# Multi-Robot Allocation of Assistance from a Shared Uncertain Operator

Clarissa Costen
University of Oxford
Oxford, United Kingdom
clarissa@robots.ac.uk

Anna Gautier
KTH Royal Institute of Technology
Stockholm, Sweden
annagau@kth.se

Nick Hawes
University of Oxford
Oxford, United Kingdom
nickh@robots.ox.ac.uk

Bruno Lacerda
University of Oxford
Oxford, United Kingdom
bruno@robots.ox.ac.uk

## ABSTRACT

Shared autonomy systems allow robots to either operate autonomously or request assistance from a human operator. In such settings, the human operator may exhibit sub-optimal behaviours, influenced by latent variables such as attention level or task proficiency. In this paper, we consider shared autonomy systems composed of multiple robots and one human. In this setting, we aim to synthesise a controller that selects, at each decision step, the actions to be taken by each robot and which (if any) robot the human operator should assist. To efficiently allocate the human operator to a robot at any given time, we propose a controller that reasons about the uncertainty over the latent variables impacting the human operator's performance. To ensure scalability, we use an online bidding system, where each robot plans while considering its belief over the human's performance, and bids according to the direct benefit of human assistance and how much information will be gained by the system about the human. We experiment on two domains, where we outperform approaches for allocation of human assistance that do not consider the human's latent variables, and show that the performance of the overall system increases when robots consider the information gained by requesting human assistance when bidding.

## KEYWORDS

Multi-agent planning, Planning under Uncertainty, Planning with abstraction and generalization

**ACM Reference Format:**
Clarissa Costen, Anna Gautier, Nick Hawes, and Bruno Lacerda. 2024. Multi-Robot Allocation of Assistance from a Shared Uncertain Operator. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

This work is licensed under a Creative Commons Attribution International 4.0 License.

*Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.* © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

## 1 INTRODUCTION

As autonomous robots become increasingly reliable and integrated into our lives [18, 21], they occasionally enter situations where they cannot complete their task without human intervention. Shared autonomy systems address this challenge by allowing the control of a robot to be shared between a human and an autonomous agent. As automation improves, the level of human supervision required for each robot decreases. In this context, we examine scenarios where a single human operator oversees multiple robots. Allocating human assistance optimally to these robots becomes a complex issue, particularly when multiple robots require assistance simultaneously. Moreover, the human operator may possess varying skills, making them more proficient at assisting with some tasks compared to others, but these skill differences may be unknown to the robots. In this paper, we introduce an efficient approach to allocate human assistance within multi-robot systems characterized by uncertainty surrounding the human operator's performance.

Shared autonomy systems reduce the workload on the human operator [2] by deciding when a human or autonomous controller should operate the robot. The system's controller determines the optimal operator (human or autonomy) by reasoning over its internal model of both the human and the autonomous robot. Typically, these internal models use Markov decision processes (MDPs) [11] to account for the stochastic nature of the operators. However, human behaviour is inherently diverse, making it challenging to predict human performance accurately before execution. To address this variability, Costen et al. [9], Nanavati et al. [22], Wray et al. [30] have utilized partially observable MDPs (POMDPs) and Bayes-adaptive MDPs (BAMDPs) to model the human operator, incorporating probability distributions to represent the human's true location or internal state. The papers that consider uncertainty over the human focus on systems involving one robot and one human. We extend this to systems with multiple robots and one human.

We define our system as a multi-robot shared autonomy system, where a human operator supervises multiple robots. The multi-robot shared autonomy system can be framed as a scenario where robots must share a constrained resource: the human operator. Multi-robot systems with constrained resources typically use decentralised approaches, as solving the joint model can become computationally infeasible. These approaches often assume fixed resource availability with known impacts on the robots [23], such

as battery charge, and typically provide offline solutions where the resources are pre-allocated to robots before deployment [32]. However, when there is uncertainty over the outcome of consuming a resource, decentralised approaches fail to provide policies that change as other robots observe the effect of a resource. Thus, decentralised approaches are not suited to our scenario, where the robots must adapt their decision-making as other robots learn about the human operator's abilities through observations.

We propose a bidding system to distribute uncertain human assistance efficiently, avoiding the need to solve a joint model. The human is described using a hidden parameter polynomial MDP (HPP-MDP), a class of uncertain MDP where the transition function is parameterised by a set of latent parameters. At every decision step, each robot uses BAMCP [17], an online tree-search algorithm, to plan for a combined human-robot HPP-MDP that models a single-robot shared autonomy system. This assumes that the robot can always access human assistance. If the optimal action requires human operation, then the robot submits a bid to the centralised controller. This bid is based on the difference in value between the human taking control and the robot executing autonomously. The centralised controller then awards human help to the highest bidder, with the other robots executing their optimal autonomous action. Then, the centralised belief distribution over the human operator's latent parameters is updated by considering the outcome of the human's action. This is then repeated until all robots have completed their tasks.

Planning with single-robot models introduces challenges: robots cannot predict if the information they receive is useful to others, and other robots may make observations that alter the posterior distribution over the latent parameters, hindering long-term planning. This is because planning over single-robot shared autonomy model does not capture the long-term effects of the other robots in the true, joint shared-autonomy model. These issues lead to undervaluing explorative actions, which leads to sub-optimal human help allocation. To address this, we add an exploration bonus which considers the variance in the posterior distribution over the latent parameters. The HPP-MDP formulation of the shared autonomy system allows us to keep a closed-form representation of the posterior distribution over the human's performance, reducing the computational complexity of calculating the variance.

Our main contributions are the modelling of a multi-robot shared autonomy system with uncertain human performance as an uncertain MDP; the bidding system to allocate human help; and the explicit reward function motivating the reduction in the variance over the belief in human performance. To the best of our knowledge, this is the first approach proposed to solve multi-robot shared autonomy systems where there is uncertainty in the human operator's performance. We compare our approach to 1) solving the joint model using the BAMCP algorithm [17], and 2) a decentralized approach proposed by Dahiya et al. [12], which considers a multi-robot system where the human operators have known success rates. We empirically show that our approach outperforms the joint model under time constraints and the method proposed in Dahiya et al. [12] in two domains.

## 2 RELATED WORK

### 2.1 Shared Autonomy Systems

Shared autonomy systems are useful in settings where the workload on the human can be reduced by letting autonomy take control in low-risk tasks [27]. Common assumptions in many of these systems is that the human operator can operate the system at all times and they can act perfectly [3]. Some systems use the human operator as an example to learn from [14, 25], with the long-term objective of reducing the reliance on human operators over time.

