there is a significant risk of early termination of the exploration, algorithms from the GLR family can be a better alternative to both random waypoint and systematic exploration. From the various variants we explored, the most valuables were found GLR-EOP-2x2 and GLR-SD-8x8 (these grid sizes might need to be appropriately scaled function of the area size). In this series of experiments, we did not see an improvement from GLR-SD to GLR-CA which would justify the additional computational cost.

ACKNOWLEDGMENTS

This study is supported in part by NSF CPS grants #1932300 and #1931767.

REFERENCES

[1] T. M. Cabreira, L. B. Brisolara, and F. J. Paulo R. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4, 2019.
[2] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.
[3] C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms*, 8(3):1–27, 2012.
[4] C. Chekuri and M. Pal. A recursive greedy algorithm for walks in directed graphs. In *46th annual IEEE symposium on foundations of computer science (FOCS'05)*, pages 245–253. IEEE, 2005.
[5] H. Choset. Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.
[6] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.
[7] C. Guestrin, A. Krause, and A. P. Singh. Near-optimal sensor placements in Gaussian processes. In *Proc. of the 22nd Intl. Conf. on Machine Learning (ICML-2005)*, pages 265–272, 2005.
[8] K.-C. Ma, L. Liu, and G. S. Sukhatme. Informative planning and online learning with sparse Gaussian processes. In *IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2017.
[9] S. Matloob, P. P. Datta, O. P. Kreidl, A. Dutta, S. Roy, and L. Bölöni. Waterberry farms: A novel benchmark for informative path planning. *arXiv preprint arXiv:2305.06243*, 2023.
[10] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet. Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena. In *Proc. of the 2014 Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS-2014)*, pages 573–580, 2014.
[11] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.
[12] T. Said, J. Wolbert, S. Khodadadeh, A. Dutta, O. P. Kreidl, L. Bölöni, and S. Roy. Multi-robot information sampling using deep mean field reinforcement learning. In *Proc. of IEEE Conference on Systems, Man and Cybernetics (SMC 2021)*, pages 1215–1220, October 2021.
[13] A. Singh. Nonmyopic adaptive informative path planning for multiple robots. Technical report, UCLA, 2009.
[14] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.

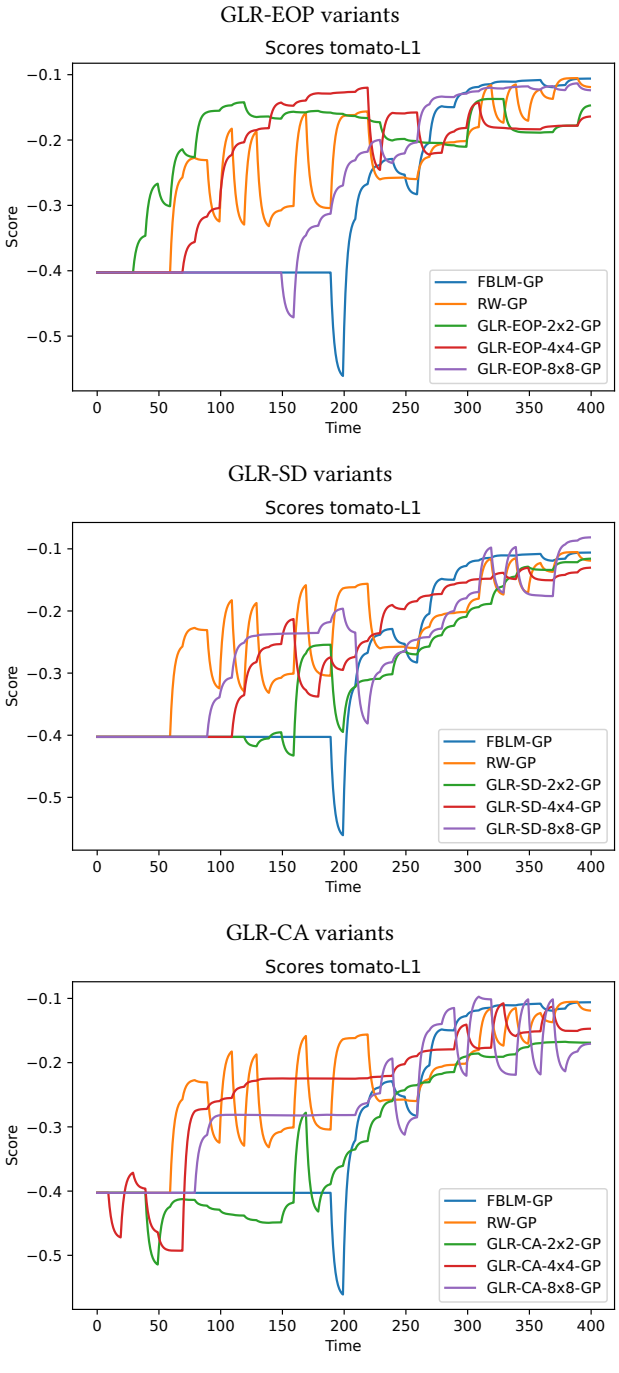

Figure 5: The evolution of the L1 score for the TYLVC detection.