Effective Searching for the Honeybee Queen in a Living Colony

Jan Blaha[1], Jan Mikula[3,1], Tomáš Vintr[2], Jiří Janota[1], Jiří Ulrich[1],
Tomáš Rouček[1], Fatemeh Rekabi-Bana[2], Laurenz Alexander Fedotoff[4], Martin Stefanec[4],
Thomas Schmickl[4], Farshad Arvin[2], Miroslav Kulich[3], and Tomáš Krajník[1]

Abstract— **Despite the importance of honeybees as pollinators for the entire ecosystem and their recent decline threatening agricultural production, the dynamics of the living colony are not well understood. In our EU H2020 RoboRoyale project, we aim to support the pollination activity of the honeybees through robots interacting with the core element of the honeybee colony, the honeybee queen. In order to achieve that, we need to understand how the honeybee queen behaves and interacts with the surrounding worker bees. To gather the necessary data, we observe the queen with a moving camera, and occasionally, we instruct the system to perform selective observations elsewhere. In this paper, we deal with the problem of searching for the honeybee queen inside a living colony. We demonstrate that combining spatio-temporal models of queen presence with efficient search methods significantly decreases the time required to find her. This will minimize the chance of missing interesting data on the infrequent behaviors or queen-worker interactions, leading to a better understanding of the queen's behavior over time. Moreover, a faster search for the queen allows the robot to leave her more frequently and gather more data in other areas of the honeybee colony.**

I. INTRODUCTION

The advances in mechatronics, control and AI resulted in robots capable to interact with animals in their natural ecosystems [1], [2]. These robots can be used to support nature conservation [3], monitoring [4]–[6], maintenance [3] and even recovery of damaged ecosystems [7]. Deployment of robots to support the ecosystems has the potential to mitigate the negative impacts of the ongoing ecological crisis [8], [9] on the environment and agricultural production.

In the European project "RoboRoyale" we work on establishing a biohybrid system where robots interact with a honeybee queen in order to support the whole colony. For this purpose, a robotic manipulator was designed, capable of detailed observations of the insides of the hive using a positionable camera. The system targets the observations of the honeybee queen, marked with a fiducial marker [10] similar to the one used in conventional beekeeping and the captured high-detail imagery is used to extract the queen's behaviors.

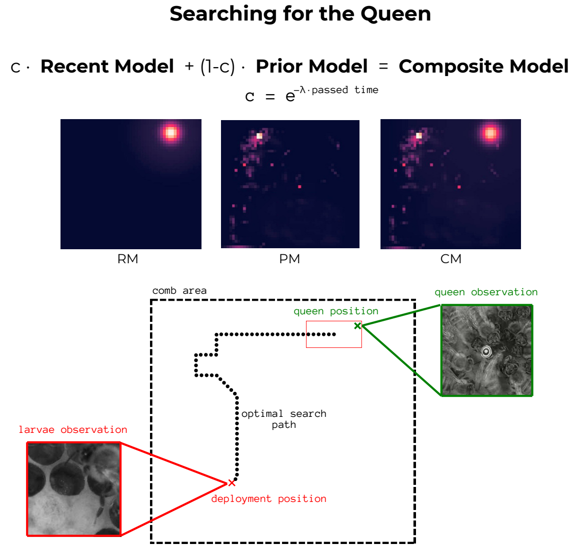

Fig. 1. In this work, we optimize the time it takes to locate the queen after deployment for observations in different parts of the comb. During regular operation of tracking the queen, the system is deployed to an arbitrary location in the hive to observe other phenomena like the larvae development. When the deployment is finished, a trajectory needs to be planned to localize the queen again. We use short-term information about the last known position ("recent model", RM) and long-term information from past observations ("prior model", PM) to inform a searching algorithm.

However, as the queen is not an isolated agent, the state of the whole colony, including the likes of nutrients, numbers of eggs, and larvae, need to be understood as well to interpret the data. Such observations are in the current version of the system achieved by scanning the whole comb several times per day and inspecting manually prespecified areas at given times. Both comb scanning and inspections interrupt the observation window and require a procedure to localize her in order to recover the tracking. Currently, sequential scanning of the whole comb is used.

In this paper, we investigate methods for more efficient localization of the honeybee queen. We aim to reduce the average time it takes to locate her by an intelligent search algorithm using a combination of long-term models of the queen's typical presence and short-term information of her last detection, as summarized in Fig. 1. We evaluate the proposed approach on a month-long dataset of queen trajectories in a living colony collected by the autonomous observation unit during the last season.

[1] Faculty of Electrical Engineering, Czech Technical University in Prague, Czechia `name.surname@fel.cvut.cz`
[2] Durham University, Computer Science Department, United Kingdom `name.surname@durham.ac.uk`
[3] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czechia `name.surname@cvut.cz`
[4] Artificial Life Lab, Department of Zoology, Institute of Biology, University of Graz; Graz, Austria `name.surname@uni-graz.at`

The code for the experiments in this paper is available at `https://gitlab.roboroyale.eu/paper-experiments/2024_case_searching`.

II. RELATED WORK

In order to formulate an intelligent algorithm for fast localization of the queen in the hive, we need to solve a general optimization problem of searching for randomly distributed targets. However, for such a solution to be useful in application, the algorithm needs to be provided with a good estimate of the target's distribution. It has been shown that models of environmental dynamics can approximate such distributions from past observations into the future and improve the performance of robots using them.

A. Search-Related Combinatorial Optimization

In later sections, we formulate the task of finding the optimal search path for locating the queen as the Graph Search Problem (GSP), which generalizes the well-studied Minimum Latency Problem (MLP) in the field of combinatorial optimization. Here, we provide background on the solution methods employed for these problems.

The MLP aims to find a tour such that the average visiting time of all vertices in the given complete graph is minimal. Although the problem is known to be NP-hard [11], several exact algorithms exist [12]–[14]. Additionally, integer linear programming with branch-and-cut and branch-cut-and-price approaches have been developed for the Time-Dependent Travelling Salesman Problem, a generalization of both the MLP and GSP [15]–[18]. However, these algorithms can only optimally solve instances with up to 100 vertices.

