

To Explore or to Exploit? Learning Humans' Behaviour to Maximize Interactions with Them

Miroslav Kulich¹(✉), Tomáš Krajník², Libor Přeučil¹, and Tom Duckett²

¹ Czech Institute of Informatics, Robotics, and Cybernetics,
Czech Technical University in Prague, Prague, Czech Republic
{kulich,preucil}@cvut.cz
<http://imr.ciirc.cvut.cz>

² Lincoln Centre for Autonomous Systems, University of Lincoln, Lincoln, UK
{tkrajnik,tduckett}@lincoln.ac.uk
<http://lcas.lincoln.ac.uk>

Abstract. Assume a robot operating in a public space (e.g., a library, a museum) and serving visitors as a companion, a guide or an information stand. To do that, the robot has to interact with humans, which presumes that it actively searches for humans in order to interact with them. This paper addresses the problem how to plan robot's actions in order to maximize the number of such interactions in the case human behavior is not known in advance. We formulate this problem as the exploration/exploitation problem and design several strategies for the robot. The main contribution of the paper than lies in evaluation and comparison of the designed strategies on two datasets. The evaluation shows interesting properties of the strategies, which are discussed.

Keywords: Distant experimentation · e-Learning · Mobile robots · Robot programming

1 Introduction

With increasing level of autonomy, mobile robots are more and more deployed in domains and environments where humans operate and where cooperation between robots and humans is necessary. One of these are public spaces like libraries, museums, galleries or hospitals which are visited by many people with no or minimal knowledge of these places and which typically need some help. A robot for example can guide a human to a specific place, to direct him/her there or to provide a guided tour through a museum or gallery. In order to act effectively, the robot has to learn not only places where its help is needed but also time periods when people ask for help or interact with the robot at such places.

Imagine for example a commercial building with many offices. The best place to interact with people in the morning is near elevators as people go usually to their job and thus the highest probability of interaction is there. On the

other hand, an entrance to a canteen might be the best place around midday assuming that people go to the canteen for lunch. The problem is that the robot does not know this behavior a priori and it has to learn it. Learning of humans behavior, i.e. where and when humans ask for help, should be done in parallel with interacting with people as well as with other daily tasks of the robot, which leads to the exploration/exploitation problem.

Although the problem looks interesting and has practical applicability, it has not been addressed in the literature. One of the reasons is probably the fact methods for automated creation and maintenance of environment representations that model the world dynamics from a long-term perspective appeared just recently [19]. On the other hand, the work [19] indicates that environment models created by traditional exploration methods that neglect the naturally-occurring dynamics might still perform sufficiently well even in changing environments.

Exploration, the problem how to navigate an autonomous mobile robot in order to build a map of the surrounding environment, has been studied by the robotics community in last two decades and several strategies which can be an inspiration to solution of the exploration/exploitation problem were introduced. The earliest works [9, 22, 23] use a greedy motion planning strategy, which selects the least costly action, i.e. the nearest location from a set of possible goal candidates is chosen. Some authors introduce more sophisticated cost functions, which evaluate some characteristics of the goal and combine them with the distance cost, which represents the effort needed to reach the goal. For example, expected information gain computed as a change of entropy after performing the action is presented in [6], while information gain evaluated as the expected a posteriori map uncertainty is introduced in [1]. Localizability, i.e. expected improvement of robot pose estimation when performing the action is used in [18]. Particular measures are typically combined as a weighted sum. More sophisticated multi-criteria decision making approach, which reflects the fact that the measures are not independent is derived in [2, 3].

All the aforementioned strategies plan only one step ahead. Tovar et al. [21], in contrast, describe an approach which selects the best tour among all possible sequences of goals of the predefined length. We extended this approach in our previous paper [15], where goal selection is defined as the Travelling Salesman Problem. The presented experiments show that the strategy which considers longer planning horizon significantly outperforms the greedy approach.

Another problem related to exploration/exploitation is robotic search which aims to find a static object of interest in shortest possible time. Sarmiento et al. [20] assume that a geometrical model of the operating space is known and formulate the problem so that the time required to find an object is a random variable induced by a choice of a search path and a uniform probability density function for the object's location. They determine a set of positions to be visited first and then find the optimal order by a greedy algorithm in a reduced search space, which computes a utility function for several steps ahead. A Bayesian network for estimating the posterior distribution of target's position is used in [8] together with a graph search to minimize the expected time needed to capture a non-adversarial (i.e. moving, but not actively avoiding searchers) object.

The variant of the problem where the model of the environment is unknown was defined in [16]. A general framework derived from frontier-based exploration was introduced and several goal-selection strategies were evaluated in several scenarios. Based on findings in [16], a goal-selection strategy was formulated as an instance of the Graph Search Problem (GSP), a generalization of the well-known Traveling Deliveryman Problem and a tailored Greedy Randomized Adaptive Search Procedure (GRASP) meta-heuristic for the GSP, which generates good quality solutions in very short computing times was introduced [17].

In this paper, we formulate the exploration/exploitation problem as a path planning problem in a graph-like environment, where the probability of an interaction with a human at a given place/node is a function of time and is not known in advance. A natural condition is to maximize the number of interactions during a defined time interval. To model probabilities at particular places, the Frequency Map Enhancement (FreME_n) [11, 12] is employed, which models dynamics of interactions by their frequency spectra and is thus able to predict future interactions.