However, human operators' capabilities are not always guaranteed and can be influenced by a variety of factors, such as fatigue, distraction, and ability [31]. The variance in the human operator's performance can be explicitly modelled using Markov models [8, 15, 20]. This allows the autonomous agent to consider the stochasticity in the human operator's behaviour when planning, such as the probability of the human failing a task. However, the factors that affect the human's performance are not always observable to the agent. Jean-Baptiste et al. [19] and [9] use POMDPs to model the uncertainty over the human operator's behaviour. The factors affecting the human operator (such as skill level or fatigue) are modelled as latent internal states, and the agent must maintain and reason over a belief distribution over the latent states. By solving models with partial observability fully, it is possible find the policy for the agent where exploration (learning about the human operator) and exploitation (using the information about the human to gain rewards) are optimally balanced. These papers only consider systems where there is one agent and one human operator. In contrast, we consider the setting where there are multiple agents and one human operator.

### 2.2 Multi-Agent Shared Autonomy Systems

As the amount of human intervention required per robot decreases, we can move to a system where a single human operator supervises multiple robots. This can be useful in search-and-rescue missions, as well as multi-robot systems in warehouses [26]. In these systems, using a single joint model to describe the multi-agent system can be intractable, as the state and action space grows exponentially with the number of agents. To solve the joint model, Rosenfeld et al. [26] restricts the horizon of the problem to only 1 or 2 decision steps. This short horizon allows the problem to be solved fully for the current and next action, but is unable to consider the effects of those actions into the future.

Swamy et al. [28] highlights the difficulty of the human appropriately choosing the best robot to help in scenarios with large numbers of robots. They focus on the human choosing a robot to help, while we consider the robots determining how useful human help would be. They suggest a method where the system learns the human's preferences for helping robots when there is a small number of robots. This learnt preference is generalised to fit problems with a larger number of robots, thus able to suggest the most appropriate robot for the human to help. However, this method assumes that the human always chooses the optimal robot to help initially, and does not consider the possibility that the human may choose the wrong robot.

Other approaches for allocating human assistance in multi-agent shared autonomy systems impose rules onto the system to reduce

the complexity of the problem. Cai et al. [7] assumes that the time taken by the human and the robot for every task is exactly known and that the human is always faster or equal to the robot. These assumptions reduce the problem to a deterministic scheduling problem.

Similar to our work, Dahiya et al. [12] considers a multi-agent system where the human operator can take over at an agent's request. They consider the possibility that the human operator may fail a task, and use an MDP to model the dynamics of the human and the autonomous agents. The agent with the highest benefit from human intervention is given human help, and this benefit is defined as the difference in the relative expected cost of the agent doing the task autonomously versus the human helping. These benefits are calculated offline before the system is deployed, and they assume that the probability of the human failing a task is known. Therefore, their system is unable to adapt if the human operator does not perform tasks as expected. We compare our approach to theirs in Section 6.

### 2.3 Multi-agent constrained resource problems

Systems with multiple agents have been modelled using multi-agent MDPs (MMDPs) [5], which have joint state and action spaces. The state space of an MMDP grows exponentially with the number of agents, making it computationally intractable to solve fully. Thus MMDPs are generally solved using decentralised methods when the transition probabilities for each agent are independent [4]. Our multi-agent shared autonomy system can be framed as a multi-agent system where the agents must share a constrained resource, the human operator with independent transition probabilities. In MMDPs with global resource constraints, each agent has its own set of transition and reward dynamics, but their actions are coupled by a global resource constraint [32], such as advertising space [6].

Both [23, 32] solve constrained resource multi-agent systems using a mixed integer linear program (MILP) to pre-allocate the resource offline to each agent at every decision step. While MILP-based solutions can guarantee that hard constraints are satisfied, they are not scalable to large problems as their complexity grows exponentially with the horizon. If the resource has a soft constraint where some constraint violations are permitted, we can use column generation algorithms for offline policy synthesis that guarantees the resource constraint to be satisfied in expectation [29, 33]. Column generation algorithms are more scalable than MILP, as they parallelise the computation by finding policies for each agent instead of solving the joint problem. Alternative approaches for soft constraints consider the risk of resource violations in the form of a chance-constraint [13] and a conditional value-at-risk constraint [16].

In contrast to [29], where the agents have partial observability of their state and a deterministic shared resource, we consider the case where the agents have full observability of their state, but they must reason over the uncertainty over the effect of the shared resource on the transition probabilities. This leads to a problem where the observation made by one agent may affect the optimal policy for another agent, because the other agents' observation may resolve some of the uncertainty in the transition function. Therefore, we cannot use decentralised methods such as column generation. In this paper, we propose an approach to address this problem of a constrained resource with an uncertain effect on the agent in a multi-agent system without solving the joint model.

## 3 PRELIMINARIES

MDPs have been widely used in planning problems to describe the stochastic behaviour of agents. MDPs allow us to reason over aleatoric uncertainty, such as the randomness in the success of the autonomous robot completing a task.

*Definition 3.1 (MDP).* A Markov Decision Process (MDP) is defined by the tuple $M = \langle S, \bar{s}, A, T, R, \gamma \rangle$, where:

- $S$ is the set of states and $\bar{s}$ is the initial state;
- $A$ is the set of actions;
- $T(s, a, s') = P(s' \mid s, a)$ is the probabilistic transition function;
- $R : S \times A \to \mathbb{R}$ is the reward function;
- $\gamma$ is the discount factor,

where the goal of the agent is to maximise the cumulative discounted reward.

When a human is operating the robot, the probability of successful task completion is dependent on unobservable factors such as their skill level or fatigue. Uncertain MDPs can be used to describe worlds where there is uncertainty over the exact transition probabilities. An **uncertain MDP** is defined by the tuple $\mathcal{M} = \langle S, \bar{s}, A, \{T_\theta\}_{\theta \in \Theta}, R, \gamma \rangle$, where the transition dynamics $\{T_\theta\}_{\theta \in \Theta}$ are governed by a set of global latent parameters, $\theta$, such that $T_\theta(s, a, s') = P(s' \mid s, a; \theta)$. The latent parameter $\theta \in \Theta$ is an $N-$dimensional vector, i.e. $\Theta = \mathbb{R}^N$. The agent has a prior distribution, $P_0(\theta)$ over the parameter space $\Theta$. As the agent observes the outcome of actions, the posterior distribution over the parameters is updated using Bayes' rule, $P_t(\theta) \propto P_0(\theta) \cdot P(\theta \mid h_t)$, where $h_t = s_0 a_0 s_1 \cdots s_t$ is the history of the agent's actions and observations.

A challenge in using uncertain MDPs is how to update the posterior distribution over the latent parameters. In our setting, we are interested in capturing the change in variance of the posterior distribution as the outcome of the human operator's action is observed. To reduce the computational complexity of this, we use a hidden parameter polynomial MDP (HPP-MDP) [10], a model that uses polynomials to represent the uncertain transitions, to model the shared autonomy system. The HPP-MDP allows us to maintain a closed-form representation of the posterior distribution over the latent parameters, and thus allows us to efficiently calculate the variance of the posterior distribution.

