# Planning Problems for Social Robots

**Gian Diego Tipaldi** and **Kai O. Arras**
Social Robotics Lab
Albert-Ludwigs-University of Freiburg
{tipaldi,arras}@informatik.uni-freiburg.de

### Abstract

As robots enter environments that they share with people, human-aware planning and interaction become key tasks to be addressed. For doing so, robots need to reason about the places and times when and where humans are engaged into which activity and plan their actions accordingly. In this paper, we first address this issue by learning a nonhomogenous spatial Poisson process whose rate function encodes the occurrence probability of human activities in space and time. We then present two planning problems for human robot interaction in social environments. The first one is the *maximum encounter probability planning* problem, where a robot aims to find the path along which the probability of encountering a person is maximized. We are interested in two versions of this problem, with deadlines or with a certainty quota. The second one is the *minimum interference coverage* problem, where a robot performs a coverage task in a socially compatible way by reducing the hindrance or annoyance caused to people. An example is a noisy vacuum robot that has to cover the whole apartment having learned that at lunch time the kitchen is a bad place to clean. Formally, the problems are time dependent variants of known planning problems: MDPs and price collecting TSP for the first problem and the asymmetric TSP for the second. The challenge is that the cost functions of the arcs and nodes vary with time, and that execution time is more important that optimality, given the real-time constraints in robotic systems. We present experimental results using variants of known planners and formulate the problems as benchmarks to the community.

## Introduction

Robots that operate in human environments require the ability to sense people and recognize their activities. But beyond that, they also need the ability to model and reason about human activities, preferences and conventions. This knowledge is fundamental for robots to smoothly blend their motions, tasks and schedules into the workflows and daily routines of people. We believe that this ability is key in the attempt to build socially acceptable robots for many domestic and service applications.

In this paper we take the approach of a spatial Poisson process to learn and represent human activity patterns through a space-time rate function (Ihler and Smyth 2007; Tipaldi and Arras 2011). The function is then used to infer the occurrence probability of the number of activity events in a certain region and during a certain time interval. The Poisson process can describe any type of human activity. However, without loss of generality, we only consider the activity of a person being in a certain position at a certain time. The model then allows to infer the *encounter probability* of humans.

Figure 1: Example office scenario. The picture shows an example path computed by the delivery robot in the simulated office environment.

In this paper, we show that the space-time information on people in an environment leads to two interesting and currently under-explored planning problems in robotics.

First, the model can be used to plan paths along which the probability of encountering a person is maximized. This problem is relevant in a number of applications: a health care robot that needs to find a nurse fast, a surveillance robot that must quickly find a patrolling human colleague, a receptionist robot that knows the location of a particular person at a given time, or a delivery robot that brings urgent goods to people such as hot coffees getting cold.

Second, the model may be used to smartly avoid people by paths that minimize the interference with humans while performing a given task. In this problem, the encounter probability is to be minimized. The Poisson rate function then acts as a cost function of the planning task. Typical applications include a vacuum cleaner that covers an apartment in a way to avoid times and places where it would annoy people or a mail delivery robot that optimally plans its course through a busy office building.

One challenge in these planning problems is the proper spatio-temporal cost function to describe the time-dependent aspect. A further challenge is that under this time-dependent cost function a coverage problem must be solved, where the robot has to visit a particular number of places and must select the order in which they are visited.

The paper is structured as follows. The next section describes the application domain and the simulator that has been built for testing it. It also contains the theory on the spatial Poisson process. We then formally introduce and describe the two problems and present their challenges. We finally conclude the paper in the last section.

# Background

## Domain Description

For the purpose of evaluating the model and testing the planning algorithms, we developed a people activity simulator for indoor environments. Simulation in this case is needed as with real humans, experiments cannot be reproduced.

The engine follows the three-layered agent architecture from (Bonasso et al. 1995) that in our case consists in the layers *activity scheduler*, *activity executor* and *action executor*. At the beginning of the day, the activity scheduler randomly generates a fixed schedule for each agent, based on how these activities are distributed during the day. When an activity is scheduled, it is passed to the activity executor. Every activity is composed of a set of actions such as enter, move, stay or leave, which in turn are activated and deactivated by the activity executor. Once an action is activated, the action executor takes care of its progress and signals back when it reached its final state. Each time an agent is engaged into an activity, its type and place form an activity observation $k_i$. These observations are used to learn the Poisson process. In a second phase, they are used to test the performance of the paths generated by the planners.