Using this model, we designed and experimentally evaluated several planning algorithms ranging from greedy exploration and exploitation strategies and their combinations to strategies planning in a finite horizon (i.e. looking for a fixed finite number of time steps ahead). For the finite horizon variant an algorithm based on depth-first search was designed and all greedy strategies were used as a gain for a single step. Moreover, both deterministic and randomized versions of the strategies, various horizons as well as resolutions of the FreME_n models were considered.

The rest of the paper is organized as follows. The problem is formally defined in Sect. 2, the method for representation and maintenance of environment dynamics is introduced in Sect. 3, while the strategies (policies) to be compared are introduced in Sect. 4. Description of the experimental setup and evaluation results on two datasets are presented in Sects. 5 and 6. Concluding remarks can be found in Sect. 7.

2 Problem Definition

To formulate the problem more formally, let $G = (V, E)$ be an undirected graph with $V = \{v_1, v_2, \dots, v_n\}$ the set of vertices, and $E = \{e_{ij} | i, j \in \{0, 1, \dots, n\}\}$ the set of edges. Let also c_{ij} be the cost of the edge e_{ij} representing the time needed to travel from v_i to v_j and $p_i(t)$ the probability of receiving an immediate reward at vertex v_i at time t (i.e. probability of interaction with a human at vertex v_i at time t). The aim is to find a policy $\pi : V \times T \rightarrow V$ that for a given vertex v_i and time t gives a vertex v_j to be visited at time $t + c_{ij}$, such that the received reward in the specified time interval $\langle t_0, t_T \rangle$ is maximal:

$$\pi = \arg \max_a \sum_{t=t_0}^{t_T} R_a(t),$$

where $R_a(t)$ is a reward received at time t if policy a is followed in the time interval $\langle t_0, t_T \rangle$.

We dealt with the problem when $p_i(t)$ is known in [14], where the task was defined as the Graph Searching Problem [10]. A search algorithm as a variant of branch-and-bound was proposed based on a recursive version of depth-first search with several improvements enabling to solve instances with 20 vertices in real-time.

The situation is more complicated when $p_i(t)$ is not known in advance. Instead, $p_i^*(t)$, apriori estimate of the reward, is available. In this case, a utility of visiting a vertex is twofold: (a) a reward received and (b) refinement of a probability in a vertex:

$$U_i(t) = \alpha R_i(t) + (1 - \alpha)e(p_i^*(t)),$$

where $R_i(t)$ is a reward received in v_i at time t , $e(\cdot)$ is a function evaluating refinement of the probability in a vertex, and α is a weight.

Given this formulation, the problem can be reformulated as finding a policy maximizing the utility:

$$\pi = \arg \max_a \sum_{t=t_0}^{t_T} U_a(t),$$

where $U_a(t)$ is a utility of a vertex visited at time t following the policy a .

3 Temporal Model

Frequency Map Enhancement (FreMEEn) [11,12] is employed to initialize and maintain particular probabilities $p_i^*(t)$. Unlike traditional approaches dealing with a static word, the probabilities in our case are functions of time and these are learnt through observations gathered during the mission. The FreMEEn assumes that majority of environment states is influenced by humans performing their regular (hourly daily, weekly) activities. The regularity and influence of these activities on the environment states is obtained by means of frequency transforms by extracting the frequency spectra of binary functions that represent long-term observations of environment states, discards non-essential components of these spectra and uses the remaining spectral components to represent probabilities of the corresponding binary states in time. It was shown that introducing dynamics into environment models leads to more faithful representation of the world and thus to improved behaviour of the robot in robot self-localization [13], search [14] and exploration [19].

Assume now that the presence of an object in a particular node of the graph is represented by a binary function of time $s(t)$ and the uncertainty of $s(t)$ by the presence probability $p(t)$.

The key idea of the FreMEEn stands in representation of a (temporal) sequence of states $s(t)$ by the most prominent components of its frequency spectrum $S(\omega) = \mathcal{F}(s(t))$. The advantage of this representation is that each spectral component

of $P(\omega)$ is represented by three numbers only which leads to high compression rates of the observed sequence $s(t)$.

To create the FrEMEn model, the frequency spectrum $S(\omega)$ of the sequence $s(t)$ is calculated either by the traditional Fourier transform or by the incremental method described in [11]. The first spectral component a_0 , that represents an average value of $s(t)$ is stored, while the remaining spectral components of $S(\omega)$ are ordered according to their absolute value and the n highest components are selected. Each component thus represents a harmonic function that is described by three parameters: amplitude a_j , phase shift φ_j and frequency ω_j . The superposition of these components, i.e.

$$p^*(t) = a_0 + \sum_{j=1}^n a_j \cos(\omega_j t + \varphi_j), \quad (1)$$

allows to estimate the probability $p(t)$ of the state $s(t)$ for any given time t . Since t is not limited to the interval when $s(t)$ was actually measured, Eq. (1) can be used not only to interpolate, but also to predict the state of a particular model component. In our case, we use Eq. (1) to predict the chance of interaction in a particular node.

The spectral model is updated whenever a state $s(t)$ is observed at time t by the scheme described in [11] for details. This is done every time a robot visits a node v and registers an interaction in the node ($s_v(t) = 1$ in that case) or it experiences that no interaction was done ($s_v(t) = 0$).

4 Policies

Several utilities leading to different policies can be defined. These utilities are typically mixtures of exploration and exploitation gains. The exploration gain of an action expresses the benefit of performing the action to the knowledge of the environment, i.e. amount of information about the environment gathered during execution of the action. The exploitation gain then corresponds to the probability that the action immediately leads to interaction.

