Human-Multi-Robot Team Collaboration for Efficient Warehouse Operation

Ariel Rosenfeld†, Oleg Maksimov, Sarit Kraus, and Noa Agmon.

Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel 52900.
† Corresponding author, arielros1@gmail.com.

Abstract. Multi-robot systems have been successfully deployed in modern warehouses where they are required to move merchandise and equipment from one place to another. In such systems, the part of the human worker is often overlooked in the system's design. In this paper we use a novel approach of utilizing automated advising agents for assisting a human worker to better manage her time and workload while supervising a team of multiple robots in a warehouse environment. We introduce the *k-Myopic Advice Optimization (k-MYAO) Heuristic* which prioritizes the human's tasks in human-multi-robot team collaboration environments. This heuristic is an extension of the *1-MYAO* heuristic that was already found to be successful in the Search And Rescue (SAR) task in previous work. We exemplify the *k-MYAO*'s benefit in designing 2 automated advising agents $AID1$ and $AID2$ which are deployed in a simulated warehouse. Our proposed intelligent advising agents were evaluated through an extensive empirical study with 60 human operators, and showed significant improvements in the human-multi-robot team performance.

Keywords: Human-Robot Interaction, Human-Agent Interaction, Human-Multi-Robot Interaction, Multi-Robot Systems.

1 Introduction

Robotic warehouse systems such as Amazon Robotics (previously known as Kiva Systems) [8], AutoStore [25] and others have changed the way modern warehouses operate. Their *goods-to-man* approach, in which goods are transported by autonomous robots to the human worker station, has been reported to save money, reduce labor and increase productivity [7,10].

In multi-robot systems for warehouse operation, the part of the human worker is often assumed to be marginal. Two hidden assumptions are made in this case: the first is that the robots perform relatively smoothly, with the infrequent need for human intervention; the second is that the human worker is only required to perform a single task at any given moment. Reality, however, can be more complicated on both accounts. Today most robotic systems should be supported by a human whenever they cannot handle a situation autonomously. Namely, robots can malfunction. Furthermore, human warehouse workers may be occupied by numerous tasks at a time; for example, packing merchandise, refiling inventory and handling robot malfunctions.

In our previous work[1] [21], we presented an intelligent agent for bridging the gap between a human operator and a (potentially large) team of robots in Search and Rescue (SAR) tasks [13]. The agent provided advice for the operator as to which actions she should take using a simple *1-step lookahead* myopic heuristic, which is called the $MYAO$ heuristic. In addition, the agent acted as a smart buffer between the robots' requests and the human operator.

In this work we adapt and extend our approach to the Warehouse Operation Task [2], which is significantly different than the SAR setting used in [21]: First, the warehouse worker's actions are strongly connected and dependent; for example, the worker cannot pack a customer's order before attaining all of the requested merchandise. This property necessitates the identification of these connections and their consideration in the advice provision process. Second, the worker's actions require significantly more time and effort, making the order in which actions are executed much more important. Last, in most SAR settings, the operator is not aware of the remaining mission time, which imposes limitations on the planning process. Naturally, in warehouse settings, the working hours are known in advance.

To address these differences and improve the agent's decision-making process we introduce the k-$MYAO$ heuristic. The k-$MYAO$ heuristic is an extension of the $MYAO$ heuristic suggested in [21], allowing k-steps-look ahead advice provision.

For the evaluation of our proposed methodology we developed 2 agents, $AID1$ and $AID2$, which use the *1-MYAO* and *2-MYAO* heuristics respectively. We test the two agents in a small simulated warehouse environment, which we built using the Gazebo robot simulation toolbox[3], in which 10 autonomous robots are required to move merchandise and products to the packaging stations where a single human worker fills orders. Through this extensive human study, with 60 human participants, we provide the following contributions:

1. We show that subjects equipped with an advising agent significantly improve their performance in terms of the task's goals (filling customers' orders). In particular, in this study, subjects equipped with the $AID1$ agent filled significantly more orders compared to subjects using a non-advising agent (Silent) by up to 13% (on average).
2. We further show that the $AID2$ agent, which uses the *2-MYAO* heuristic, significantly outperforms the $AID1$ agent, which uses the *1-MYAO* heuristic, for mildly experienced subjects. That is, for subjects who have already experienced the system once in the past, the $AID2$ agent significantly outperforms the $AID1$ agent.
3. We also show that the acceptance rate of advice suggested by $AID2$ was significantly lower than for the advice suggested by $AID1$.