The simulator then models a generic workday in which a number of simulated human agents perform activities that are typical in the respective context. Three different scenarios have been designed: home, office, and warehouse. The first scenario models an office environment. To learn realistic activity patterns, an anonymous questionnaire had been handed out to 27 colleagues. The subjects filled in their daily work activities over a two week period (arriving to work, working, eating, smoking, drinking coffee, going to the restroom, etc.) including time and duration of each activity. From this information we learned statistical distributions from which we sample to generate the actual activities of the simulated agents. For this scenario we imagine a delivery robot that brings urgent goods to people whereever they currently stay in the environment.

As a second scenario, we model a warehouse environment. This is a mixed office-factory environment and shares some of the characteristics of the first scenario. The agent activities are partly derived from the data of the questionnaire and typical work shift patterns of warehouse employees. We assume to have two kinds of robots, a delivery robot like the one in the previous scenario and an inspection robot that has to cover the environment during the day.

Finally, we model a home environment as the third scenario. The activities are designed to reflect typical routines of a family with two kids. With this environment we aim at an autonomous vacuum cleaner equipped with a minimum interference planner to minimizes annoyance to the members of the household.

The simulated environments are shown in Fig. 2.

## Spatial Poisson Process

Under the assumption that events in time occur independently, a *Poisson process* can deal with distributions of time intervals between events. Concretely, let $N(t)$ be a discrete random variable to represent the number of events occurring up to time $t$ with rate $\lambda$. Then we have that $N(t)$ follows a Poisson distribution with parameter $\lambda t$

$$P(N(t) = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!} \qquad k = 0, 1, \ldots \tag{1}$$

In general, the rate parameter may change over time. In this case, the generalized rate function is given as $\lambda(t)$ and the expected number of events between time $a$ and $b$ is

$$\lambda_{a,b} = \int_a^b \lambda(u)\, du. \tag{2}$$

A homogeneous Poisson process is a special case of a non-homogeneous process with constant rate $\lambda(t) = \lambda$.

The *spatial* Poisson process introduces a spatial dependency on the rate function given as $\lambda(\vec{x}, t)$ with $\vec{x} \in X$ where $X$ is a vector space such as $\mathbb{R}^2$ or $\mathbb{R}^3$. For any subset $S \subset X$ of finite extent (e.g. an area in space), the number of events occurring inside this area can be modeled as a Poisson process with associated rate function $\lambda_S(t)$

$$\lambda_S(t) = \int_S \lambda(\vec{x}, t)\, d\vec{x}. \tag{3}$$

Learning the spatio-temporal distribution of events in an environment is equivalent to learning the generalized rate function $\lambda(\vec{x}, t)$. A first approach is to perform maximum likelihood estimation. This approach however has the problem that there is no information about cells that have not been observed yet. Rarely used places in an environment cannot be properly initialized. We therefore take a Bayesian inference approach that can provide information on such cells through priors. We model the parameter $\lambda$ using a Gamma distribution, as it is the conjugate prior of the Poisson distribution. Learning the rate parameter $\lambda$ then becomes a parameter estimation problem of a Gamma distribution parametrized by $\alpha$ and $\beta$. The posterior mean of $\lambda$ in a single cell is finally obtained as the expected value of the Gamma,

$$\widehat{\lambda}_{\text{Bayesian}} = \mathbb{E}[\lambda] = \frac{\alpha}{\beta} = \frac{\#\text{positive events} + 1}{\#\text{observations} + 1}\ . \tag{4}$$

Obtaining the encounter probability along a path is done as following. Since a path is a mapping from time to space, $\mathcal{P} : t \to \vec{x}$, we have that the number of people encountered in a certain path follows a non-homogeneous Poisson process whose rate function depends on the path itself

$$\lambda^{\mathcal{P}}(t) = \lambda(\mathcal{P}(t), t). \tag{5}$$

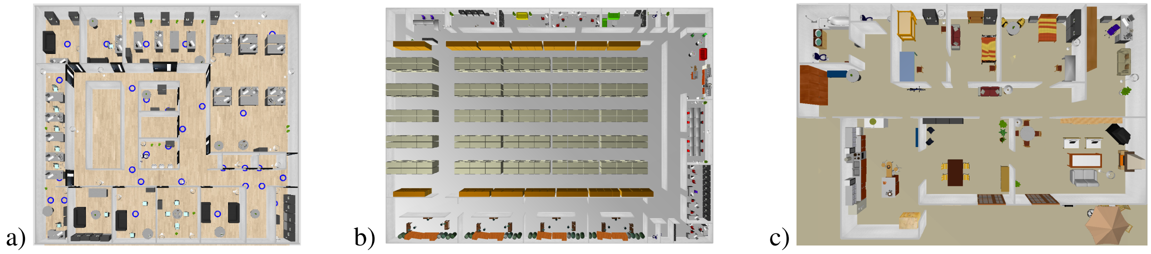

Figure 2: The three simulated environments: a) office, b) warehouse and c) apartment.