*Definition 3.2 (HPP-MDP).* An HPP-MDP is defined by the tuple $\mathcal{M} = \langle S, \bar{s}, A, \Theta, P_0(\theta), T_\Theta, R, \gamma \rangle$, where:

- $S$ and $\bar{s} \in S$, $A$, $R$ and $\gamma$ are as in the MDP definition;
- $\Theta = [0, 1]^N$ is the parameter space of a set of $N$ latent parameters. We denote elements of $\Theta$ as $\theta = (\theta_1, \cdots, \theta_N)$;
- $P_0(\theta)$ is the prior distribution over $\theta$, where $P_0(\theta) \in Pol(\theta)$ i.e. it is a polynomial function of $\theta$;
- $T_\Theta : S \times A \times S \to Pol(\theta)$ is the polynomial set of possible transition functions.

To be well formed, the transition functions must be valid for all $\theta \in \Theta$, such that

$$\sum_{s' \in S} T_\theta(s, a, s') = 1, \forall s \in S, a \in A, \theta \in \Theta. \quad (1)$$

The optimal policy maps the history of the trajectory the action with the highest Q-value, which is defined as:

$$Q^*(s, h_t, a) = R(s, a) + \max_{a \in A} \int_{\theta \in \Theta} \sum_{s' \in S} T_\theta(s, a, s') P_t(\theta) \gamma V^*(s', h_t a s') d\theta \quad (2)$$

where $V^*(s, h_t) = \max_{a \in A} Q^*(s, h_t, a)$ is the optimal value function.

Uncertain MDPs can be solved using a Monte-Carlo search tree based algorithm, such as the BAMCP algorithm [17]. BAMCP is a Monte-Carlo tree search algorithm which builds a history-dependent tree by sampling the prior distribution $P_0(\theta)$ to generate a sample MDP. The sample MDP used to simulate a run, and the trajectory taken is used to update the history-dependent nodes. BAMCP estimates the Q-value of each node, and the estimates converge towards the optimal Q-value as the number of samples increases. The optimal policy will give the optimal trade-off between exploratory actions to reduce the uncertainty over the human operator's performance, and exploitative actions that minimise the expected cost of the system.

# 4 SHARED AUTONOMOUS SYSTEMS WITH UNCERTAIN HUMAN OPERATORS

We first consider a shared autonomy system with one human operator and one robot, and then extend this to consider multiple robots and one human operator. In our shared autonomy system, the controller determines when and which robot the human operator should asisst, and determines the best action the robots and human should take.

## 4.1 Autonomous Robot Model

Autonomous robot $i \in [K] = \{1, \dots K\}$ has an environment described by the set of states $S_i$, and an initial state of $\bar{s}_i$. The behaviour of autonomous robot $i$ is known, so we can describe their behaviour with an MDP, $M_i = \langle S_i, \bar{s}_i, A_i, T_i, R_i, \gamma \rangle$. $A_i$ is the set of possible actions in the autonomous robot's domain, such as cardinal directions. $R_i(s, a)$ defines the reward associated with autonomous robot $i$ attempting action $a$ in state $s$.

## 4.2 Single-Robot Shared Autonomy Systems

In a shared autonomy system with a single robot and a human operator, the controller of the system must determine whether the human operator or the autonomy should perform the next action, while reasoning over the uncertainty of the human operator's performance. The autonomous robot $i$ is described by $M_i = \langle S_i, \bar{s}_i, A_i, T_i, R_i, \gamma \rangle$.

When the robot is operated by a human, we describe their performance with an HPP-MDP, $\mathcal{M}_{i,h} = \langle S_i, \bar{s}_i, A_{i,h}, P_0(\theta), T_{i,\Theta}, R_{i,h}, \gamma \rangle$. $A_{i,h}$ is the set of actions the human can take, and $P_0(\theta)$ is the controller's prior distribution over the human operator's latent parameters. The transition probability function $T_{i,\theta}(s, a, s')$ is a polynomial function of $\theta \in \Theta$, where $\Theta$ describes the space of all possible human performance. $R_{i,h}(s, a)$ is the reward function for the controller when a human attempts an action.

The aim of the controller of the system is to determine whether the human or the autonomy should operate the robot and the action they should take to maximise the expected cumulative reward of the system. As the controller does not know the true value of $\theta$, the controller must balance learning about the human operator's performance by requesting the human to attempt actions that will inform the posterior distribution $P_t(\theta)$, and using the current posterior distribution to choose actions that will maximise the expected reward of the next action. We use a HPP-MDP to model the shared autonomy system.

*Definition 4.1 (Single-Robot Shared Autonomy System).* The single-robot shared autonomy system for robot $i$ is described by the tuple $\mathcal{M}^i_{SA} = \langle S_i, \bar{s}_i, A^i_{SA}, P_0(\theta), T^i_{SA}, R^i_{SA}, \gamma \rangle$, where:

- The set of actions $A_{SA}$ is the union of the set of actions by the robot and the human operator, $A^i_{SA} = A_i \cup A_{i,h}$;
- The transitions $T^i_{SA}$ are defined by the following:

$$T^i_{SA}(s, a, s') = \begin{cases} T_{i,\theta}(s, a, s') & \text{if } a \in A_{i,h} \\ T_i(s, a, s') & \text{if } a \in A_i \end{cases} \quad (3)$$

  where the transition probabilities for the actions taken by the robot is fixed, while the transition probabilities for the actions taken by the human operator are governed by the history of the trajectory.
- The reward function $R^i_{SA}$ is the union of the reward functions for the autonomous robot and the human operator, such that

$$R^i_{SA}(s, a) = \begin{cases} R_{i,h}(s, a) & \text{if } a \in A_{i,h} \\ R_i(s, a) & \text{if } a \in A_i. \end{cases} \quad (4)$$

## 4.3 Multi-Robot Shared Autonomy Systems

We consider the problem where there are $K$ robots and one human operator, where there is uncertainty over how the human operator may perform. The controller must determine when to deploy human assistance and which robot to deploy the human to. Each robot has a separate set of tasks, so the set of robots operated by autonomy can be described as $\langle M_1, M_2, \dots, M_K \rangle$. Similarly, each robot can be operated by the human operator, so the set of robots operated by the human operator can be described as $\langle \mathcal{M}_{1,h}, \mathcal{M}_{2,h}, \dots, \mathcal{M}_{K,h} \rangle$. We model $K$ robot shared autonomy system with a HPP-MDP.