More specifically, the **exploitation** utility of the action a which moves the robot to the node v_i is expressed as the estimated probability of interaction at a given time:

$$U_a^{exploitation} = p_i^*(t),$$

while the **exploration** utility for the same case is expressed by entropy in the node v_i :

$$U_a^{exploration} = -p^*(t) \log_2 p^*(t) - (1 - p^*(t)) \log_2 (1 - p^*(t))$$

Figure 1(a) and (b) shows graphs for these two utilities. Note that while exploitation prefers probabilities near 1, exploration is most beneficial in nodes with highest uncertainty.

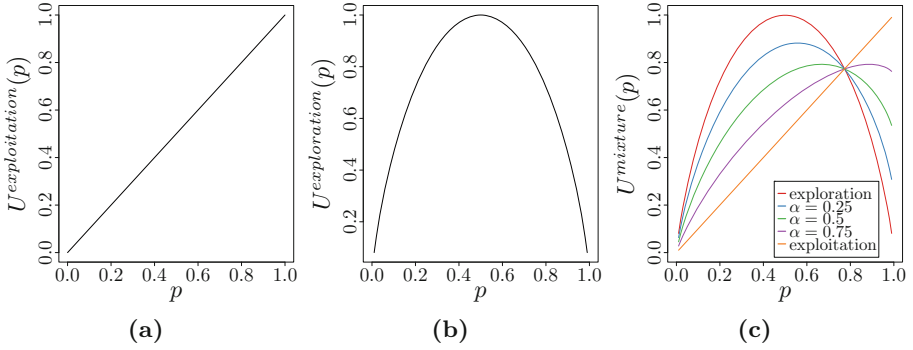


Fig. 1. Utility functions of exploration and exploitation: (a) exploitation utility, (b) exploration utility and (c) their mixture with various weights.

A linear combination of exploration and exploitation defines a new utility which is referred as **mixture** [11]. Ratio of exploration and exploitation is tuned by the parameter α (see also Fig. 1(c)):

$$U_a^{mixture} = \alpha p^*(t) + (\alpha - 1)(p^*(t) \log_2 p^*(t) + (1 - p^*(t)) \log_2(1 - p^*(t)))$$

The disadvantage of this linear combination is that the resulting function has one peak, which moves based on setting of the parameter α as can be seen in Fig. 1(c). In fact, a function which prefers (a) uncertain places, i.e. nodes with probability around 0.5 as well as (b) nodes with high probability of interaction is preferred. An example of such function is shown in Fig. 2(c). This function was formed as a combination of two functions (see Figs. 2(a) and (c)) as is expressed as

$$U_{artificial}(t) = \frac{\alpha}{1 - p^*(t)} + \frac{1}{1 + \beta(\frac{1}{2} - p^*(t))^2}$$

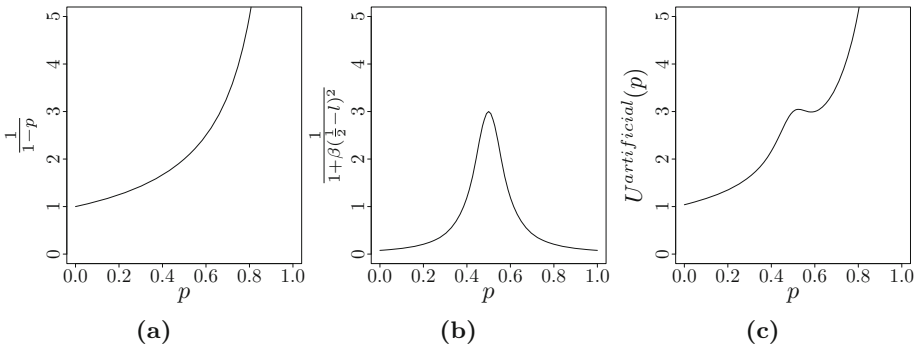


Fig. 2. Construction of the artificial utility. (a) $\frac{\alpha}{1 - p^*(t)}$ function (b) $\frac{3}{1 + 150(\frac{1}{2} - p^*(t))^2}$ and (c) their sum.

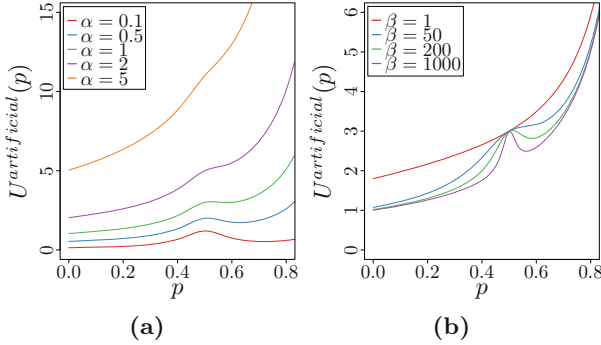


Fig. 3. Various shapes of the artificial utility with one of the parameters fixed. (a) $\beta = 100$ (b) $\alpha = 1$

A shape of the resulting **artificial** utility can be modified by tuning the parameters α and β as depicted in Fig. 3.

A randomized version based on **Monte Carlo** selection is also considered for each of the aforementioned methods. An action with the highest utility is not selected, a random action is chosen instead, but the random distribution is influenced by utilities. In other words, probability of an action to be selected directly is proportional to its utility: the higher the utility the higher chance to be selected. This process can be modeled as an “biased” roulette wheel, where an area of a particular action is equal to its utility.