[1] A short video summarizing our previous work is available at `https://www.youtube.com/watch?v=mSh67zb0Zm4`

[2] A short video summarizing our current work is available at `http://www.youtube.com/watch?v=rC1a4c6Voco`

[3] `http://www.gazebosim.org/`

2 Related Work

Recent studies in human-multi-robot interaction have shown that human operators devote their resources in a sub-optimal manner, resulting in sub-optimal performance by the entire system. These studies include ground robots [31,32,2,28,12,21] and areal robots [14,4,16,19].

Several proposals have addressed this ineffectiveness by providing intelligent interfaces and interaction modes that have been shown to ease the human operator's burden and increase the system's performance [30,29,18,3,26]. In [24], the authors suggest a different approach, in which robots should request (and receive) help from humans for actions they could not have performed alone due to lack of capabilities. However, in the presence of many robots' requests, this approach could (potentially) overwhelm a human operator. To the best of our knowledge, other than our previous work [21], no work has addressed our proposed methodology of utilizing advising agents to assist a human operator to better manage a team of multiple robots in complex environments.

Advising agents have been investigated and deployed in different settings where humans are in need of assistance. These agents have been shown to enhance human positive behavior and decrease negative behavior. A few recent examples are [23,22,1]. Generally speaking, an advising agent is faced with an optimization problem which it should solve [20]. Unfortunately, in our settings, calculating optimal advice is intractable due to the exponential size of the state space and the high uncertainty induced by the environment, robots and humans.

Myopic search (also known as lookahead search) is one of the most widely used techniques for handling intractable decision optimization problems [17]. In myopic search, instead of searching through the entire relevant state space, an agent uses a local, *bounded depth* search tree of possible actions and reactions. The agent then chooses the action that maximizes its payoff in its search space. Agents deploying myopic search are especially prominent in human-agent negotiation [6], economical decision-making [9] and game playing [15]. In this work, we design and evaluate 2 myopic agents, $AID1$ and $AID2$, deploying 1-step-lookahead and 2-steps-lookahead myopic heuristics, respectively.

3 The Advice Optimization Problem

We consider a set of N robots engaged in a cooperative task. A single human operator, denoted by O, is in charge of supervising the robots as well as performing different domain specific actions, e.g., packing merchandise. The *state space* S consists of all information regarding the robots (e.g., robot location, battery capacity, operational status) and the task, which is domain specific. An instance of the state space is denoted by $s \in S$. The operator, O, can perform *actions* during the task, at any time t, from a predefined set—A. Note that O can choose to execute no actions, i.e., $NULL \in A$. The task takes place during a predefined time interval $\mathbb{T} = [0, T]$, where 0 denotes the beginning time of the task and T is its termination time.

Advice is defined as a possible action $a \in A$ suggested by an automated agent for the human operator to perform. The operator is not obligated to accept the agent's advice, namely, she does not have to perform the suggested actions.

In state s and time t, the operator, according to her abilities, endures a cost (usually in terms of time) for performing action a, denoted by $C_o(s, a)$. If a is *infeasible* in state s (defined by the domain characteristics), then $C_o(s, a) = \infty$. We assume that $C_o(s, NULL) = 0$. We refer to this cost function as the *operator model*.