Metaheuristics, which provide good quality solutions in a reasonable time, appear to be more useful in practice. For example, Salehipour et al. employ the Greedy Randomized Adaptive Search Procedure (GRASP) for MLP and compare the impact of the VNS procedure as a local search phase with Variable Neighbourhood Descent (VND) [19]. Salehipour's results were improved by [20], who proposed a General Variable Neighborhood Search (GVNS). Even better results were reported by [21], who proposed the GILS-RVND, a simple multi-start heuristic combined with the Iterated Local Search (ILS) procedure. Recently, Santana et al. [22] improved GILS-RVND by means of data mining, and Mikula and Kulich [23] proposed a Multi-start GVNS (Ms-GVNS), which outperforms GILS-RVND within a strict computational time budget ranging from 1 to 100 seconds.

The GSP, first studied in [24], generalizes the MLP by assuming that each vertex has an assigned probability of finding the object when visiting the vertex, and these probabilities differ in general. Besides some theoretical results regarding approximation schemes presented in [25], no further developments are present in the related literature. The only exception is a tailored GRASP metaheuristic for the GSP introduced in [26] and its extension for multiple agents [27].

B. Spatio-Temporal Models

Maps of Dynamics (MoD) [28] are internal representations of dynamic patterns of a robot's operational environment that enable predicting the environmental state out of the robot's perceptual range [29]. Recent research showed the indispensability of incorporating MoD into the autonomous robots' task planning systems [30]–[33]. Although the problem of environment changes was typically connected with robot mapping and self-localisation [34], [35], MoDs were also successfully applied to a robotic search task [36]. However, the integration of MoD into the task planning system was almost exclusively applied to human-designed and human-populated environments [37]–[39].

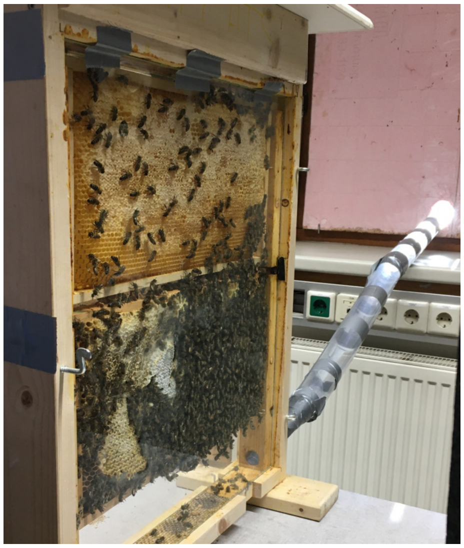

Fig. 2. Construction of an observation hive, used in biology to study honeybee behavior inside the colony, courtesy of [40]. Standardized combs behind a glass panel allow for observation of a living colony with minimal disturbance, typically conducted under deep red or near-infrared light. The hive is connected to the outside by a plastic tube entrance.

In this work, we exploit spatio-temporal MoD that generalise independent area-based observations to model only one specific insect entity customs. Moreover, an integration of MoD to enhance a decision-making process in the bio-hybrid systems was never studied. For a higher conclusiveness, we chose to evaluate a subset of simplified versions of previous successful predictive maps defined and evaluated in [32].

III. PROBLEM

A typical hive consists of standardized combs—we use $420\,\mathrm{mm} \times 220\,\mathrm{mm}$ dimensions—stacked on top of each other, with a glass panel on the side to allow observations inside a living colony, as shown in Figure 2. Despite being indoors, the hive is connected to the outside via a plastic tube to allow the bees to fly out.

In a previous version of an automated observational system [41], four static cameras were watching the entire hive, providing broad, low-detail observations. The system we use here, is explicitly designed for very detailed observations anywhere in the hive by a positionable camera. The camera provides $1920\,\mathrm{px} \times 1080\,\mathrm{px}$ images at $30\,\mathrm{Hz}$.

A. Scheduling Observations

The system is designed for the automation of long-term biological observation. While primarily the system follows the honeybee queen using the WhyCode fiducial marker [10] placed on the queen's abdomen, it is intended for complex data collection of the whole colony. Apart from simple scanning behavior, it is possible to manually schedule selective data collection by specifying the target position and duration of observation. This way, the system can produce, for example, timelapse videos of larvae development by collecting images in predefined intervals at locations preselected by a biologist. Due to the high level of detail, it is only possible to watch a small portion of the hive, which makes measurements of various phenomena exclusive, requiring an intelligent scheduling of the observations.

In the current setup, the implemented recovery behavior searching for the queen performs a sequential sweep of the entire comb. This is why it is currently inefficient to send the camera away to observe for a long time. After leaving the queen for a longer time, the queen could have moved substantially far from its last known position to return to it. Some portion of the time, the queen rests, so she does not move from her last position at all, but she is active and can travel far in the rest of the time. The longer observations are scheduled, the higher the need for efficient searching behavior.

B. Searching for the Queen

In this work, we formally define our task of searching for a queen after observation by a tuple $D = (t, \mathbf{p}_q, \mathbf{p}_d, d)$, where t is the time of deployment, $\mathbf{p}_q \in \mathbb{R}^2$ is the last known position of the queen (also the position where the camera starts its deployment), $\mathbf{p}_d \in \mathbb{R}^2$ is the target position of deployment for observation and d is its duration. Given D, we want to find a sequence of camera positions throughout the hive that lead to the fastest localization of the honeybee queen. This corresponds to the problem of effective target localization after scheduled observation.

We consider the manipulator to leave the queen at t, travel at its given speed v to $\mathbf{p}_d$, and spend there time d. At time $s = ||\mathbf{p}_q - \mathbf{p}_d||/v + d$, it starts searching for the queen and travels the hive for up to $10\,\mathrm{min}$ until the marker is well detectable in the image.

The search at time s can be informed by two sources of information, giving each potential location a weight representing the likelihood of finding the queen. The first is the recent knowledge of $\mathbf{p}_q$, the queen's location at time t, which we denote a recent model RM. The second source of information can be long-term observations of the queen's behavior, which we refer to as a prior model PM. These two sources are both relevant, but their relevance changes over time, and combining them requires considering the passed time $s - t$.