Strategies using the previously described utilities are greedy in the sense that they consider only immediate off without taking into account subsequent actions. This behavior can be heavily ineffective: the greedy strategy can for example guide a robot into a remote node which can bring slightly more information than other nodes, but with a risk that no (or little) new information will be gathered on the way back. Therefore, utilities that consider some finite planning horizon are introduced. A naïve approach to compute these utilities constructs all possible routes with the given length and take the route with the highest sum of utilities of particular nodes¹ on the route. This approach is not scalable as the number of routes exponentially grows with the size of the planning horizon. Depth-first search in the space of all possible routes is therefore applied with a simple pruning: if the current sub-route cannot lead to a route with higher utility than the current optimum, the whole subtree of routes is discarded from consideration. As will be shown, this technique allows to compute utilities in the presented experiments in reasonable time.

Moreover, three simple strategies are also considered. The first one is called **Random Walk** as it randomly selects a node. A uniform distribution is used in this case, which means that probabilities of all nodes to be selected are equal. While Random Walk serves as a low bound for comparison, the **Oraculum**

¹ Exploration, exploitation, mixture or artificial utility can be used as the utility in a particular node.

strategy provides an upper bound. As the name suggests, using this strategy to select a node always results in an successful interaction. The Oraculum strategy is used only for comparison purposes and employs information about future interactions which is not known to other strategies.

5 Evaluation on the Aruba Dataset

The first evaluation and comparison of the strategies was performed on the Aruba dataset from the WSU CASAS datasets [4] gathered and provided by Center for Advanced Studies in Adaptive Systems at Washington State University. This dataset contains after some processing² data about presence and movement of a home-bound single person³ in a large flat, see Fig. 4 in a period of 4 months. The data were measured every 60 seconds and the flat was represented by a graph with 9 nodes.

Robot behavior was simulated and its success of interactions was evaluated according to the dataset. Given a policy, the robot started in the corridor and it was navigated to the node chosen by the policy as the best every 60 seconds assuming that movement between two arbitrary nodes takes exactly 60 seconds. Every time a new node was reached, the FreMEn model of the node (initially set to constant 0.5) was updated accordingly. This was repeated for the whole dataset and for all the greedy strategies described in the previous section. Moreover, several parameter setups were considered for the Artificial strategy. As the graph is considered to be full and costs of all edges are the same, it has no sense to evaluate strategies with longer planning horizon.

The results are summarized in several graphs. First, the number of interactions, i.e. the number of time moments when the robot was present at the same node as the person was tracked, see Fig. 5. As expected Oraculum provides the best result (we will talk about SuperOraculum in the next paragraph).



Fig. 4. The Aruba environment layout.

² The original dataset [4] contains one year-long collection of measurements from 50 different sensors spread over the apartment and we filtered this data to contain information about presence of the person in particular rooms and at particular times.

³ In fact, the person was not present in the flat occasionally or was visited by another people.

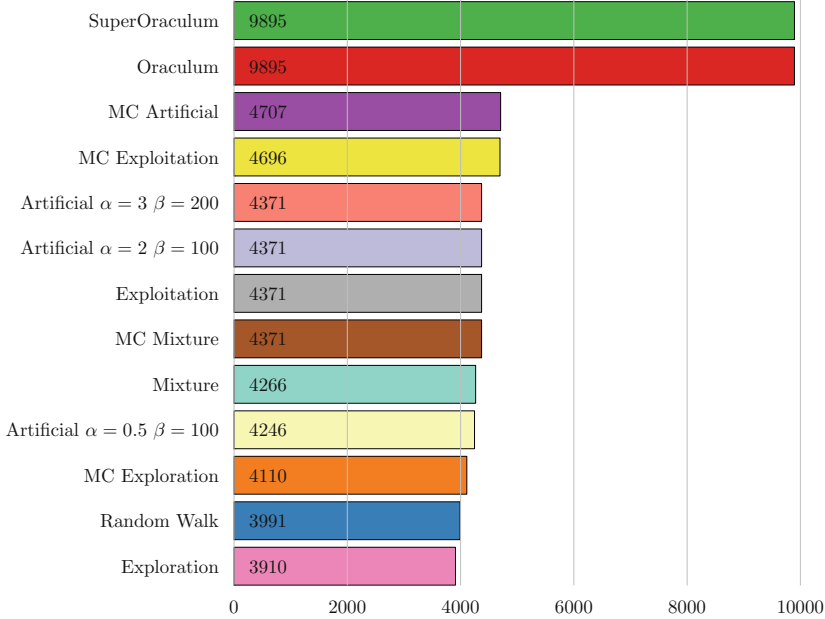


Fig. 5. The number of interactions done for the particular policies. Note that the policies in the legend are ordered according to their performance.

The randomized versions of the artificial utility (with $\alpha = 3$ and $\beta = 200$) and exploitation follow, which are by 8% better than the other methods. The worst method is Exploration, which is even worse than Random Walk and its randomized version is then just slightly better. This is not surprising as the objective of exploration guides the robot to not yet visited areas and thus probability of interaction is small.