The *transition model*, denoted by $P_o(s_1, a, s_2, C_o(s_1, a))$, provides the probability of reaching state s_2 given action a in state s_1 and the operator model, $C_o(s_1, a)$. Note that the transition from s_1 to s_2 given a is not instantaneous. Specifically, during the transition from s_1 to s_2 an intermediate state s' is experienced. s' may be the same as s_1 (i.e., during a's execution the world state does not change from s_1) or it could be different from s_1 (i.e., during a's execution a new state is experienced). We define the intermediate state $s' = Int(s_1, a, s_2, C_o(s_1, a))$. Namely, $Int(\cdot)$ is a function returning the intermediate state experienced when performing action a in order to transition from state s_1 to state s_2.

The *reward function*, $R_o(s)$, provides a real value representing the expected reward of state s, usually in terms of task fulfillment. For example, in a game setting, one can define $R_o(s)$ to be the expected number of points gained per one minute of game-play given the current state s.

Ideally, we would like the operator to execute the **optimal policy** $\pi_o^* : S \times \mathbb{T} \to A$ which maximizes the expected accumulative reward given $S, A, P_o(\cdot), Int(\cdot), R_o(\cdot)$ and $C_o(\cdot)$. Finding π_o^* naturally translates into a Markov Decision Process (MDP) [33] consisting of state space S, action space A, a transition model composed of P_o and Int and a reward function determined by R_o and C_o. However, calculating π_o^* is generally intractable due to the exponential size of the state space and the high uncertainty induced by the environment, robots and operators. This difficulty also stems from the complexity of many multi-robot problems, such as the NP-hard task allocation problem [11].

We consider a more tractable advice optimization problem which uses a myopic, k-steps-lookahead heuristic.

The *k-Myopic* Advice Optimization Heuristic

The k-Myopic Maximization[4] Advice Optimization (k-$MYAO$) Heuristic is defined as follows.

Given state $s \in S$, time t and $k \geq 1$, we define the $Value$ function capturing the expected cumulative future reward by using k pieces of advice, starting at state s at time t.

[4] The minimization form uses argmin instead.

$$Value(a, s, t, k) \quad (1)$$

$$= \begin{cases} 0 & \text{If } t \notin \mathbb{T} \\ \int_{s' \in S} P_o(s, a, s', C_o(s, a))(\int_t^{min\{t+C_o(s,a),T\}} R_o(Int(s, a, s', C_o(s, a))dt & \text{If } k \geq 1. \\ +\max_{a'} Value(a', s', t + C_o(s, a), k - 1))ds') & \\ \int_t^T R_o(s) & \text{Otherwise.} \end{cases}$$

Using Equation 1, k-MYAO heuristic offers advice a^* such that:

$$a^* = \text{argmax}_a Value(a, s, t, k) \quad (2)$$

In words, in state s at time t, the agent suggests the action a^* which maximizes (minimizes) the expected *k-steps-lookahead* reward.

Algorithm 1 solves the *k-MYAO* Heuristic for a given k.

Algorithm 1 k-Step Lookahead Advice Provision

Require: k, T
1: $s \leftarrow InitialState()$
2: $t \leftarrow 0$
3: **repeat**
4: $max \leftarrow -\infty$
5: **for each** $a \in A$ **do**
6: $expRwd \leftarrow Value(s, a, t, k)$ ▷ Eq. 2
7: **if** $expRwd > max$ **then**
8: $advice \leftarrow a$
9: $max \leftarrow expRwd$
10: $OUTPUT\ advice$
11: $Spin(k-millisec)$.
12: $s \leftarrow GetCurrentState()$
13: $t \leftarrow GetTime()$
14: **until** $t \geq T$

a^* is not simple to calculate for a large k due to the recursive definition of $Value(s, a, t, k)$. Namely, the time complexity of Algorithm 1 is exponential in k. Therefore, in this work we focus on the *1-MYAO* ($k = 1$) and *2-MYAO* ($k = 2$) heuristics which are relatively fast and easy to solve. In our evaluation (Section 4.2), we were unable to solve the *3-MYAO* ($k = 3$) heuristic in real-time despite using a state-of-the-art PC with 2 CPUs, each of 6 cores. Each core operates at 2.8 GHz. Therefore, in this study we focus on the *1-MYAO* and *2-MYAO* heuristics.