IV. METHODS

A. Informing the Search

A model to inform the search is a function $f(t, \mathbf{p})$, which gives the relative likelihood of the queen being at position p at the time t. We consider two distinct models and their combination in time. The first is the recent model RM, which represents a short-term memory of the last known observation. The second is the model of long-term dynamics, which we refer to as the prior model PM.

1) Recent Model: We define the recent model to be a probability density function (pdf) of the displacement of the queen for each value of passed time discretized to seconds,

$$RM(s, \mathbf{p}) = pdf_{(s-t)}(\mathbf{p} - \mathbf{p}_q), \tag{1}$$

where s is the time at the beginning of the search, $\mathbf{p}$ is the queried position, and $\mathbf{p}_q$ is the last known position of the queen from time t.

In our work, we use the empirical estimate of RM on the training data. For each second from $0\,\mathrm{min}$ to $10\,\mathrm{min}$, we compute the Gaussian kernel density estimate with bandwidth set to $0.02\,\mathrm{m}$. We show the fitted model in Fig. 4.

2) Prior Models: To model the long-term dynamics of the queen's location, we employ a selection of simplified MoD that were not developed for specific tasks or environments [32]. These spatio-temporal MoD cover discrete [42] (*HistDayGrid*), temporalily continous [43] (*FreMEnGrid*), spatially continuous [44] (*time_window_GMM*), and continuous [45] (*HyTted_GMM*, *WHyTened_kMeans*) generalizations of observed dynamics. Following the original paper, the names are usually composed of a prefix specifying the forecasting method and a postfix specifying the spatial structure. Chosen maps use three types of spatial structure—the *Grid*, where the spatial component is decomposed into rectangular grid cells, and the data-oriented spatial dividing, *kMeans* and *GMM*, which split the space according to a clustering of the observations. The forecasting methods are derived from history knowledge aggregations [42] (*Mean*, *HistDay*), spectral analysis of the time domain [43] (*FreMEn*), and warped hypertime [46] (*HyTted* and *WHyTened*). There are two exceptions: *HyT_X_GMM* that models temporal and spatial dynamics independently and forecasts their combination, and *GMM* that models only the spatial behaviour of the queen.

Upon learning, every MoD gives us a prediction function f of the likelihood of the queen's location for a given time t and positions $\mathbf{p}$. Given a MoD method and a discretized set of positions to evaluate the prediction for, we simply get

$$PM_{\text{method}}(s, \mathbf{p}) = \frac{f_{\text{method}}(s, \mathbf{p})}{\sum_{\mathbf{q}} f_{\text{method}}(s, \mathbf{q})}, \tag{2}$$

which are the MoD predictions normalized over all the available positions $\mathbf{q}$.

3) Composed Model: The short-term information from RM and long-term information from PM need to be combined with respect to the time since the last observation, as the RM deteriorates over time. Although for a different purpose, previous works in MoDs have considered the decay of short-term information to be exponential [47]. We also adopt this idea and for switching between the RM and PM we, therefore, employ an exponential function $c(\delta) = \exp(-\lambda\delta)$ weighting the passed time δ as

$$\alpha = c(s - t), \tag{3}$$

$$CM = \alpha \cdot RM + (1 - \alpha) \cdot PM. \tag{4}$$

For $(s-t) \to 0$, the mixing parameter $\alpha \to 1$, preferring the recent model RM, for $(s-t) \to \infty$, the mixing parameter $\alpha \to 0$, preferring the prior model PM. For better interpretability, we reparametrize the function c from λ to h according to $\lambda = -\log\left(\frac{1}{2}\right)/h$. We refer to h as "recency" or "recency half-life" as it allows us to specify the information decay by the time at which α reaches 0.5, and the contribution of both models is the same.

B. Optimizing the Search Path

We approach the task of finding the optimal search path for locating the honeybee queen by introducing the following simplifying assumptions: during the search period, the queen is considered static and located at one of a discrete set of possible locations of the camera. To reduce the continuous space of feasible camera positions to a graph, we adopt a simple discretization to a regular grid whose cells form the set of searched locations. Additionally, we assume that the discrete probability distribution of finding the queen over the set of locations is known. Then, the natural criterion is to minimize the expected time to find the queen while traveling along a path that visits (or 'searches') all these possible locations where the queen may be found. In general, minimizing the *expected* completion time of a task aligns with the search paradigm [48], in contrast to the inspection paradigm, which aims to minimize the *worst-case* completion time.

Based on the simplifying assumptions, we formulate the following optimization problem. We posit the existence of a static object of interest (the queen) located at an unknown location Q (a random variable) among a discrete set of locations $V = \{1, \dots, n\}$. The probability $P(Q = i)$ that the object is located at a particular location $i \in V$ is known and denoted as $p(i)$, where $\sum_i p(i) = 1$. Our goal is to minimize the expected time $\mathrm{E}[T|\pi]$ to find the object along a path $\pi : V \mapsto V$ (a permutation of V). Based on the definition of expectation, we can express this criterion as:

$$\mathrm{E}[T|\pi] = \sum_{i=1}^{n} p(\pi(i)) \cdot \delta_\pi(i), \tag{5}$$

where T is a random variable taking the value of time when the object is found, and $\delta_\pi(i)$ denotes the latency of arriving at $\pi(i)$, measured since initiating the search task. Given an additional, fixed starting location $\pi(0) = 0$ and a travel time $t(j,k)$ from location $j \in \{0\} \cup V$ to location $k \in V$, the latency $\delta_\pi(i)$ can be defined as follows:

$$\delta_\pi(i) = \sum_{l=1}^{i} t(\pi(l-1), \pi(l)). \tag{6}$$

Combining Eq. (5) and Eq. (6), the optimal search path $\pi^\star$ can be expressed as:

$$\pi^\star = \arg\min_\pi \sum_{i=1}^{n} p(\pi(i)) \sum_{l=1}^{i} t(\pi(l-1), \pi(l)). \tag{7}$$

Finally, since the optimization result remains unaffected by a constant factor that multiplies the optimization criterion, Eq. (7) can be equivalently written as:

$$\pi^\star = \arg\min_\pi \sum_{i=1}^{n} w(\pi(i)) \sum_{l=1}^{i} t(\pi(l-1), \pi(l)), \tag{8}$$

where $w(i) \propto p(i)$ represents an unnormalized probability mass of location i, also referred to as the *weight* of i. Eq. (8) precisely describes the GSP, which we provide background on in Sec. II-A.