*Definition 4.2 (Multi-Robot Shared Autonomy System).* A shared autonomy system with $K$ robots is described by the tuple $\boldsymbol{\mathcal{M}}_{SA} = \langle \boldsymbol{S}_{SA}, \bar{\boldsymbol{s}}_{SA} \boldsymbol{A}_{SA}, P_0(\theta), T_{SA}, R_{SA}, \gamma \rangle$, where:

- $\boldsymbol{S}_{SA} = \bigtimes_{i=1}^{K} S_i$ is the state space;
- $\bar{\boldsymbol{s}}_{SA} = (\bar{s}_1, \bar{s}_2 \dots \bar{s}_K)$ is the initial state;
- The set of actions $\mathbf{A}_{SA}$ is defined as the subset of $(A_{1,h} \cup A_1) \times \cdots \times (A_{K,h} \cup A_K)$ for which there is at most one element corresponding to a human actions. More precisely, $(a_1, \cdots, a_K) \in \mathbf{A}_{SA}$ if and only if there is at most one $j$ for which $a_j \in A_{i,h}$ and for all other $k \neq j$, $a_k \in A_k$. Thus, there is at most one action done by the human operator in the joint action.

- For $\boldsymbol{a} = (a_1, \cdots a_K) \in \mathbf{A}_{SA}$, define $\boldsymbol{a}|_h$ as the index of the human operator action in $\boldsymbol{a}$, or $\perp$ if there is no human operator action in $\boldsymbol{a}$. The joint transition function $T_{SA}$ is defined as:

$$T_{SA}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s'}) = \begin{cases} \prod_{i \in [K]} T_i(s_i, a_i, s_i{}') & \text{if } \boldsymbol{a}|_h = \perp \\ T_{j,\theta}(s_j, a_j, s_j{}') \prod_{i \in [K]^{-j}} T_i(s_i, a_i, s_i{}') & \text{if } \boldsymbol{a}|_h = j \end{cases} \tag{5}$$

  where $[K]^{-j} = \{1, \ldots, j-1, j+1, \ldots, K\}$.
- The joint reward function $R_{SA}(\boldsymbol{s}, \boldsymbol{a})$ is the sum of the reward functions for each robot, $R_{SA}(\boldsymbol{s}, \boldsymbol{a}) = \sum_{i=1}^{K} R_i(s_i, a_i)$.

Solving the $\boldsymbol{\mathcal{M}}_{SA}$ will give the optimal policy that balances the robots' need for assistance and the need to learn about the human operator's performance.

# 5 BIDDING SYSTEM FOR SHARED UNCERTAIN OPERATOR

The shared autonomy system model with multiple robots, $\boldsymbol{\mathcal{M}}_{SA}$ has a state space $\mathbf{S}_{SA}$ that expands exponentially with the number of robots, $K$. Therefore, trying to solve this model directly is intractable. We instead propose a bidding system where each robot bids for the human operator's assistance. The bid describes how much the robot will benefit from the human operator helping them, given the robot's current state. We define this as the difference in the expected reward when the human operates the robot, and when the autonomy operates the robot. The bid for the $i$th robot in local state $s$ with a history $h_t$ is:

$$\text{Bid}_i(s) = \max_{a_h \in A_{i,h}} Q_{\mathcal{M}_{i,h}}(s, h_t, a_h) - \max_{a_i \in A_i} Q_{\mathcal{M}_{i,h}}(s, h_t, a_i), \tag{6}$$

where $Q_{\mathcal{M}_{i,h}}(s, h, a)$ is the Q-value defined in Equation 2 for the HPP-MDP $\mathcal{M}_{i,h}$. We will first outline the bidding system, then how the bids are generated.

## 5.1 Bidding System

In our bidding system, there is a centralised controller, and $K$ robots, where for robot $i$, $M_i$ and $\mathcal{M}_{i,h}$ respectively models the robot being operated by auto and the human. The robots maintain a shared prior distribution $P_0(\theta)$ over the human operator's latent parameter $\theta$. Each robot generates a bid for human help, and the controller instructs the human to assist the robot with the highest positive bid. The human will act according to $\arg\max_{a_h \in A_{i,h}} Q(s, h_t, a_h)$, and the other robots will act according to $\arg\max_{a_i \in A_i} Q(s, h_t, a_i)$. The robot receiving human help will observe the outcome of the human's action, and updates the posterior distribution $P_t(\theta)$ with the information gained during the interaction. This posterior distribution is shared with the other robots. This is because the system is cooperative, and the robots choose to share the observations of the human operator with the other robots to prevent asymmetry in the information available to each robots. Such asymmetry in information may result in sub-optimal bids from robots that do not have all the information available to them. This process is repeated until all robots have finished their tasks. A diagram of the bidding process is shown in Figure 1.

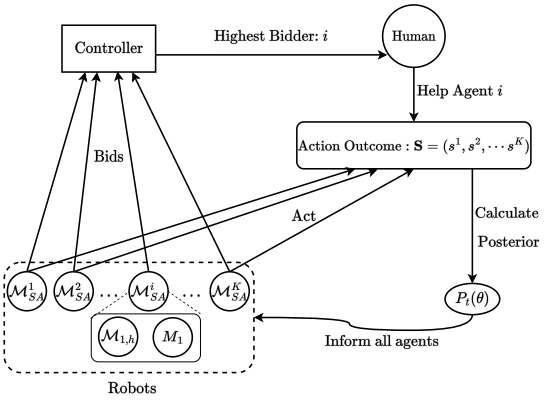



**Figure 1: Diagram of the bidding process.**

## 5.2 Bid Generation

For a given posterior distribution $P_t(\theta)$ over the latent variables, and a human operator described by $\mathcal{M}_h$, the robot $M_i$ can formulate a shared autonomy system $\mathcal{M}^i_{SA}$. The robot assumes that they have access to human help at any time, and only they are able to observe the human and update the posterior distribution. This shared autonomy system can be solved using the BAMCP algorithm to get the history-dependent Q-values. These Q-values would only consider the expected rewards collected by the robot, and not the expected rewards collected by the other robots. We can use these Q-values to generate a bid for the human operator using Equation 6. However, this does not consider 1) how the observations made by the robot can help increase the rewards collected by other robots, and 2) how the posterior distribution may change during a run due to observations made by others. We address these problems by making the following modifications to the BAMCP algorithm.

*5.2.1 Planning Horizon.* In a typical BAMCP algorithm, a search tree is built by sampling the prior distribution to generate a sample MDP. The sample MDP is then simulated until reaching a goal state, and the trajectory taken is used to update the history-dependent nodes. This is defined as a single trial. As the number of samples increases, the distribution of the sample MDPs reaching nodes will converge to the posterior distribution over the latent variables given the node's history. However, information gained by other robots during a run can change the posterior distribution, making simulating until the end of a run unrealistic, as it cannot account for external changes in the distribution.