The probability of encountering at least one person along a path is obtained by considering the probability of not encountering anyone and using the law of total probability

$$\begin{aligned} p(N^{\mathcal{P}}(t_{max}) > 0) &= 1 - p(N(t_{max}) = 0) \\ &= 1 - e^{-\int_{t_0}^{t_{max}} \lambda^{\mathcal{P}}(u)du}, \end{aligned} \quad (6)$$

This probability, furthermore, is also the probability of disturbing a person, showing the duality of the problems that will be presented in the following section.

## The Problems

In this section, we will pose the problems in terms of state space and possible actions, so that it can be addressed in a computationally efficient way. First, we note that obtaining a solution to these path planning problems in the continuous space is infeasible as it would mean to search on the manifold of all possible continuous curves in a three dimensional space. For this reason, we overlay a regular metric grid onto the environment in space and time. Considering the grid approximation and assuming the robot moves at constant speed, a path is represented by the sequence of grid cells the robot traverses. Integrals are then proportional to the sum of the Poisson rate over those cells.

Formally, both problems can be formulated as follows. Our state space is represented by the cells that are within the free space of the environment and the day time interval the robot is in, $s = \{i, j, \tau\}$. In each cell, a set of (maximum) nine actions are defined. They account for movements in the 8-neighborhood and waiting in the current cell. The actions are not deterministic and their outcome – the so called state transition distribution – may be derived from the robot odometry subject to uncertainty reflecting the accuracy of the odometric system. Whenever the robot reaches a cell in the world, it collects a *reward* (respectively a *penalty*) that concurs to increase the encountering probability of the current path. Since the probability is proportional to the Poisson rate, the reward (penalty) is given by $R(s') = \lambda_{i'j'}(\tau')$. In the following we refer to this setup to describe the two problems.

### Maximum Encounter Probability Planning

A robot that needs to find a person under time constraints requires a path along which the probability to encounter a person is maximized.

Given the problem setup and an initial state $s_0$, we are interested in finding a sequence of states $\mathcal{S} = \{s_0, s_1, \ldots, s_n\}$, such that they form the best path to find a person given either a time or a probability constraint. In case of the time constraint, the robot is given a deadline, resulting in limiting the length of the sequence. In case of a probability constraint, the robot has to plan a path with a given confidence value (or above) that has the minimum length. Although the two problems look similar, they differ in the underlying mathematical structure. The first problem can be addressed with a modification of a finite horizon MDP, where the horizon is reduced at every step. The optimal solution can be found using dynamic programming with a slight modification of the Bellmann algorithm. The second problem is much more complex and it can be reduced to a price collecting traveling salesman problems (Balas 1989) with time-dependent rewards.

We have compared a finite horizon MDP planner with several informed (using the map) and uninformed (not using the map) heuristic planners in the office environment. These preliminary experiments confirm the optimality of the MDP solution but also that a greedy heuristic planner can give good results. The poorest two strategies to find people in this setting turned out to be random walk and waiting on the spot. Details can be found in (Tipaldi and Arras 2011).

### Minimum Interference Coverage

Coverage is a well-known robotics planning problem where a robot needs to sweep over a particular area subject to minimum time, path length or number of turns. Typical applications include floor care or demining. Human-aware coverage is, to our knowledge, a novel problem relevant in all coverage applications in which the operation of the robot causes a disturbance or potential thread to people. Thus, a socially aware robot minimizes the interference with people in the environment during task execution.

Formally, the problem shares the same cell representation than the previous one with the exception that the reward is now a *cost* function and needs to be minimized. The solution is a sequence of states such that all states are visited at least once while minimizing the encounter probability. The order of the states is important since both the movement costs and the interference probability strongly depend on when a state is visited. The problem can be reduced to an asymmetric traveling salesman problem with time-dependent costs (ATDTSP). The asymmetry arises from the time-dependent structure of the costs. In fact, going from node $a$ to node $b$ is different than the opposite since the nodes are visited at different times and thus have different costs.