In our original task of searching the honeybee queen, we adopt a metaheuristic approach to solve instances of the GSP. These instances are defined by the grid discretization of the comb area, with each grid cell with coordinates $\mathbf{q}$ representing a possible queen location i. The weights $w(i)$ assigned to these locations are determined by the respective model, which predicts the likelihood of the queen's presence, e.g. if location i lies at coordinates $\mathbf{q}$, the search at time t using prior model PM would set $w(i) = PM(t, \mathbf{q})$. We estimate the travel time between locations based on a constant-velocity assumption, calculated as the Euclidean distance between the centers of the cells. Lastly, the initial location $\pi(0)$ is set to location l that corresponds to the current position of the camera $\mathbf{q}_t$.

To solve the GSP instances, we employ the Ms-GVNS metaheuristic [23]. Ms-GVNS is a restarting method that, with each restart, generates a greedy solution. It then systematically switches between the VND local search method [49] to find local optima with respect to several neighborhood operators, and random perturbations with increasing intensity to escape these optima and diversify the search for an optimal solution. Specifically, it employs the standard improvement operators *2opt*, *insert*, *or-opt2*, *or-opt3*, and *or-opt4* in VND (in that order), and *4opt*, *8opt*, and *12opt* (in increasing intensity order) as random perturbations.

Ms-GVNS was originally designed for the MLP, which aligns with the GSP where $\forall i : w_i = 1$. In the experimental evaluation detailed in [23], it demonstrated exceptional performance in solution quality compared to the reference metaheuristic GILS-RVND [21] within constrained computational time limits (1 to 100 seconds) and for instances up to size 1000. This is particularly appealing in our context, where bigger time budgets for finding solutions are not anticipated. To use the Ms-GVNS in this work, we adapt it for the weighted version of the problem, defined as in Eq. (8), by deriving efficient $\mathcal{O}(1)$ improvement computations for the VND operators, similar to those described in [23].

V. EXPERIMENTS

To evaluate the ability of our proposed methods, we first experiment with the setting of parameters of the GSP solver. Then, we simulate the time it would take our methods to locate the queen on a set of simulated deployment tasks corresponding to possible scheduled observations We generate the testing scenarios from a real dataset from the testing setup collected during the last observational season, 2023.

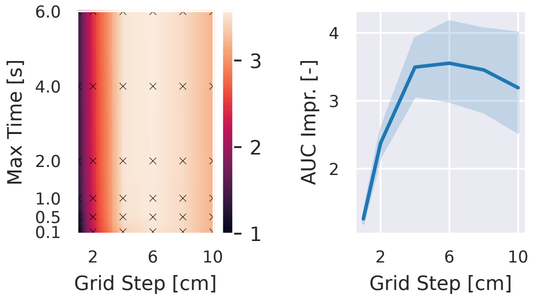

Fig. 3. The mean relative AUC performance in information gathering by the camera compared to discrete GSP solution for various settings for the Ms-GVNS solver. On the **left**, we show how much the camera would gather over setting the grid step to $1\,\mathrm{cm}$ and maximal runtime $0.1\,\mathrm{min}$. Markers represent the sampled parameter combinations; the rest of the figure is interpolated linearly. On the **right**, we show only the dependency of the mean relative AUC only on the grid step with the corresponding $95\,\%$ confidence interval; the optimal setting seems to be the $4\,\mathrm{cm}$.

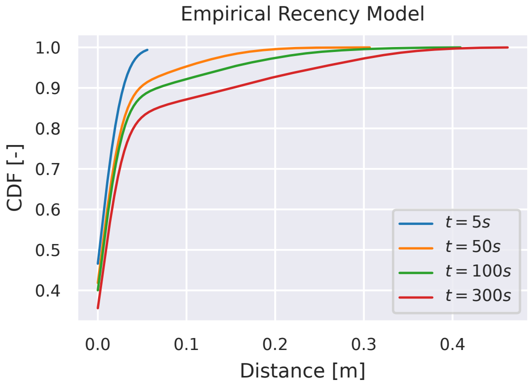

Fig. 4. Empirical recency model RM, showing the cumulative density function (CDF) of displacement of the queen from the last known location $||\mathbf{p}_q - \mathbf{p}||$ for several values of the passed time t.

A. Dataset

The data from first 20 days are used for the training and the rest for testing, because the MoDs require long-term observations and we need to keep the parts temporally separated. To create the recent model RM, we randomly generated 10 thousand continuous tracklets from the training dataset. We generated 2 thousand random deployment tasks D_i to evaluate the proposed searching methods.

The tracking in the dataset is sometimes interrupted, or the queen changes sides. We select D_i so that $\mathbf{p}_q$ is one of the detected queen positions, deployment position $\mathbf{p}_d$ is selected uniformly in the hive area, the duration d is selected uniformly from $0.1\,\mathrm{min}$ to $30\,\mathrm{min}$. We also required a $10\,\mathrm{min}$-long continuous detections to exist starting at time s so that we could evaluate the planned trajectory of the camera with the real position of the queen, given that the success of the search is defined by localization in at most $10\,\mathrm{min}$.

B. Planning Parameters

To solve the graph search problem, it is necessary to discretize the continuous space of the comb area and assign weights to the set of vertices. The standard formulation (as per Eq. (8)) considers the weights completely independent. In our case, the vertices are assigned weights according to the expected queen location; however, the camera's field of view always covers more locations at once, which violates the assumption. The level to which it does so can be adjusted by setting the coarseness of the discretized grid S. Further, as the problem is complex, our solver is a heuristical one. It generates its solution by incremental improvements to an intermediate solution and, therefore, needs a stopping condition set by the maximal runtime M. We expect the maximal runtime to also depend on the grid spacing.