The graph in Fig. 6 shows another characteristics of the policies: precision of the model built by FreMEN. Given a model at some stage of exploration/exploitation, Precision is expressed as a sum of squares of differences of the real state and the state provides by FreMEN at the stage estimated over all nodes for all times:

$$error = \sum_{t=0}^T \sum_{i=1}^N (state_i(t) - p_i^*(t))^2,$$

where $state_i(t)$ is the real state of the node i , T is time of the whole exploration/exploitation process, and N is the number of nodes. First, note masterfully biggest error for the Oraculum policy. This is caused by the fact, that this policy guides the robot only to places with a person, thus FreMEN has positive samples only and it assumes that a person is present at all nodes all time. Therefore, another policy called SuperOraculum was introduced, which behaves similarly to Oraculum with one exception: it assumes that there is one person in

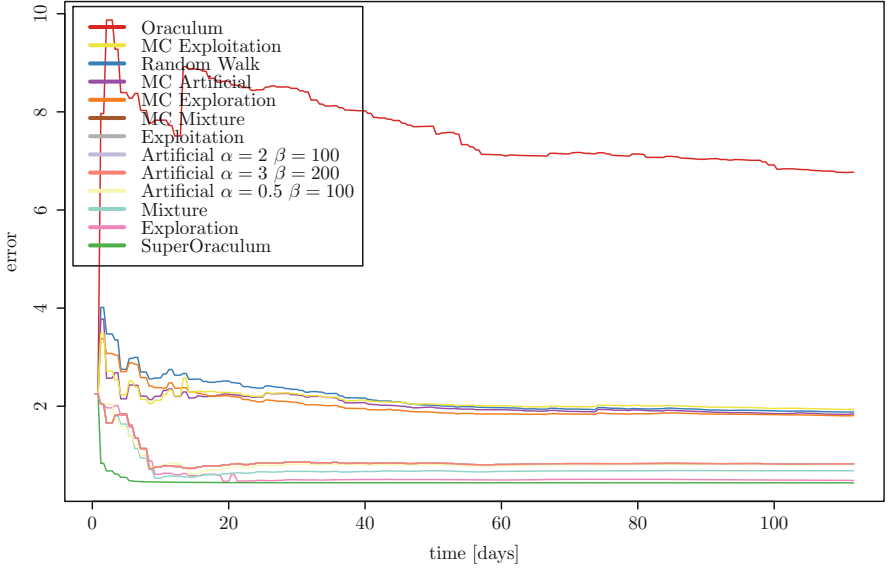


Fig. 6. Progress of FreMEN model precision.

the flat at maximum and thus probabilities of all nodes other than the currently visited are updated also. This update is done the same way the robot physically visits a node and recognizes no interaction. As can be seen, error of this policy is much smaller and serves as a lower limit. Assuming the real policies, the best one is Exploration, which is even comparable to SuperOraculum, followed by the Mixture and two Artificial policies. The other strategies provide similar results. Note also that error of the best strategies almost stabilizes (which means that the model is learned) after 14 days, while it takes longer time for the others.

Finally, the expected number of humans in the flat as assumed by the FreMEN model is depicted in Fig. 7. The number for a given FreMEN model is computed as the average number of nodes, where probability of interaction is higher than 0.5:

$$num = \frac{\sum_{t=0}^T \sum_{i=1}^N (p_i^* > 0.5)}{TN}$$

Note that the real number of humans is lower than one as the person is not always present in the flat. The results correspond to model precision. Again, the best estimates are provided by the Exploration, Mixture and Artificial policies, while the rest highly overestimates the number of humans. When comparing with the number of interactions, the results are almost reversed: the methods with a high number of interactions model the dynamics in the environment with less success than policies with a low number of interactions.

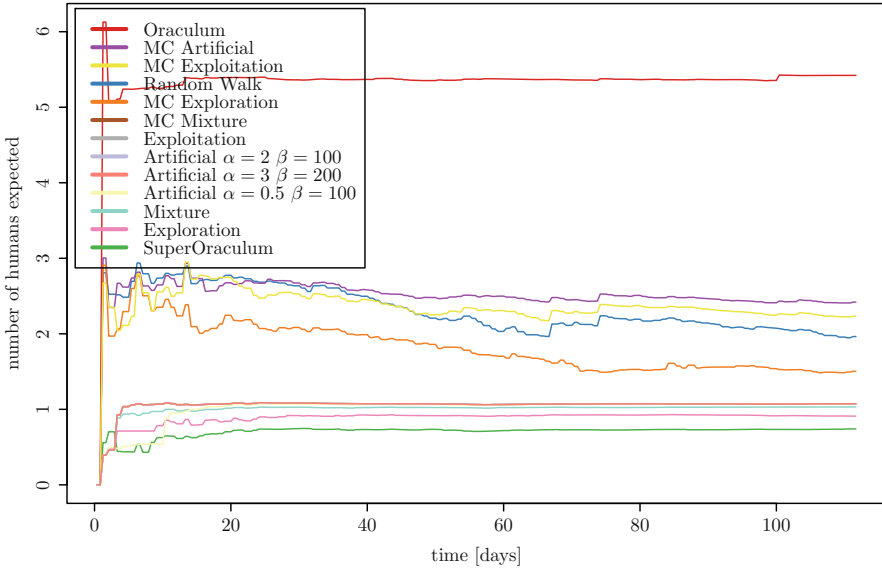


Fig. 7. The Expected number humans in the flat.

6 Deployment at a Care Site

Another evaluation was performed on the data collected within the STRANDS project (<http://strands.acin.tuwien.ac.at>) during a real deployment of a mobile robot at the “Haus der Barmherzigkeit”, an elder care facility in Austria [5, 7]. The robot was autonomously navigated in the environment consisting of 12 nodes (see Fig. 8) and all interactions were logged each minute during a period

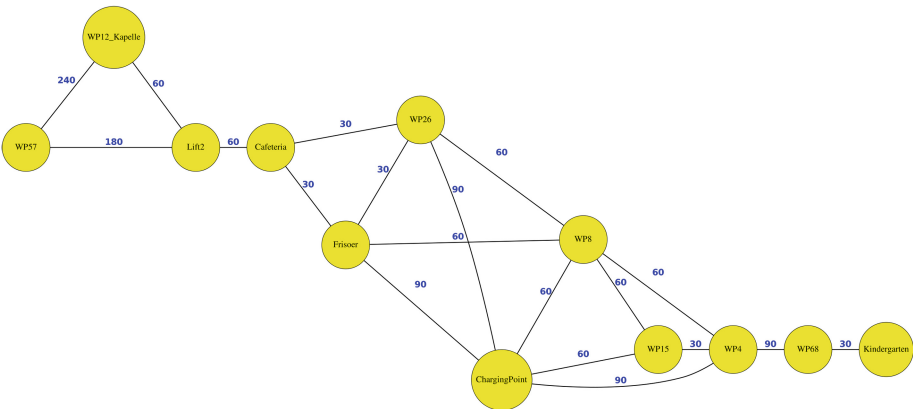


Fig. 8. The graph representing the environment in the hospital.