4 Warehouse Operation

In this section, we instantiate the k-MYAO heuristic (Section 3) for providing advice in a warehouse operation task. We will first describe the warehouse operation task and our simulation, followed by our agents' design.

4.1 The Warehouse Operation Task

In this study, we focus on a small warehouse acting as a fulfillment center (also known as a packing center) where there is one human worker and 10 mobile ground robots. The task is conducted in a simulated warehouse which we built using the Gazebo robot simulation toolbox. The warehouse consists of 10 mobile ground robots capable of transporting shelves from one place to another over the warehouse floor. There are 52 shelves in the warehouse, each consisting of between 1 and 3 different products, which the robots can transport to 8 packing stations. At a packing station, the human worker can unload products from the arrived shelves. See Figure 1 for a snapshot of the warehouse simulation.

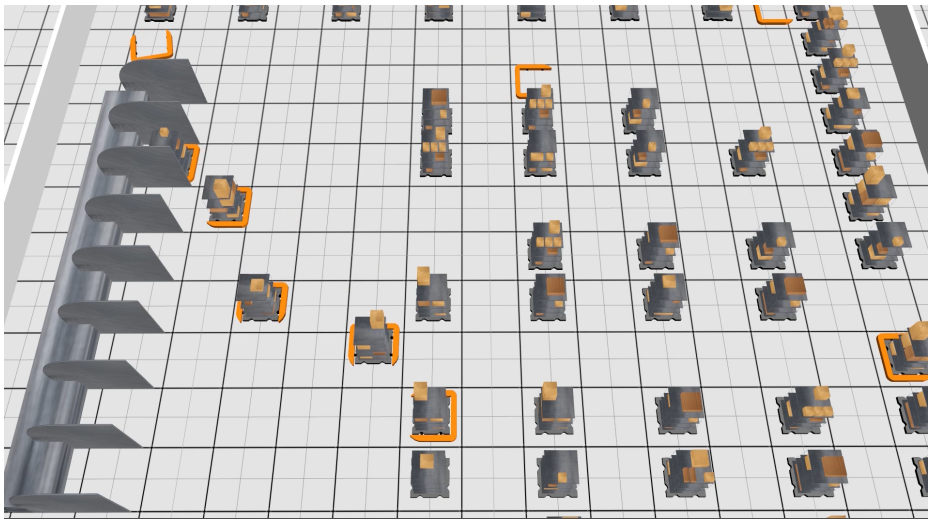

Fig. 1. Simulated Warehouse Environment.

When an order (a customer's requested set of products) arrives at the warehouse system, the system automatically sends the closest available robots to move the relevant shelves to the packaging stations. When a robot arrives at a packaging station, it is the human worker's job to remove the relevant merchandise from the shelf. Once the required products are unloaded from the shelf, the worker can send the robot off to move the shelf back to its original position and to continue its work. When an order has been filled, namely all requested products have been unloaded, the human worker can pack it. An order is removed from the system once it is packed.

During the robots' work, products may fall in the work area, potentially causing the robots to get stuck. Once a product has fallen on the warehouse floor, the human worker can alert the robots so that they can avoid the problematic

spot or she can remove the fallen product altogether. If the human worker decides to keep the robots from going through the problematic spot, then the robots will work out new, and possibly longer, paths. However, if the human worker decides to remove the product that fell on the warehouse floor, the robots will be restricted from approaching the area in order to avoid any danger to the human worker. This might delay the robots from completing their tasks. The robots can also malfunction regardless of fallen products. For example, a technical problem could cause one of the robots to deviate from its course. In such a situation the human worker will need to manually drive or guide the robot to its destination.