To test both parameters, we define $p_c(u)$ as the amount of probability the camera collects at time u following a solution π. We compute the solutions for each prior model's first few scenarios, randomizing the search's origin and varying step size for grid definition and maximal time for stopping the solver. That gives us the AUC of the cumulative reward as

$$AUC_{S,M} = \int_0^M \int_0^t p_c(u)\,du\,dt. \tag{9}$$

We expect that with higher maximal time, the solution would be better, and we expect a sweet spot for the value of the grid step, so we compute the relative $AUC_{S,M}/AUC_{\min S, \min M}$ and average it over the models.

C. Search Efficiency

To test the abilities of our methods, we compute the solutions for all the scenarios and the time it would take to locate the queen. As the baseline of the current setup, we also include a model *Sequential*, that scans the comb from the bottom to the top by rows in a snake-like fashion. To select the recency parameter h appropriate for our data, we compute the results for all $h \in \{5, 10, \ldots, 30\}$ min, together with pure PM model (referred to as $h = 0\,\mathrm{min}$) and investigate how fast the information from the recent model RM deteriorates. Having the best value for the recency, we look at the median and average searching times for each method. We compare them also statistically using the Quade statistical test for complete block design studies [50], implementation of [51], with Šidák's correction [52] for multiple testing, setting the significance level to $5\,\%$. Finally, we look at the effect of deployment time in combination with the recency half-life on the search times to inspect the deterioration of the short-term information.

VI. RESULTS

A. Planning Parameters

To properly set up the Ms-GVNS solver for problem 7, we had to select the coarseness of the discretized grid S and the maximal runtime M. The values of relative improvement in the AUC metric computed across all of the prior models PM are shown in Fig. 3. We see that for the problems we are solving, the maximal runtime M makes no substantial difference to the mean improvement beyond the lowest tested

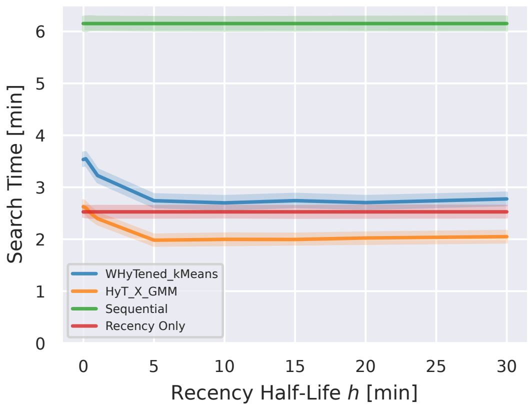

Fig. 5. Mean search time as a function of recency parameters h for different models of dynamics shown with $95\,\%$ confidence interval. This figure shows the effect of the short-term information about the last known position on the effectiveness of the search. The plot only shows the best and worst prior models *HyT_X_GMM*, *WHyTened_kMeans* compared to the *Sequential* baseline and pure recent model RM, labeled *Recent Only*. The result of the *Recent Only* method is shown for reference across all values of h for comparison, although it does not depend on it.

setting $M = 0.1\,\mathrm{s}$. For further experiments, we take $M = 1\,\mathrm{s}$ as that presents a safe choice realistic to the operation of our system where $1\,\mathrm{s}$ is anyway a cost under discernment.

The step in the discretization S, however, makes a great difference, allowing the information collection to be more than three times more effective. In the right panel, we look at the detail of the dependence of the mean AUC improvement on the grid step and see that up to $6\,\mathrm{cm}$ it rises. The values of improvement for $4\,\mathrm{cm}$ and $6\,\mathrm{cm}$ are very simmilar. Given its field of view, the camera sees about $7\,\mathrm{cm}$ in the vertical axis; we choose to be on the safe side of not missing any area and set $S = 4\,\mathrm{cm}$.

B. Empirical Recency Model RM

Figure 4 shows the empirical recency model RM created on the tracklets of the training dataset. The distribution manifests as bimodal, where the queen either rests in the same small area or leaves and slowly gets quite far. This is conditioned on the passed time t, where the longer since the last observation, the more likely the queen has left. After $5\,\mathrm{min}$, there is about $25\,\%$ chance of the queen being away from $\mathbf{p}_q$.

C. Search Efficiency

1) Effects of Recency h: First, we test the effect of the recency half-life h on the average search times. Figure 5 presents the results for a set of models selected for better readability. All informed methods performed significantly better than the baseline *Sequential*, which represents the current setup of the system. All methods informed by a prior model PM have qualitatively similar behavior, so we selected two representatives with overall best and worst performance—*HyT_X_GMM* and *WHyTened_kMeans*.

The prior models, on their own, give worse search times than the recent model. Once the short-term information is

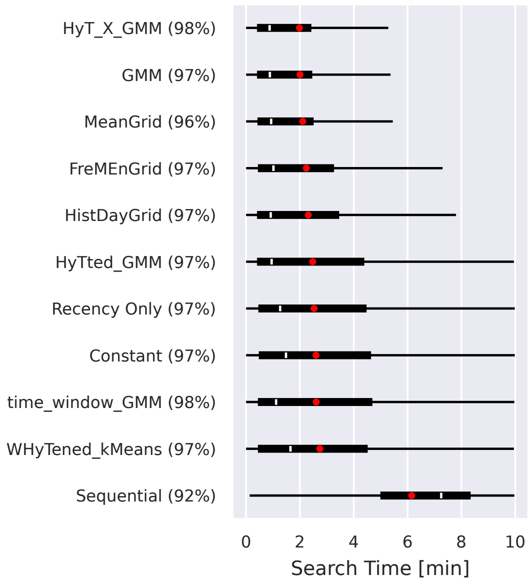

Fig. 6. The search times for each method with recency half-life $h = 5\,\mathrm{min}$, boxes show the interquartile range, white stripes the median, and red dots correspond to the mean. Methods are ordered according to their average search time with the percentage of successful searches shown next to their name.