We therefore use a short planning horizon, as the runs with longer planning horizons do not accurately reflect the posterior distribution. This allows for a shorter run time. At the end of the planning horizon, we use an heuristic value function $V_i(s)$ to estimate the expected reward collected from the state. The heuristic value function $V_i(s)$, where the robot assumes no help from the human operator, is found by using value iteration [24] to solve $M_i$. This expected reward is then back-propagated through the nodes. This allows us to simulate the run with a short planning horizon to consider the benefits of asking for human help, while still considering the long-term aim of reaching a goal state.

*5.2.2 Variance-Based Reward Bonus.* When an robot has a short planning horizon, the Q-value of an action does not reflect the true value of an informative action, as the exploitation that can occur in later transitions is not considered. An informative action is a human action that may not directly result in rewards, but will inform the posterior distribution over the latent variables. Furthermore, in a multi-robot system, there may be actions that can benefit other robots, but not the robot that is taking the informative action. The robot only considers their own state-action space to compute the Q-values of these actions, and thus will not consider the expected value of the action for other robots. This leads to the robot undervaluing informative actions. To address this problem, we explicitly reward actions that reduce the variance of the belief distribution over the human operator's behaviour by altering the reward function of the BAMCP algorithm. The new history-dependent reward function for robot $i$ is defined as:

$$R^i_{\text{new}}(s,h,a) = (1-\alpha)\cdot R^i_{SA}(s,a) + \alpha\cdot\left(\text{var}(\theta \mid h) - \int_{\theta\in\Theta}\sum_{s'\in S} T^i_{SA}(s,a,s')P_t(\theta)\text{var}(\theta \mid h,a,s')d\theta\right), \tag{7}$$

where $\alpha$ is the tuning parameter determining the weight of the variance term.

Calculating the variance using sample-based methods can be computationally expensive. The HPP-MDP reduces the computational complexity, as the posterior distribution over the latent variables is a polynomial function of $\theta$. We can express the posterior distribution at a node with history $h$ as:

$$P(\theta \mid h) = \sum_{j=1}^{J}\beta_j\prod_{i=1}^{N}\theta_i^{a_i^j}, \tag{8}$$

where $\beta_j$ is the coefficient of the $j$th term of the polynomial function, and $a_i^j$ is order of $\theta_i$ in the $j$th term of the polynomial function [10]. This can be used to calculate the variance of the posterior distribution at a node with history $h$ as:

$$\begin{aligned}\text{var}(\theta \mid h) &= \sum_{k=1}^{N}\left[\int_{\theta\in\Theta} P(\theta \mid h)\cdot\theta_k^2 d\theta - \left(\int_{\theta\in\Theta} P(\theta \mid h)\cdot\theta_k d\theta\right)^2\right]\\ &= \sum_{k=1}^{N}\left\{\sum_{j=1}^{J}\beta_j\left(\prod_{i\neq k}\frac{1}{a_i^j+1}\right)\cdot\frac{1}{a_k^j+3} - \left[\sum_{j=1}^{J}\beta_j\left(\prod_{i\neq k}\frac{1}{a_i^j+1}\right)\cdot\frac{1}{a_k^j+2}\right]^2\right\}.\end{aligned} \tag{9}$$

The closed-form expression for the variance allows us to efficiently calculate the variance of the posterior distribution at each node in the search tree. We use this to calculate the new reward function in Equation 7 when running the BAMCP algorithm.

# 6 EXPERIMENTS

We compare the performance of our bidding system to the joint HPP-MDP approach and a baseline where the system assumes to know the human's behaviour, proposed by Dahiya et al. [12]. The methods are compared on two domains, a sequential task and a robot navigation task. We show empirically that our bidding system can produce a similar performance to the joint HPP-MDP approach under fixed time constraints. We outperform the approach proposed by Dahiya et al. [12] when there is uncertainty over the human's performance.

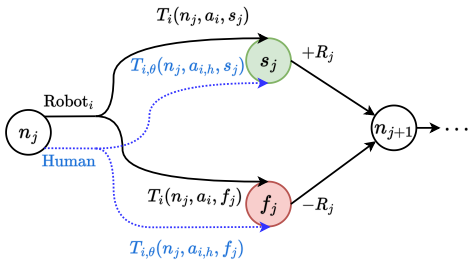



**Figure 2: The $j$th task for robot $i$ in the sequential task domain. The probability of robot $i$ entering the success state is known to be $T_i(n_j, a_i, s_j)$. The probability of the human entering the success state is $T_{i,\theta}(n_j, a_{i,h}, s_j)$, a function of $\theta$.**

## 6.1 Sequential Tasks

We use this domain to directly compare joint HPP-MDP approach to our bidding system, and analyse the impact of $\alpha$ on the total rewards collected. We use a small domain with two robots, as the joint HPP-MDP approach cannot scale to larger domains.

*6.1.1 Domain.* We consider a domain with two robots, where they each have 10 tasks to complete. The tasks must be completed in a sequential order, and the task will either result in success or failure. This will either yield a positive or negative reward. The robot can attempt the task, or they can request help from the human operator. The probability of success for the robot for any given task is known, while the human operator's probability of success is expressed as a polynomial function over a latent parameter, $\theta$. For example, the probability of success for task $j$ for the human operator could be expressed as $T_\theta(n_j, a^h, s_j) = 0.4 + 0.3\cdot\theta$, where $\theta$ is unknown to the robot. An example of a task is shown in Figure 2. We configure the tasks such that 80% of robot 1's tasks have low reward and high uncertainty over the human's success rate ( $T_\theta(n_j, a^h, s_j) = 0.1 + 0.9\cdot\theta$), while 20% of robot 2's tasks have high reward and high uncertainty over the human's success rate ($T_\theta(n_j, a^h, s_j) = 0.05 + 0.8\cdot\theta$). This configuration is designed to demonstrate how the information gained about the human by one robot can be beneficial to others, so acting independently does not result in overall higher rewards. The decay rate $\gamma$ is 1.0, and at the end of the 10 tasks, the robot will enter a zero-reward absorbing state.

*6.1.2 Algorithms.* We apply the following methods on the sequential task domain: Our **bidding system** with a planning horizon of 2 steps, and $\alpha \in [0, 1]$. The **joint HPP-MDP** model is solved with a BAMCP-based solver.