The GUI (Figure 2) provides real-time feedback on the task's state. When a new order arrives, it is automatically added to the active orders area, which presents all unpacked orders. The robots' and shelves' positions are marked on a 2D grid-map of the warehouse floor along with the robots' planned paths. An enlarged camera view of the robot of interest is available as well as a command panel for controlling that robot manually. In order to set interest on a specific robot, the operator can click its thumbnail within the thumbnails area or on its location on the grid map. The operator can view the robots' requests and provide solutions using the alarm log. For the operator's convenience, we maintain a first-come-first-serve alarm log indicating the (active) alarms.

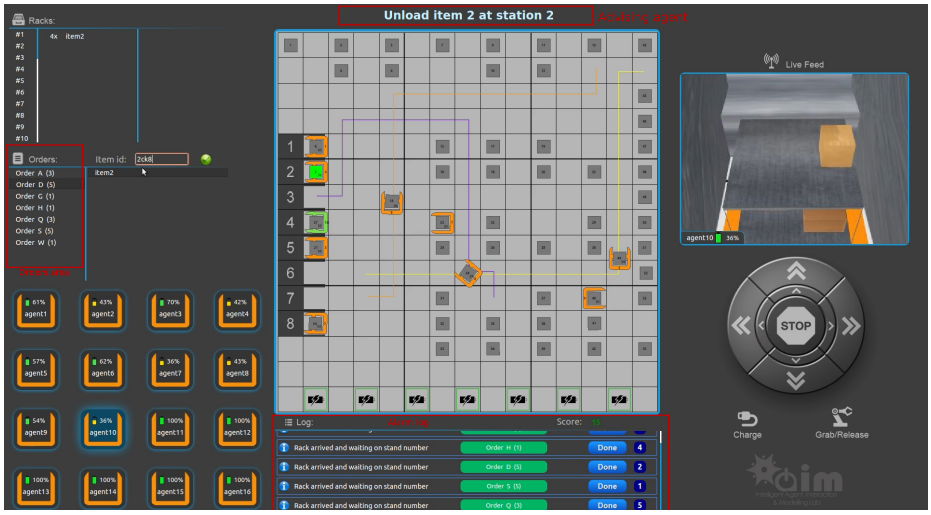

Fig. 2. The graphical user interface used in this study.

Overall, the human worker has to make complicated decisions in real-time while taking into consideration the active orders, the movement of the robots and her own personal task load.

4.2 The $AID1$ and $AID2$ Agents

We designed two agents, $AID1$ and $AID2$, which implement Algorithm 1 for solving the *1-MYAO* and *2-MYAO* heuristics, respectively. Recall that in order to correctly solve the optimization problem we need to estimate $C_o(\cdot)$, $R_o(\cdot)$, $P_o(\cdot)$ and $Int(\cdot)$.

The agents should be able to support the operator from its first interaction with the system. Therefore, we suggest using *generalized models*. Specifically, our agent uses three generalized models: an *operator model* to approximate $C_o(\cdot)$; a *robot model* to approximate $R_o(\cdot)$; a *transition model* to asses $P_o(\cdot)$ and an *intermission model* to capture $Int(\cdot)$. We will first define the generalized models and then describe their training.

The *operator model* quantifies the operator's abilities and technique in terms of expected time to execute *advice* a in state s. We use a general operator model $C(\cdot)$ independent of a specific O, and define $C(\cdot)$ as the expected time for completing a by an average operator.

The *robot model* quantifies the robots' productivity in terms of the expected number of fulfilled orders per minute. In order to collect data on robots' behavior and productivity, which can be very expensive as a human operator is needed to support the warehouse operation, we use a simulated *fully autonomous environment*. In our fully autonomous environment, there is no need for human interference. For example, once a shelf arrives at a packing station, the relevant products are automatically offloaded. Similarly, orders are automatically packaged once filled. Accordingly, we define $R(\cdot)$ as the expected number of orders to be filled in a *fully autonomous environment*. Note that our agents estimate $R(\cdot)$ based on a fully autonomous environment and utilize this estimation in order to generate advice in environments in which a human operator is needed.