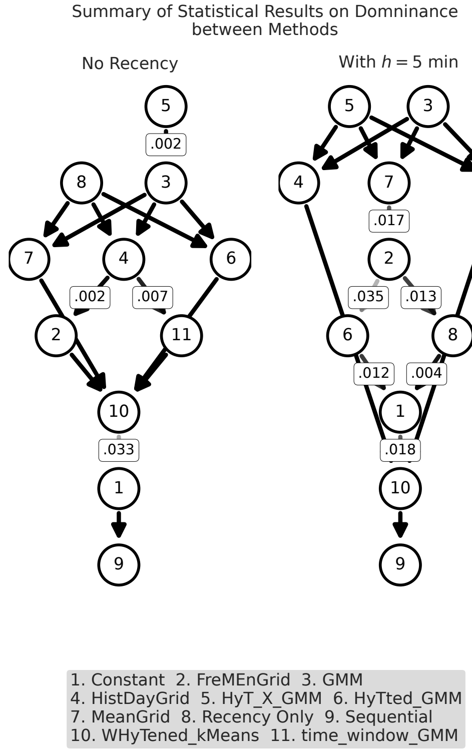
Summary of Statistical Results on Domninance between Methods
No Recency
With *h* = 5 min
5
.002
8
3
7
4
6
.002
.007
2
11
10
.033
1
9
5
3
4
7
11
.017
2
.035
.013
6
8
.012
.004
1
.018
10
9
1. Constant 2. FreMEnGrid 3. GMM
4. HistDayGrid 5. HyT_X_GMM 6. HyTted_GMM
7. MeanGrid 8. Recency Only 9. Sequential
10. WHyTened_kMeans 11. time_window_GMM

Fig. 7. DAG of statistically significant differences in the average search time between individual methods according to the rank on corresponding scenarios. An edge is drawn between two methods if the p-value corresponding to the test of their difference is less than the set significance level of $5\,\%$, and the edge cannot be removed by transitive reduction. For clarity, we explicitly show only those p-values where the $p \geq 1\times10^{-3}$. On the **left**, we show the performance of prior models PM with no recent information, including comparison with *Recent Only* and *Sequential*. On the **right**, with the selected optimal recency parameter $h = 5\,\mathrm{min}$.

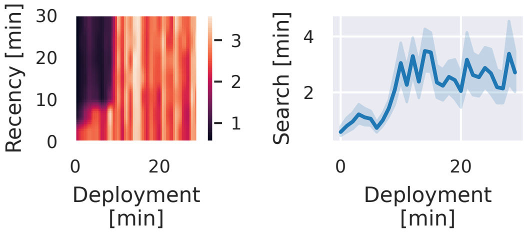

Fig. 8. Mean search time for the best performing model *HyT_X_GMM* as a function of deployment duration and recency parameter h. On the **left**, we give the plot of full interactions; on the **right**, we show only the mean search time (with $95\,\%$ confidence interval) for the previously selected $h = 5\,\mathrm{cm}$. Up to $10\,\mathrm{min}$ of deployment, we see that with the recency model RM the method can find the queen on average close $\approx 1\,\mathrm{min}$, which probably is caused by the queen not leaving its last location $\mathbf{p}_q$ with the search time being dominated by the time to return from the deployment position.

included, the average search time is improved for all the methods, for most improving beyond the *Recent Only* model. We see a minimum at $h = 5\,\mathrm{min}$, which means that at $5\,\mathrm{min}$ of passed time, the search balances equally the prior model PM and the recent model RM. Higher h then leads to a slight worsening of the performance.

2) Search Times: Looking at the average search time when $h = 5\,\mathrm{min}$ in Fig. 6, we see that the three best-performing models both achieve much better performance than baseline and *Recent Only* model, with the best *HyT_X_GMM* improving by $68\,\%$ and $22\,\%$, respectively. Apart from the average case, our methods also achieve much lower 3rd quartile. We also show that apart from differences in search times, the methods have different success rates ranging from $92\,\%$ to $98\,\%$, and our best methods improve the success rate, lowering the failure rate by $75\,\%$. If the queen is not found in the time of $10\,\mathrm{min}$ in the real system, the search would be repeated, but the chances of not finding her in a second run are naturally lower as well.

The results of testing the rankings of the methods according to the search time across all of the testing scenarios are given in Fig. 7. We show a diagram visualization of where the testing, identified statistically significant differences. Methods that do not have a directed path between them were not orderable.

In the left panel, we show the performance of the pure models PM and RM; on the right, we test the with the $h = 5\,\mathrm{min}$, which seems like the reasonable setting of the recency half-life based on our results. When no recency is used, the prior *HyT_X_GMM* and *Recent Only* dominated all the models but could not be ordered between themselves. For combined models, we cannot significantly order *HyT_X_GMM* and *GMM* as the difference in their means is rather small (see Fig. 6), however the success rate is higher.

3) Interaction of Recency and Deployment Duration : The longer the deployment, the less we expect the information from the recent model to matter. Figure 8 shows the average search time as a function of the deployment duration and its interaction with the recency half-life. We see that specifically for cases, when the deployment lasted under $10\,\mathrm{min}$, we our methods achieved the average search times even under $1\,\mathrm{min}$ with the recent information properly included in the search, which is likely due to the queen rarely leaving the original position.

VII. Conclusion

Our robotic system allows for long-term observations of the honeybee queen in a living colony. However, we want it to perform other localized observations in the hive leading to the need for fast relocalization of the queen. As the current searching behavior is naïve, we proposed and tested several methods for informed searching using long-term models of her behavior and short-term memory of her last known location.

In our experiments, all of the methods of searching for the queen using extra information were significantly better than the sequential searching baseline in both the average time and the success rate. We see that both short-term and long-term information separately leads to improvements. Usefullness of the long-term prediction based on periodical processes implies that the queen exhibits sorts of repetitive behavior. When combining the information from PM and RM, we found the recency half-life of $5\,\mathrm{min}$ leads to the best results—we were able to achieve $68\,\%$ search time improvement over the baseline. Given the tested deployment durations of up to $30\,\mathrm{min}$, we conclude that the optimal recovery path can be formed by an intelligent path planning algorithm that takes into account the last known position of the queen as well as her usual locations at a given time.