*6.1.3 Results.* The joint HPP-MDP solver took on average 10.5 minutes to run 150, 000 trials per decision step. Our bidding system

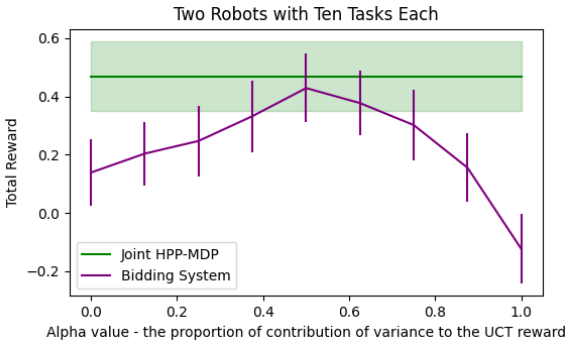


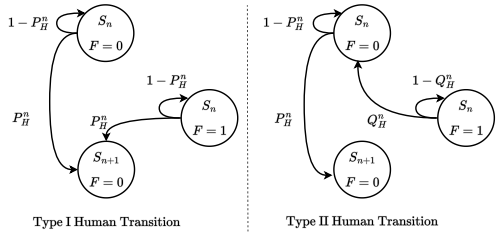

**Figure 3: Results for bidding system and joint HPP-MDP solver on the sequential tasks domain with $k = 2$ robots and $N = 10$ tasks.**

was run for 2 minutes per decision step. The results are shown in Figure 3. Our bidding system performs at an equivalent level to the joint HPP-MDP approach when $\alpha$ is between 0.4 and 0.6. As shown in Figure 3, the bidding system is sensitive to extreme values of $\alpha$, where low values of $\alpha$ result in a lack of informative actions, while high values of $\alpha$ result in a lack of exploitative actions. As the number of tasks increases, the time taken to solve the joint HPP-MDP model increases, while the time taken to solve our bidding system remains constant. For example, we found that a three robot domain with 10 tasks took approximately 193 minutes to run 150, 000 steps, making it impractical to use the joint HPP-MDP model to allocate the human operator. Therefore, in problems where there are longer horizons or a higher number of robots, the joint HPP-MDP model may not be able to solve the problem in a reasonable amount of time, while our bidding system can still produce a competitive solution.

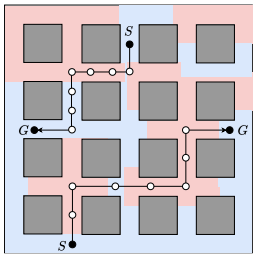
## 6.2 Robot Navigation

Dahiya et al. [12] proposes a framework for multi-robot shared autonomy, where the human operator can help the robot navigate to a goal. They used MDPs to model the robot and the human's behaviour, and each operator is modelled with a known probability of failure for each segment of the path to the goal. However, it is difficult to know how well the human can teleoperate the robot in a given environment. Therefore, a model with uncertainty over the human's performance may be able to better capture the true environment. We compare the performance of our bidding system approach to the approach proposed by Dahiya et al. [12] on a domain where the human's failure probabilities are unknown.

*6.2.1 Domain.* **Fixed Path Domain** Four robots are navigating in a city block-like randomly generated environment. The start and end location is also randomly generated, and the route planner finds a path. The path is split into 8 segments, where each segment is described as a single transition that can be done either by the robot or the human operator. In each segment, the autonomous robot can attempt the transition and may end in a fail state, where they cannot progress. Once in a fail state, a human must intervene to leave the fail state. There are two types of transitions possible, type-I and type-II transitions. In a type-I transition, the human operator




**(a) Type I and II transitions for human operators. $P_H^n$ is the probability of reaching the next state when the human controls the robot, and $Q_H^n$ is the probability the human can recover the robot from a fail state in type II transition.**



**(b) Example Domain with two robots. The regions with type I transitions are coloured in blue, type II transitions are coloured in red.**

**Figure 4: Robot Navigation Domain**

can go from a fail state to the end of the segment, while in a type-II transition, the human operator must return to the start of the segment before the autonomous robot can attempt the transition again. This is shown in Figure 4a. An example of the domain is shown in Figure 4b, where the type-I and II transition regions are shown in the blue and red areas. The decay rate for this domain was set to $\gamma = 1.0$, and the goal state was defined by a zero-reward absorbing state.

**Generalized Domain** Dahiya et al. [12]'s domain assumed the path was known to the robot, and the route was split into 8 segments independent of the start and end location. We generalize this domain to a five-by-five grid, where $k$ robots can move in the cardinal directions. The transition from adjacent states has an equal probability of being a type I or type II transition. The start and end location in the grid is randomly generated.

In both the fixed path and general domains, the human success probabilities are randomly generated by sampling from a continuous range of possible values. The algorithms cannot directly observe the human's success probabilities, but they are given a continuous range of possible values. We fix the reward of attempting the transition at $-2$, the reward of entering the fail state is $-4$, and the reward of the human helping the robot is $-0.75$.

*6.2.2 Algorithm.* **Fixed Path Domain** Dahiya et al. [12] proposes a decentralised method for human help allocation in a multi-robot shared autonomy system, where the robot with the highest need for human assistance is given human help. The robot determines the need for human assistance based on their MDP. The MDP outlines a direct path of eight transitions from the start to the end location and gives the probabilities of success for each transition for the human

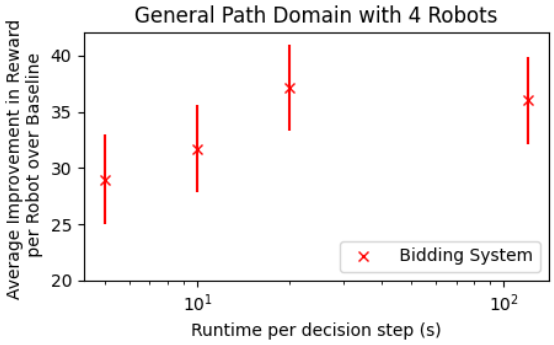



**Figure 5: The difference in average reward per agent between our bidding system and the baseline, Dahiya et al. [12].**

and autonomous operator. This is used to compute the Whitter Index [1] for each state. The index represents the additional human cost required to add into the reward function such that the expected total reward of the robot attempting the task is higher than when the human attempts the task. This involves an implicit assumption that the human will always outperform the robot, and the method assumes that the performance of the human is known. The robot with the highest Whitter index is given human help.

**Generalized Domain** In Dahiya et al. [12], the MDP is generated by placing seven waypoints between the start and end location, and the path planning between the two points is considered to be a separate problem. In this general domain, the path between the start and end location is found by solving the MDP where we only consider autonomous actions. This path is then used to generate a sequence of transitions from the start to the end location. Unlike the fixed path domain, the number of transitions is not fixed at eight.

**Bidding System** In both the fixed path and generalized domain, we apply our bidding system with a planning horizon that terminates once the robot reaches the next waypoint, and $\alpha = 0.99$. We set $\alpha$ high to balance the relative magnitudes of the first and second term in Equation 7. Every robot was given two minutes to compute their bid. As the algorithm was implemented in python, these computation times could be substantially improved. The two minutes bid computation time was fixed to ensure convergence of the policy. As shown in Fig. 5, the bidding system outperforms the baseline even at low computation times. In comparison, Dahiya et al. [12] took $15.2 \pm 4.8$ seconds to compute their static policy.