We define $P(\cdot)$ as independent of O. Recall that $P(s, a, s', C(s, a))$ is the probability for transitioning from s to s' using action a, given $C(s, a)$. As S is a continuous set, we cannot assess the actual $P(\cdot)$, yet given s, a and $C(s, a)$ we can consider 2 options: 1) a was performed successfully and reached s'; 2) a was not performed or was unsuccessful and the robots reached state $s'' \neq s'$ (for example, the operator was unable to fix a malfunction). Due to the relatively short time frames in which pieces of advice are executed (an average action takes 6 seconds for the average operator), we assume that $s'' = s$.

For estimating $Int(s, a, s', C(s, a))$, which returns the intermediate state experienced in the process of transitioning from state s to s' using a, we used the help of an expert. An expert based function $S \times A \times S \rightarrow S$ was articulated, mapping each $< s, a, s' >$ tuple to an intermediate state s''. For example, when an obstacle falls on the warehouse floor (s) and the action $a = clean$ is performed, the state in which the robots cannot approach the area is experienced (s'') before reaching the state in which the obstacle has been removed (s').

Combining the above models, the $AID1$ and $AID2$ agents solve the *1-MYAO* and *2-MYAO* heuristics (respectively) using Algorithm 1 online at every second. Namely, the agents provide the best advice (per their optimization problem definition) to the user every second. The presented advice may change at time t if the

advice was successfully completed or when a more urgent and productive advice becomes available, making our agents *adaptive* to the changing environment. We used an efficient priority queue and updating methods to avoid redundant calculations.

Training The Models In order to train the above models, we used our warehouse simulation (see Figure 1). In our simulation Products fall on the warehouse floor and robots experience malfunctions according to a predefined schedule which determines when and where a failure will occur.

Recall that we defined $R(s)$ as the expected number of filled orders per minute given state s in a fully autonomous environment. Hence, to learn $R(\cdot)$, we ran 150 hours of sessions, each of 12.5 minutes, in our warehouse environment. During these sessions, the system receives 26 orders which are uniformly distributed over the 12.5 minutes. Two orders out of the 26 consist of 3 requested products, eight consist of 2 requested products and the remaining orders consist of a single requested product. The products requested in each order are selected at random. In order to learn the effect that fallen products have on the robots' productivity, we use between 0 and 3 products that are uniformly scheduled to fall on the warehouse floor during each session. As no human operator is present during the session, the fallen products caused robots to get stuck. Recall that the human worker can keep the robots from going through problem spots by marking these spots as "restricted spots". To quantify the effect such restricted spots have on the robots' productivity, during each session we set between 0 and 3 restricted spots at random on the map.

An obstacle or a restricted cell is considered *critical* if it prevents the completion of a task. Specifically, if a robot cannot arrive at its destination due to the existence of an obstacle or a restricted cell, it is considered critical.

Each recorded session was translated into 690 segments, each representing one minute starting at a different second (each session lasts 750 seconds). Each segment was translated into a feature vector according to the system state s in the beginning of the segment. Specifically, from each state s, we extract the number of inactive robots ($Inactive(s)$), the number of active orders ($Orders(s)$), the number of obstacles ($Obstacles(s)$), the number of critical obstacles ($CObstacles(s)$), the number of restricted cells ($Restricted(s)$) and the number of critical restricted cells ($CRestricted(s)$).

We estimated $R(s)$ using the following Equation:

$$\begin{aligned} R(s) = e^{-(\alpha_0 Restricted(s)+\alpha_1 CRestricted(s)+\alpha_2 Obstacles(s)+\alpha_3 CObstacles(s))} \\ \cdot \alpha_4 Orders(s) - \alpha_5 Inactive(s)^2 \end{aligned} \tag{3}$$

where $\alpha_0, \ldots, \alpha_5$ are learned using non-linear regression [27].