These results allow us to substantially improve our system for the next season, enabling us to capture a higher amount of extra observations while reducing the risk of missing out on rare queen activities and behaviors. This brings us closer to the biohybrid system integrated into the living ecosystem.

Acknowledgment

This research was conducted under the European Union's Horizon 2020 research and innovation program grant "RoboRoyale" (agreement no. 964492). JM, MK, and TK were funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590). Some authors were further supported by the Grant Agency of the Czech Technical University in Prague (JB, JU and TR by grant no. SGS22/168/OHK3/3T/13, JM by grant no. SGS23/175/OHK3/3T/13).

References

[1] D. Romano, M. Porfiri, P. Zahadat, and T. Schmickl, "Animal–robot interaction—an emerging field at the intersection of biology and robotics," *Bioinspiration Biomimetics*, vol. 19, no. 2, Feb. 2024, Art. no. 020201.

[2] R. Barmak *et al.*, "A robotic honeycomb for interaction with a honeybee colony," *Sci. Robot.*, vol. 8, no. 76, Mar. 2023, Art. no. eadd7385.

[3] M. Chellapurath, P. C. Khandelwal, and A. K. Schulz, "Bioinspired robots can foster nature conservation," *Frontiers Robot. AI*, vol. 10, Oct. 2023, Art. no. 1145798.

[4] W. Rajewicz *et al.*, "Organisms as sensors in biohybrid entities as a novel tool for in-field aquatic monitoring," *Bioinspiration Biomimetics*, vol. 19, no. 1, Nov. 2023, Art. no. 015001.

[5] O. M. Cliff, D. L. Saunders, and R. Fitch, “Robotic ecology: Tracking small dynamic animals with an autonomous aerial vehicle,” *Sci. Robot.*, vol. 3, no. 23, p. Art. no. eaat8409, Oct. 2018.

[6] F. Angelini *et al.*, “Robotic monitoring of habitats: The natural intelligence approach,” *IEEE Access*, vol. 11, pp. 72 575–72 591, Jul. 2023.

[7] M. Stefanec *et al.*, “A minimally invasive approach towards “ecosystem hacking” with honeybees,” *Frontiers Robot. AI*, vol. 9, Apr. 2022, Art. no. 791921.

[8] J. E. Watson *et al.*, “Catastrophic declines in wilderness areas undermine global environment targets,” *Current Biol.*, vol. 26, no. 21, pp. 2929–2934, Nov. 2016.

[9] C. A. Hallmann *et al.*, “More than 75 percent decline over 27 years in total flying insect biomass in protected areas,” *PLOS ONE*, vol. 12, no. 10, Oct. 2017, Art. no. e0185809.

[10] J. Ulrich, J. Blaha, A. Alsayed, T. Rouček, F. Arvin, and T. Krajník, “Real time fiducial marker localisation system with full 6 DOF pose estimation,” *ACM SIGAPP Appl. Comput. Rev.*, vol. 23, no. 1, pp. 20–35, Mar. 2023.

[11] F. Afrati, S. Cosmadakis, C. H. Papadimitriou, G. Papageorgiou, and N. Papakostantinou, “The complexity of the travelling repairman problem,” *Theor. Inform. Appl.*, vol. 20, no. 1, pp. 79–87, 1986.

[12] I. Méndez-Díaz, P. Zabala, and A. Lucena, “A new formulation for the Traveling Deliveryman Problem,” *Discrete Appl. Math.*, vol. 156, no. 17, pp. 3223–3237, Oct. 2008.

[13] R. Roberti and A. Mingozzi, “Dynamic ng-path relaxation for the delivery man problem,” *Transp. Sci.*, vol. 48, no. 3, pp. 413–424, Aug. 2014.

[14] T. Bulhões, R. Sadykov, and E. Uchoa, “A branch-and-price algorithm for the Minimum Latency Problem,” *Comput. Oper. Res.*, vol. 93, pp. 66–78, May 2018.

[15] H. Abeledo, R. Fukasawa, A. Pessoa, and E. Uchoa, “The time dependent traveling salesman problem: polyhedra and algorithm,” *Math. Program. Computation*, vol. 5, no. 1, pp. 27–55, Mar. 2013.

[16] J. J. Miranda-Bront, I. Méndez-Díaz, and P. Zabala, “An integer programming approach for the time-dependent TSP,” *Electron. Notes Discrete Math.*, vol. 36, pp. 351–358, Aug. 2010.

[17] J. J. Miranda-Bront, I. Méndez-Díaz, and P. Zabala, “Facets and valid inequalities for the time-dependent travelling salesman problem,” *Eur. J. Oper. Res.*, vol. 236, no. 3, pp. 891–902, Aug. 2014.

[18] M. T. Godinho, L. Gouveia, and P. Pesneau, “Natural and extended formulations for the Time-Dependent Traveling Salesman Problem,” *Discrete Appl. Math.*, vol. 164, pp. 138–153, Feb. 2014.

[19] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy, “Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem,” *4OR*, vol. 9, pp. 189–209, Jun. 2011.

[20] N. Mladenović, D. Urošević, and S. Hanafi, “Variable neighborhood search for the travelling deliveryman problem,” *4OR*, vol. 11, pp. 57–73, Mar. 2013.

[21] M. M. Silva, A. Subramanian, T. Vidal, and L. S. Ochi, “A simple and effective metaheuristic for the Minimum Latency Problem,” *Eur. J. Oper. Res.*, vol. 221, no. 3, pp. 513–520, Sep. 2012.

[22] Í. Santana, A. Plastino, and I. Rosseti, “Improving a state-of-the-art heuristic for the minimum latency problem with data mining,” *Int. Trans. Oper. Res.*, vol. 29, no. 2, pp. 959–986, Mar. 2022.

[23] J. Mikula and M. Kulich, “Solving the traveling delivery person problem with limited computational time,” *Central Eur. J. Oper. Res.*, vol. 30, no. 4, pp. 1451–1481, Dec. 2022.