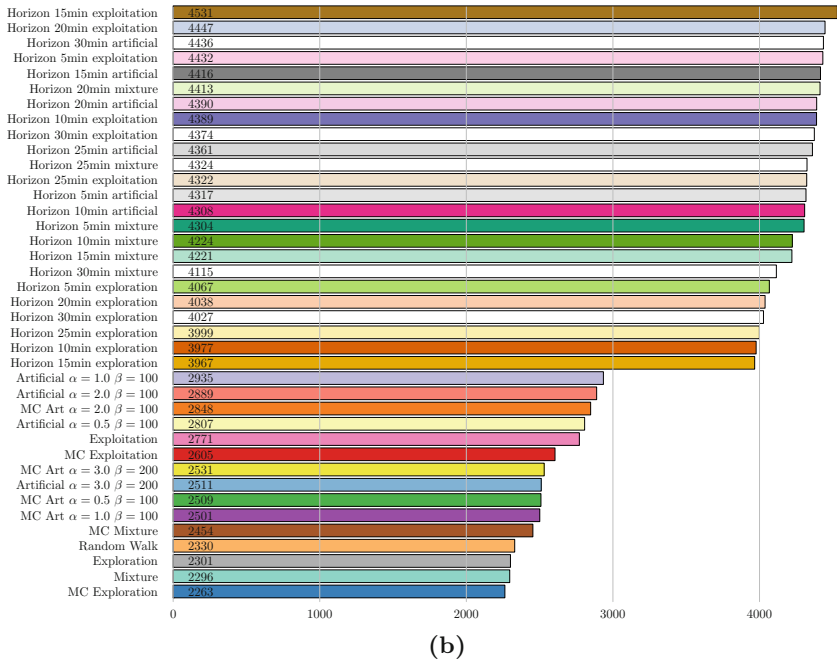
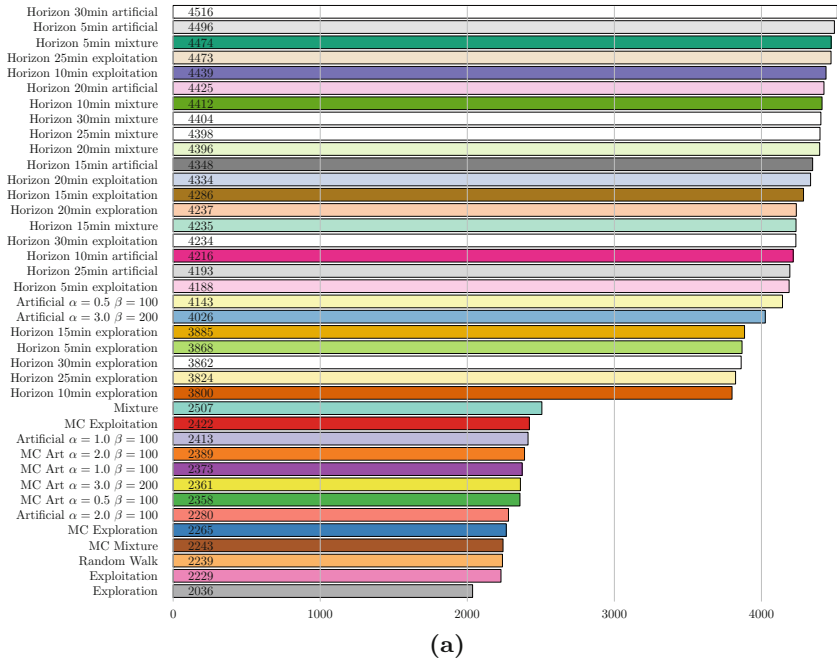


Fig. 9. The number of interactions done for FreMEN with (a) order = 0 (b) order = 1.