*6.2.3 Results.* **Fixed Path Domain** In the fixed path domain, under Dahiya et al. [12] the robots had an average reward of $-200.44 \pm 2.27$, while our bidding system had an average reward of $-132.07 \pm 1.29$. In a deadlock situation, where multiple robots submit the same bid, the robot with the lowest index is given human help, and this can be seen in Figure 6.

**Generalized Path Domain** We varied the number of robots in the domain from 2 to 8 and compared the average reward per robot for our bidding system and the method proposed by Dahiya et al. [12]. As shown in Figure 7, our bidding system outperforms the method proposed by Dahiya et al. [12] in all cases. The average reward per robot decreased as the number of robots increased for both approaches. This is because the human is being shared among more robots, and thus the human is less likely to be able to help an robot.

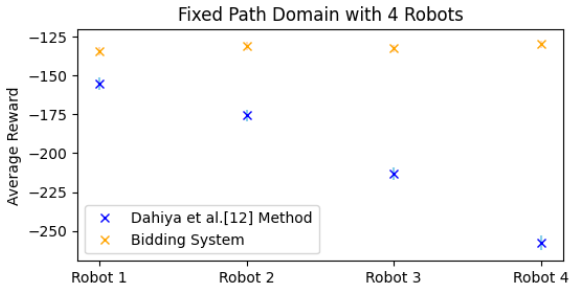



**Figure 6: The average reward of the robots in the fixed path domain, using our bidding system and the method proposed by Dahiya et al. [12].**

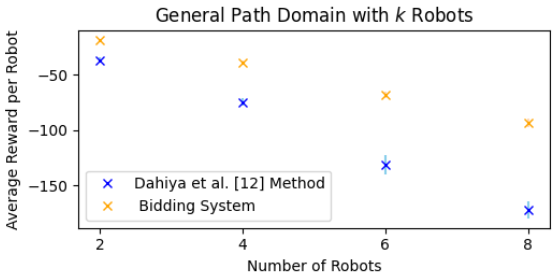



**Figure 7: Average reward per robot in the general domain, where the number of robots is varied.**

## 7 CONCLUSION

We have presented a efficient approach to allocating human assistance in a multi-robot shared autonomy system when there is uncertainty over the human operator's performance. Our online bidding system explicitly considered reducing the variance over the latent parameters governing the human operator. We demonstrated the computational infeasiblity of using a joint model for shared autonomy systems with large number of robots, and compared our approach to one where the uncertainty over the human's performance was not considered. In future work, we will consider systems with multiple human operators, and where the human operator's performance can change during execution.

## ACKNOWLEDGMENTS

This work was supported by the Defence Science and Technology Laboratory, the EPSRC Programme Grant 'From Sensing to Collaboration' (EP/V000748/1), the Clarendon Fund at the University of Oxford, and a gift from Amazon Web Services. This document is an overview of UK MOD's Defence Science and Technology Laboratory (DSTL) sponsored research and is released for informational purposes only. The contents of this document should not be interpreted as representing the views of the UK MOD, nor should it be assumed that they reflect any current or future UK MOD policy.

## REFERENCES


[1] Nima Akbarzadeh and Aditya Mahajan. 2022. Conditions for indexability of restless bandits and an O(K^3) algorithm to compute Whittle index. *Cambridge University Press* 54, Advances in Applied Probability (2022), 1164–1192. https://doi.org/10.1017/apr.2021.61 arXiv:2008.06111 [cs, eess, math].

[2] Sterling J Anderson, Steven C. Peters, Karl D. Iagnemma, and Tom E. Pilutti. 2009. A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE International Conference on Systems, Man and Cybernetics, 2032–2037. https://doi.org/10.1109/ICSMC.2009.5346330 ISSN: 1062-922X.

[3] Connor Basich, Justin Svegliato, Kyle Hollins Wray, Stefan Witwicki, Joydeep Biswas, and Shlomo Zilberstein. 2020. Learning to Optimize Autonomy in Competence-Aware Systems. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 123–131.

[4] Raphen Becker, Victor Lesser, and Shlomo Zilberstein. 2005. *Analyzing Myopic Approaches for Multi-Agent Communication*. Vol. 2005. IEEE/WIC/ACM International Conference on Intelligent Agent Technology. https://doi.org/10.1109/IAT.2005.44 Journal Abbreviation: Proceedings - 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'05 Pages: 557 Publication Title: Proceedings - 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'05.

[5] Craig Boutilier. 1996. Planning, Learning and Coordination in Multiagent Decision Processes. In *TARK '96: Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*. Morgan Kaufmann Publishers Inc., The Netherlands, 195–210. https://dl.acm.org/doi/10.5555/1029693.1029710

[6] Craig Boutilier and Tyler Lu. 2016. Budget allocation using weakly coupled, constrained Markov decision processes. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence (UAI'16)*. AUAI Press, Arlington, Virginia, USA, 52–61.

[7] Yifan Cai, Abhinav Dahiya, Nils Wilde, and Stephen L. Smith. 2022. Scheduling Operator Assistance for Shared Autonomy in Multi-Robot Teams. In *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, Cancun, Mexico, 3997–4003. https://doi.org/10.1109/CDC51059.2022.9993014 ISSN: 2576-2370.

[8] Jack-Antoine Charles, Caroline P. C. Chanel, Corentin Chauffaut, Pascal Chauvin, and Nicolas Drougard. 2018. Human-Agent Interaction Model Learning based on Crowdsourcing. In *Proceedings of the 6th International Conference on Human-Agent Interaction (HAI '18)*. Association for Computing Machinery, New York, NY, USA, 20–28. https://doi.org/10.1145/3284432.3284471

[9] Clarissa Costen, Marc Rigter, Bruno Lacerda, and Nick Hawes. 2022. Shared Autonomy Systems with Stochastic Operator Models. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 4614–4620. https://doi.org/10.24963/ijcai.2022/640

[10] Clarissa Costen, Marc Rigter, Bruno Lacerda, and Nick Hawes. 2023. Planning with Hidden Parameter Polynomial MDPs. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 10 (June 2023), 11963–11971. https://doi.org/10.1609/aaai.v37i10.26411 Number: 10.

[11] Murat Cubuktepe, Nils Jansen, Mohammed Alshiekh, and Ufuk Topcu. 2021. Synthesis of Provably Correct Autonomy Protocols for Shared Control. *IEEE Trans. Automat. Control* 66, 7 (July 2021), 3251–3258. https://doi.org/10.1109/TAC.2020.3018029 Conference Name: IEEE Transactions on Automatic Control.