[24] E. Koutsoupias, C. Papadimitriou, and M. Yannakakis, “Searching a fixed graph,” in *Automata, Languages Program.*, Paderborn, DE, July 1996, pp. 280–289.

[25] G. Ausiello, S. Leonardi, and A. Marchetti-Spaccamela, “On salesmen, repairmen, spiders, and other traveling agents,” in *Algorithms Complexity*, Rome, IT, Mar. 2000, pp. 1–16.

[26] M. Kulich, J. J. Miranda-Bront, and L. Přeučil, “A meta-heuristic based goal-selection strategy for mobile robot search in an unknown environment,” *Comput. Oper. Res.*, vol. 84, pp. 178–187, Aug. 2017.

[27] M. Kulich and L. Přeučil, “Multirobot search for a stationary object placed in a known environment with a combination of GRASP and VND,” *Int. Trans. Oper. Res.*, vol. 29, no. 2, pp. 805–836, Mar. 2022.

[28] T. P. Kucner, A. J. Lilienthal, M. Magnusson, L. Palmieri, and C. S. Swaminathan, *Probabilistic Mapping of Spatial Motion Patterns for Mobile Robots*, 1st ed. Cham, CH: Springer, 2020.

[29] T. P. Kucner *et al.*, “Survey of maps of dynamics for mobile robots,” *Int. J. Robot. Res.*, vol. 42, no. 11, pp. 977–1006, Sep. 2023.

[30] T. Krajník *et al.*, “Chronorobotics: Representing the structure of time for service robots,” in *Proc. 2020 4th Int. Symp. Comput. Sci. Intell. Control*, Newcastle upon Tyne, UK, Nov. 2020, pp. 1–8.

[31] L. Palmieri *et al.*, Eds., *IEEE Robot. Automat. Lett., Special Issue on Long-Term Human Motion Prediction*, vol. 6, no. 3, Jul. 2021.

[32] T. Vintr *et al.*, “Toward benchmarking of long-term spatio-temporal maps of pedestrian flows for human-aware navigation,” *Frontiers Robot. AI*, vol. 9, Jul. 2022, Art. no. 890013.

[33] Y. Zhu *et al.*, “CLiFF-LHMP: Using spatial dynamics patterns for long-term human motion prediction,” in *2023 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Detroit, MI, USA, Oct. 2023, pp. 3795–3802.

[34] C. Cadena *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[35] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník, “Artificial intelligence for long-term robot autonomy: A survey,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4023–4030, Oct. 2018.

[36] T. Krajník, M. Kulich, L. Mudrová, R. Ambrus, and T. Duckett, “Where’s waldo at time t ? Using spatio-temporal models for mobile robot search,” in *2015 IEEE Int. Conf. Robot. Automat. (ICRA)*, Seattle, WA, USA, May 2015, pp. 2140–2146.

[37] N. Hawes *et al.*, “The STRANDS project: Long-term autonomy in everyday environments,” *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 146–156, Sep. 2017.

[38] V. H. Bennetts, K. Kamarudin, T. Wiedemann, T. P. Kucner, S. L. Somisetty, and A. J. Lilienthal, “Multi-domain airflow modeling and ventilation characterization using mobile robots, stationary sensors and machine learning,” *Sensors*, vol. 19, no. 5, Mar. 2019, Art. no. 1119.

[39] C. S. Swaminathan *et al.*, “Benchmarking the utility of maps of dynamics for human-aware motion planning,” *Frontiers Robot. AI*, vol. 9, Nov. 2022, Art. no. 916153.

[40] K. Žampachů, “Visual analysis of beehive queen behaviour,” B.S. thesis, Dept. Comput. Graph. Interact., Czech Tech. Univ. Prague, Prague, CZ, 2022.

[41] K. Žampachů *et al.*, “A vision-based system for social insect tracking,” in *2022 2nd Int. Conf. Robot., Automat. Artif. Intell. (RAAI)*, Singapore, SG, Dec. 2022, pp. 277–283.

[42] U. Blanke and B. Schiele, “Daily routine recognition through activity spotting,” in *Location Context Awareness*, Tokyo, JP, May 2009, pp. 192–206.

[43] T. Krajník, J. P. Fentanes, J. M. Santos, and T. Duckett, “Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments,” *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 964–977, Aug. 2017.

[44] F. Kubiš, “Application of spatiotemporal modeling used in robotics for demand forecast,” B.S. thesis, Dept. Cybern., Czech Tech. Univ. Prague, Prague, CZ, 2020.

[45] Z. Zhou and D. S. Matteson, “Predicting melbourne ambulance demand using kernel warping,” *Ann. Appl. Statist.*, vol. 10, no. 4, pp. 1977–1996, Dec. 2016.

[46] T. Krajník *et al.*, “Warped hypertime representations for long-term autonomy of mobile robots,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3310–3317, Oct. 2019.

[47] J. M. Santos, T. Krajník, and T. Duckett, “Spatio-temporal exploration strategies for long-term autonomy of mobile robots,” *Robot. Auton. Syst.*, vol. 88, pp. 116–126, Feb. 2017.

[48] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson, “Planning expected-time optimal paths for searching known environments,” in *2004 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sendai, JP, Sep. 2004, pp. 872–878.

[49] A. Mjirda, R. Todosijević, S. Hanafi, P. Hansen, and N. Mladenović, “Sequential variable neighborhood descent variants: An empirical study on the traveling salesman problem,” *Int. Trans. Oper. Res.*, vol. 24, no. 3, pp. 615–633, May 2017.

[50] W. J. Conover, *Practical Nonparametric Statistics*, 3rd ed. New York, NY, USA: Wiley, 1999.

[51] M. Terpilowski, “scikit-posthocs: Pairwise multiple comparison tests in Python,” *J. Open Source Softw.*, vol. 4, no. 36, Apr. 2019, Art. no. 1169.

[52] Z. Šidák, “Rectangular confidence regions for the means of multivariate normal distributions,” *J. Amer. Statistical Assoc.*, vol. 62, no. 318, pp. 626–633, Jun. 1967.