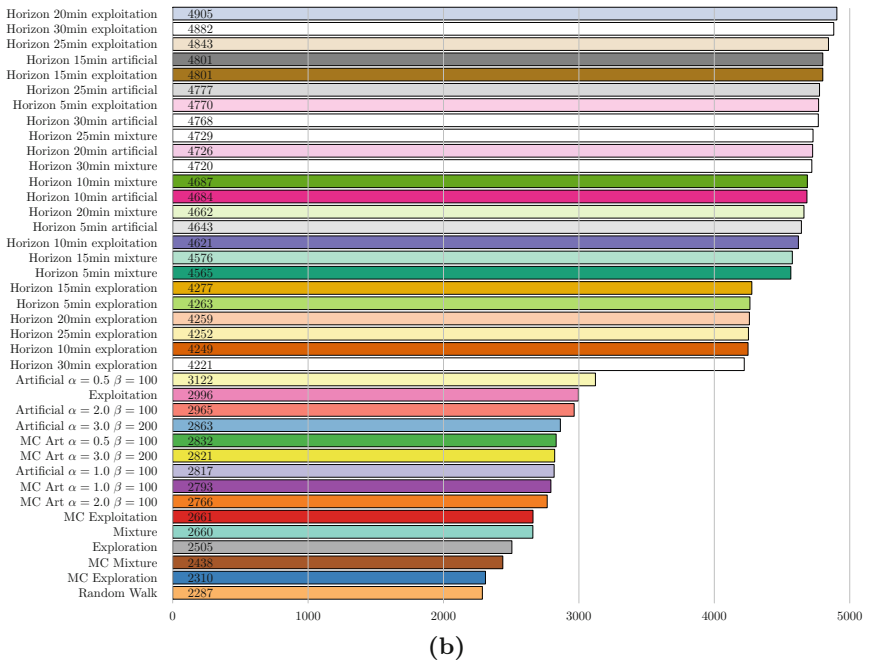
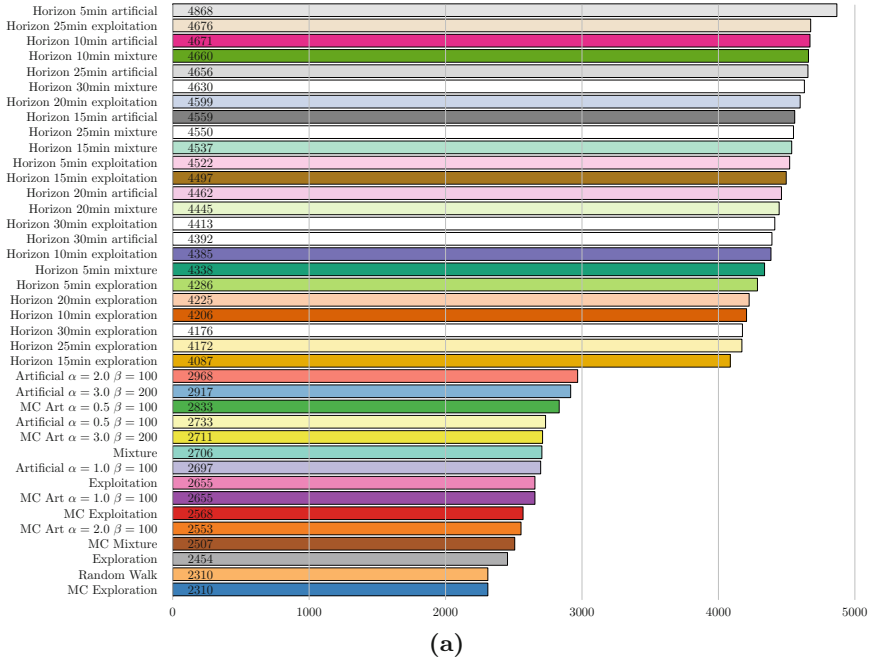


Fig. 10. The number of interactions done for FreMEN with (a) order = 2 (b) order = 4.

of one month, e.g. measurements at 40325 time moments were taken. The data can not be used directly, as information about interactions is available only for places, where the robot was present at a given time. A FreMEn model with $\text{order}=2$ was therefore built in advance and used as ground truth to simulate interactions at all nodes and all times: interaction is detected if the model returns probability higher than 0.5. Contrary to the Aruba experiment, a number of people in the environment varies in time significantly and the time needed to move from one node to another one is not constant (the real values are drawn in Fig. 8). Strategies taking into account a planning horizon are therefore considered together with the policies evaluated in the previous section. To ensure that the robot does not stay on a single spot, we introduces an additional penalty for the current node.

The experiments were performed similarly to the Aruba case. The robot started in the *Kindergarten* node and the whole month deployment was simulated for each strategy. Experiment with each strategy was repeated several times for the order of the FreMEn model equal to 0, 1, 2, and 4. Note that the order equal to 0 means a static model, i.e. probability of interaction does not change in time.

The results are shown in Figs.9 and 10. Generally, the policies planning several steps ahead significantly outperform the greedy ones for all assumed orders, even for the static model. The best results are obtained with the variants employing the artificial and exploitation exploitation utilities followed by mixture. Horizon planning with the exploration utility exhibits a noticeably worse behavior but still better than the greedy policies. Notice also not good performance of pure exploitation for $\text{order}=0$, which is caused by the fact that the model is static and exploitation thus guides the robot to the same nodes regardless time of the day. It can be also seen that model order plays an important role for efficiency; the number of interactions increased between $\text{order}=0$ and $\text{order}=4$ by approx. 9%. Small differences between various lengths of planning horizons can be explained by randomness of interactions and inaccuracy of the models. Interactions can be detected at times and places where they are not expected and do not occur at nodes they are expected by the model.

The proposed horizon planning can be used in real-time. Planning for twenty minutes horizon takes approx. 15 ms, while 300 ms are needed to plan for 30 min horizon, 1600 ms for 35 min horizon, 10 s for 40 min horizon, and 220 s for 50 min planning horizon.

7 Conclusion

The paper addresses the problem of concurrent exploration and exploitation in a dynamic environment and compares several policies to plan actions in order to increase exploitation, which is specified as a number of interactions with humans moving in the environment. Simulated experiments based on real data show several interesting facts:

- The policies with the highest numbers of interactions build the worst models and vice versa. Good strategies should be therefore based on exploitation rather than exploration.

- Consideration of several steps ahead in the planning process leads to significant performance improvement. The best policies with a planning horizon outperform the best greedy ones by 54–80%; the biggest improvement is for the model which assumes a static environment.
- Although FreMen does not model dynamics in the environment exactly, it is precise enough to increase exploitation performance. The higher orders of the model lead to better results.