[12] Abhinav Dahiya, Nima Akbarzadeh, Aditya Mahajan, and Stephen L. Smith. 2022. Scalable Operator Allocation for Multi-Robot Assistance: A Restless Bandit Approach. *IEEE Transactions on Control of Network Systems* 9 (Sept. 2022), 1397–1408. https://doi.org/10.1109/TCNS.2022.3153872 arXiv:2111.06437 [cs, eess].

[13] Frits de Nijs, Erwin Walraven, Mathijs de Weerdt, and Matthijs Spaan. 2017. Bounding the Probability of Resource Constraint Violations in Multi-Agent MDPs. *Proceedings of the AAAI Conference on Artificial Intelligence* 31, 1 (Feb. 2017). https://doi.org/10.1609/aaai.v31i1.11037 Section: Main Track: Planning and Scheduling.

[14] Francesco Duchetto, Ayse Kucukyilmaz, Luca Iocchi, and Marc Hanheide. 2018. Do Not Make the Same Mistakes Again and Again: Learning Local Recovery Policies for Navigation From Human Demonstrations. *IEEE Robotics and Automation Letters* PP (July 2018), 1–1. https://doi.org/10.1109/lra.2018.2861080

[15] Lu Feng, Clemens Wiltsche, Laura Humphrey, and Ufuk Topcu. 2016. Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems. *IEEE Transactions on Automation Science and Engineering* 13, 2 (April 2016), 450–462. https://doi.org/10.1109/TASE.2016.2530623 Conference Name: IEEE Transactions on Automation Science and Engineering.

[16] Anna Gautier, Marc Rigter, Bruno Lacerda, Nick Hawes, and Michael Wooldridge. 2023. Risk-Constrained Planning for Multi-Agent Systems with Shared Resources. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 113–121.

[17] Arthur Guez, Nicolas Heess, David Silver, and Peter Dayan. 2014. Bayes-Adaptive Simulation-based Search with Value Function Approximation. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/hash/839ab46820b524afda05122893c2fe8e-Abstract.html

[18] Kazi Mahmud Hasan, Abdullah-Al-Nahid, and Khondker Jahid Reza. 2014. Path planning algorithm development for autonomous vacuum cleaner robots. In *2014 International Conference on Informatics, Electronics Vision (ICIEV)*. 1–6. https://doi.org/10.1109/ICIEV.2014.6850799

[19] Emilie M. D. Jean-Baptiste, Pia Rotshtein, and Martin Russell. 2015. POMDP Based Action Planning and Human Error Detection. In *Artificial Intelligence Applications and Innovations (IFIP Advances in Information and Communication Technology)*, Richard Chbeir, Yannis Manolopoulos, Ilias Maglogiannis, and Reda Alhajj (Eds.). Springer International Publishing, Cham, 250–265. https://doi.org/10.1007/978-3-319-23868-5_18

[20] Sebastian Junges, Nils Jansen, Joost-Pieter Katoen, Ufuk Topcu, Ruohan Zhang, and Mary Hayhoe. 2018. Model Checking for Safe Navigation Among Humans. In *Quantitative Evaluation of Systems (Lecture Notes in Computer Science)*, Annabelle McIver and Andras Horvath (Eds.). Springer International Publishing, Cham, 207–222. https://doi.org/10.1007/978-3-319-99154-2_13

[21] C. Longjard, Prayoth Kumsawat, Kitti Attakitmongkol, and Arthit Srikaew. 2007. Automatic lane detection and navigation using pattern matching mode. In *Proceedings of the 7th WSEAS International Conference on Signal, Speech and Image Processing (SSIP'07)*. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 44–49.

[22] Amal Nanavati, Christoforos Mavrogiannis, Kevin Weatherwax, Leila Takayama, Maya Cakmak, and Siddhartha Srinivasa. 2021. Modeling Human Helpfulness with Individual and Contextual Factors for Robot Planning. In *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2021.XVII.016

[23] Frits de Nijs, Matthijs Spaan, and Mathijs de Weerdt. 2018. Preallocation and Planning Under Stochastic Resource Constraints. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (April 2018). https://doi.org/10.1609/aaai.v32i1.11592 Number: 1.

[24] Martin Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Ltd.

[25] Marc Rigter, Bruno Lacerda, and Nick Hawes. 2020. A Framework for Learning From Demonstration With Minimal Human Effort. *IEEE Robotics and Automation Letters* 5, 2 (April 2020), 2023–2030. https://doi.org/10.1109/LRA.2020.2970619

[26] Ariel Rosenfeld, Noa Agmon, Oleg Maksimov, and Sarit Kraus. 2017. Intelligent agent supporting human–multi-robot team collaboration. *Artificial Intelligence* 252 (Nov. 2017), 211–231. https://doi.org/10.1016/j.artint.2017.08.005

[27] Stephanie Rosenthal and Manuela Veloso. 2012. Mobile robot planning to seek help with spatially-situated tasks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*. AAAI Press, Toronto, Ontario, Canada, 2067–2073.

[28] Gokul Swamy, Siddharth Reddy, Sergey Levine, and Anca D. Dragan. 2020. Scaled Autonomy: Enabling Human Operators to Control Robot Fleets. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 5942–5948. https://doi.org/10.1109/ICRA40945.2020.9196792 ISSN: 2577-087X.

[29] Erwin Walraven and Matthijs T. J. Spaan. 2018. Column Generation Algorithms for Constrained POMDPs. *Journal of Artificial Intelligence Research* 62 (July 2018), 489–533. https://doi.org/10.1613/jair.1.11216

[30] Kyle Hollins Wray, Luis Pineda, and Shlomo Zilberstein. 2016. Hierarchical approach to transfer of control in semi-autonomous systems. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, New York, New York, USA, 517–523.

[31] Bo Wu, Bin Hu, and Hai Lin. 2017. Toward efficient manufacturing systems: A trust based human robot collaboration. In *2017 American Control Conference (ACC)*. 1536–1541. https://doi.org/10.23919/ACC.2017.7963171 ISSN: 2378-5861.

[32] Jianhui Wu and Edmund H. Durfee. 2010. Resource-Driven Mission-Phasing Techniques for Constrained Agents in Stochastic Environments. *Journal of Artificial Intelligence Research* 38 (July 2010), 415–473. https://doi.org/10.1613/jair.3004 arXiv:1401.3845 [cs].

[33] kirk A. Yost and Alan R. Washburn. 2000. *The LP/POMDP Marriage: Optimization with Imperfect Information*. Technical Report. NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF OPERATIONS RESEARCH. https://apps.dtic.mil/sti/citations/ADA487441 Section: Technical Reports.