This form of function assumes that only a part of the active orders are eligible for packing during the next minute. This portion is then reduced by number and types of obstacles and restricted cells in an exponential manner. It also assumes that the squared number of inactive robots decreases the expected number of

filled orders to be completed in the next minute. This form of function was compared to more than 100 other forms and yielded the best fit to the data we collected[5]. All parameters are assumed to be positive, and in fact reached positive values.

To learn the operator's model $C(s, a)$ and the transition model $P(s, a, s', C(s, a))$, we first define A as the action set that the operator can take. Recall that A is also the advice set from which the agent can propose advice.

We defined A as the 518 instantiations of the following 6 action schemes: "Robot i is waiting for your command" (10 options), "Unload item x at station y." (480 options), "Complete the packing of order z." (26 options), "Clear the obstacle from the floor." (1 option), and "Obstacle was detected – restrict its cell." (1 option) where $i \in [1, \ldots, 10]$, $x \in [1, \ldots, 60]$, $y \in [1, \ldots, 8]$ and $z \in [A, B, \ldots, Z]$.

To train the models, we recruited 30 Computer Science senior undergraduate students, ranging in age from 21 to 39 (mean=26, s.d.=2.8) with similar demographics, 19 males and 11 females, to participate in a warehouse operation task equipped with a *naïve* advising agent. Our naïve advising agent maintains an open list of possible advice. The agent adds advice to the list once the advice is eligible, that is, the advice can be performed by the human worker. For example, once a shelf arrives at the packing station, the agent adds the mission "Unload item x at station y." according to the relevant item and station. The agent provides the advice at a first-come-first-serve fashion (i.e., naïvely). If the advice at the top of the list has become irrelevant or the advice was already performed by the operator then the advice is discarded and the next advice is used instead. The agent presents the advice in both textual format (see Figure 2) as well as in a prerecorded, human-voice message, played in the operator's head-set. Note that the agent *filters and prioritizes* robots' messages, yet in a naïve way.

For motivation, subjects were paid 1 NIS (about 25 cents) per order they pack, and 0.5 NIS per advice to which they adhere.

Surprisingly, all of the advice that our agents' gave was executed and completed. Therefore, we set $P(s, a, s', C(s, a))$ to be 1 whenever completing advice a is possible, and 0 otherwise. When analyzing the operators' behavior, we were unable to find good *cross-subject* features that would help us to predict the expected time it would take to complete the advice. Therefore, we used the mean execution time of a across our subjects as the estimation for $C(s, a)$.

Note that given personalized models, $C_o(\cdot)$, $P_o(\cdot)$ and $R_o(\cdot)$, our agent could also provide personalized advice. However, in the scope of this work these models are unavailable.

Evaluation In our previous work [21], we showed that an intelligent agent that supports the operator can lead to better performance of the human-multi-robot

[5] Some of the other function forms that were tested include other parameters that were found insignificant, such as the distance between the robots, the number of open packing stations, etc.

team in SAR tasks. In this section, we first extend this result to account for the warehouse operation tasks as well.

We first evaluate the $AID1$ agent compared to the condition in which no advising agent is deployed. We recruited 30 subjects, who did not participate in the data collection phase in Section 4.2, to participate in the warehouse operation task. Subjects were BSc and MSc Computer Science students, ranging in age from 21 to 39 (mean=26, s.d.=2.8) with similar demographics, 15 males and 15 females. Subjects were motivated to fill as many orders as possible during the task's 12.5 minutes using a small monetary reward (1 NIS per order). However, they did not receive any monetary reward for accepting the agent's advice. Each subject was trained prior to the experiment; she watched a short video explaining the task[6], underwent a structured 1-on-1 hands-on tutorial on the system by our research assistant and had to pass a short test. The test was conducted to make sure that the operator was capable of successfully completing the task. During the test, the subject had to fulfill 3 orders without any time limit while she encountered two simulated malfunctions.