The next step is to employ and evaluate the best strategies in a real long-term scenario. It will be also interesting to design more sophisticated planning methods, which will be usable in large environments and for longer planning horizons.

Acknowledgments. This work has been supported by the Technology Agency of the Czech Republic under the project no. TE01020197 “Centre for Applied Cybernetics” and by the EU ICT project 600623 ‘STRANDS’.

References

1. Amigoni, F., Caglioti, V.: An information-based exploration strategy for environment mapping with mobile robots. *Robot. Auton. Syst.* **58**(5), 684–699 (2010)
2. Basilico, N., Amigoni, F.: Exploration strategies based on multi-criteria decision making for an autonomous mobile robot. In: *Proceedings of 4th European Conference on Mobile Robots*, pp. 259–264. KoREMA (2009)
3. Basilico, N., Amigoni, F.: Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Auton. Robot.* **31**(4), 401–417 (2011)
4. Cook, D.J.: Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* **2010**(99), 1 (2010)
5. Gerling, K., Hebesberger, D., Dondrup, C., Körtner, T., Hanheide, M.: Robot deployment in long-term care. *Zeitschrift für Gerontologie und Geriatrie* 1–9 (2016). <http://dx.doi.org/10.1007/s00391-016-1065-6>
6. Gonzalez-Banos, H.H., Latombe, J.C.: Navigation strategies for exploring indoor environments. *Int. J. Robot. Res.* **21**(10–11), 829–848 (2002)
7. Hebesberger, D., Dondrup, C., Koertner, T., Gisinger, C., Pripfl, J.: Lessons learned from the deployment of a long-term autonomous robot as companion in physical therapy for older adults with dementia: a mixed methods study. In: *11th ACM/IEEE International Conference on Human Robot Interaction, HRI 2016*, pp. 27–34. IEEE Press, Piscataway (2016). <http://dl.acm.org/citation.cfm?id=2906831.2906838>
8. Hollinger, G., Djughash, J., Singh, S.: Coordinated search in cluttered environments using range from multiple robots. In: Laugier, C., Siegwart, R. (eds.) *Field and Service Robotics. STAR*, vol. 42, pp. 433–442. Springer, Berlin Heidelberg (2008)
9. Koenig, S., Tovey, C., Halliburton, W.: Greedy mapping of terrain. In: *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3594–3599 (2001)
10. Koutsoupias, E., Papadimitriou, C., Yannakakis, M.: Searching a fixed graph. In: Meyer auf der Heide, F., Monien, B. (eds.) *ICALP 1996. LNCS*, vol. 1099, pp. 280–289. Springer, Heidelberg (1996). doi:10.1007/3-540-61440-0_135

11. Krajník, T., Santos, J.M., Duckett, T.: Life-long spatio-temporal exploration of dynamic environments. In: 2015 European Conference on Mobile Robots (ECMR), pp. 1–8, September 2015
12. Krajník, T., Fentanes, J.P., Cielniak, G., Dondrup, C., Duckett, T.: Spectral analysis for long-term robotic mapping. In: 2014 IEEE International Conference on Robotics and Automation (ICRA) (2014)
13. Krajník, T., Fentanes, J.P., Mozos, O.M., Duckett, T., Ekekrantz, J., Hanheide, M.: Long-term topological localization for service robots in dynamic environments using spectral maps. In: International Conference on Intelligent Robots and Systems (IROS) (2014)
14. Krajník, T., Kulich, M., Mudrová, L., Ambrus, R., Duckett, T.: Where's Waldo at time t ? Using spatio-temporal models for mobile robot search. In: 2014 IEEE International Conference on Robotics and Automation (ICRA) (2015)
15. Kulich, M., Faigl, J., Přeučil, L.: On distance utility in the exploration task. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 4455–4460, May 2011
16. Kulich, M., Přeučil, L., Miranda Bront, J.: Single robot search for a stationary object in an unknown environment. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5830–5835, May 2014
17. Kulich, M., Miranda-Bront, J.J., Přeučil, L.: A meta-heuristic based goal-selection strategy for mobile robot search in an unknown environment. *Comput. Oper. Res.* (2016). ISSN 0305-0548, <http://dx.doi.org/10.1016/j.cor.2016.04.029>
18. Makarenko, A.A., Williams, S.B., Bourgault, F., Durrant-Whyte, H.F.: An experiment in integrated exploration. In: IEEE/RSJ International Conference on Intelligent Robots and System, pp. 534–539. IEEE (2002)
19. Santos, J.M., Krajník, T., Pulido Fentanes, J., Duckett, T.: Lifelong information-driven exploration to complete and refine 4D spatio-temporal maps. *Robot. Autom. Lett.* **1**, 684–691 (2016)
20. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: A multi-robot strategy for rapidly searching a polygonal environment. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) *IBERAMIA 2004. LNCS (LNAI)*, vol. 3315, pp. 484–493. Springer, Heidelberg (2004)
21. Tovar, B., Muñoz-Gómez, L., Murrieta-Cid, R., Alencastre-Miranda, M., Monroy, R., Hutchinson, S.: Planning exploration strategies for simultaneous localization and mapping. *Robot. Auton. Syst.* **54**(4), 314–331 (2006)
22. Tovey, C., Koenig, S.: Improved analysis of greedy mapping. In: 2003 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2003), vols. 3 and 4, pp. 3251–3257, October 2003
23. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 146–151. IEEE Computer Society Press (1997)