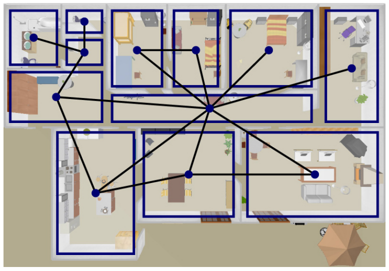

Figure 3: The room graph of the apartment domain as used for the human-aware coverage problem.

We have conducted comparisons in the home environment in which we evaluated a dynamic program for the ATDTSP with two informed TSP heuristics (nearest neighbor and greedy) and an uninformed TSP planner. Over a series of runs with varying cleaning times per room and start times, the ATDTSP planner clearly outperforms all other strategies in terms of number of disturbed persons and total interference time with the human agents.

## Challenges and Discussion

The above problems introduce several planning challenges mainly because of the very limited computational resources on board of real robots that prevent them to run the optimal solutions in all but very small domains. Since robot actions are uncertain and environments change over time, plans need also to be available in quasi real-time. Time-dependent costs, moreover, add another level of difficulty that requires standard algorithms to be modified accordingly.

The complexity of the dynamic program for the maximum encounter probability with deadlines is $O(h^3)$ in both space and time, where $h$ denotes the deadline. Although this is polynomial, the runtime of the Bellmann algorithm is very slow for large environments in practical applications. Therein, average grid cell sizes are typically in the range of 10 to 20 cm, meaning that we face problems in the order of millions of cells. Only tailored heuristics can solve such problems at the expense of optimality. The situation is even more problematic for the problem statement with probability constraints. The underlying TSP problem is known to be NP-complete and the time-dependent costs make 2-Opt and 3-Opt heuristics more computationally involved.

The same situation arises for the minimum interference coverage problem. Here, however, a simplification can be made in terms of state space. One option (taken in our experiments) is to increase the cell size to the level of entire rooms and solve the reduced problem (see Fig. 3). A specialized geometric coverage algorithm can then be used to cover the area of each room. However, coverage tasks in large environments with many rooms quickly become intractable again. The real challenge resides in the asymmetry of the problem itself, since the heuristics for the asymmetric TSP are not as good as the heuristics for the symmetric version.

Another aspect to account for is the transformation of the task into a TSP. The original graph is very sparse since the robot can only move through neighboring cells. To reduce the problem to a TSP of whichever form, the graph must first be completed using either the Johnson or the Floyd-Warshall algorithm whose complexity is minimum $O(V^2 \log(V)+VE)$, where $V$ is the number of nodes (cells) and $E$ the number of edges. Considering the time variability of the costs, the algorithms need to be repeated once for each possible time step, multiplying another $V$ term.

In the light of this discussion, the transformation of the problems into their TSP equivalents is likely not the right way to pursue, although they can theoretically provide an optimal solution. We argue that specific heuristics or algorithms that exploit the sparse nature of the problem should be developed. We thus see the contribution of this paper also in the introduction of two new planning benchmarks for probabilistic time-varying domains as they arise in human-aware planning and hope that progress in the near future will produce efficient heuristics for the two presented problems. To this end, the simulation environment is made publicly available on the webpage of the authors.

## Conclusion

We have introduced two novel human-aware planning problems for social robots that arise when robots fulfill a number of tasks in a space shared with people. We presented a simulation engine for such environments, three realistic scenarios, and proposed a spatial Poisson process model to learn and reason about spatio-temporal human activity events.

The challenges posed in this paper have several real-world applications ranging from health care robots that need to find a nurse as fast as possible, over delivery robots that bring urgent goods to people, to vacuum robots that learn which are the right times and places to clean an apartment to minimize annoyance to people.

We believe that these problems are attractive for the planning community. They are intrinsically combinatorial optimization problems with cost/reward functions that vary over time. We presented how they can be solved optimally with particular instances of MDP and TSP problems and gave preliminary results of their performance. However, the optimal solutions scale poorly with environment size which makes them intractable for many real-world problems given the limited embedded computational power of robots. More research is needed to find efficient near-optimal heuristics.

## References


Balas, E. 1989. The prize collecting traveling salesman problem. *Networks* 19:621–636.

Bonasso, R. P.; Kortenkamp, D.; Miller, D. P.; and Slack, M. 1995. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence* 9:237–256.

Ihler, A. T., and Smyth, P. J. 2007. Learning time-intensity profiles of human activity using non-parametric Bayesian models. In *Neural Information Processing Systems 19*, 625–632.

Tipaldi, G. D., and Arras, K. O. 2011. I want my coffee hot! Learning to find people under spatio-temporal constraints. In *Proc. IEEE Int. Conf. on Robotics and Automation*.