Our preliminary results pointed out a significant improvement in operators' performance when performing the task for the second time. Therefore, we use a between-subjects experimental design where each subject performed the warehouse operation task twice (a week apart) and the subjects' results from the first session are disregarded. All 30 subjects were first asked to perform the task without the $AID1$ agent's help to get an understanding of the system. Then, a week afterwards, half of the subjects (15) were asked to perform the task once again without the $AID1$ agent's help while the other half were equipped with the $AID1$ agent. Only the subjects' second trial results were considered in the following analysis.

Results reported in this section were found to be significant using a standard t-test with $p < 0.05$.

The results show a statistically significantly higher average number of filled orders under the $AID1$ condition compared to the condition in which no agent was deployed. Subjects equipped with the $AID1$ agent averaged 24 packed orders (out of a possible 26) while subjects that were not assisted by an agent averaged 22 orders. See Figure 3 for a summary.

We further evaluate the $AID1$ agent, this time compared to the $AID2$ agent. We recruited 30 new subjects, who had not participated in the study thus far, to participate in the warehouse operation task. Again, subjects were senior BSc Computer Science students, ranging in age from 18 to 30 (mean=15, s.d.=2.2) with similar demographics, 20 males and 10 females. To compare the agents, each subject performed the task twice (a week apart); once equipped with the $AID1$ agent and once equipped with the $AID2$ agent. Subjects were counter-balanced as to which condition was applied first. The experiment protocol was the same as the one used above for comparing the $AID1$ agent to the no advising agent condition. Specifically, subjects' scores in their first sessions were disregarded.

[6] The tutorial video is available on `http://vimeo.com/152946192` (in Hebrew).

The results show that subjects equipped with the $AID2$ agent significantly fulfilled more orders compared to subjects equipped with the $AID1$ agent. Subjects equipped with the $AID1$ agent averaged 23.8 packed orders (out of a possible 26) while subjects equipped with the $AID2$ agent averaged 25 orders. See Figure 3 for a summary.

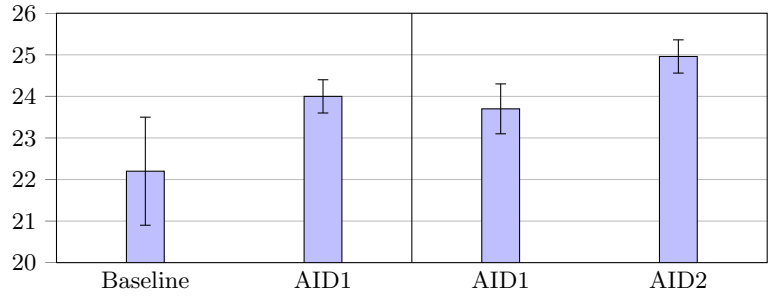

Fig. 3. Average number of filled orders per condition. Error bars indicate standard errors.

During each session, an average of 56.5 pieces of advice was presented to a subject. Advice is considered *accepted* if it is performed by the subject while it is displayed on the GUI. Surprisingly, subjects equipped with the $AID1$ agent significantly accepted more of the agent's advice than subjects equipped with the $AID2$ agent. Specifically, considering the subjects' second trial, subjects equipped with the $AID1$ agent accepted an average of 51.8 pieces of advice (91.6%), which is significantly higher than subjects equipped with the $AID2$ agent who accepted an average of 45.2 pieces of advice (80.6%), $p < 0.05$. Subjects' acceptance rate for an agent's advice is defined as the percentage of accepted advice by all subjects combined. Figure 4 presents subjects' acceptance rate per 60 seconds of trial for both $AID1$ and $AID2$.

5 Conclusions and Future Work

In this work, we showed that intelligent advising agents are able to significantly enhance the performance of operators in the multi-robot warehouse operation task. These results extend previous results showing the benefit of intelligent advising agents in the multi-robot SAR task. Our methodology can accommodate future advancements in robotics hardware and algorithms and is not restricted to a certain type or quantity of robots in the environment.

We also extended the MYAO heuristic, which considered only a 1-step-lookahead search heuristic, to account for k-steps-lookahead search. We showed that in our warehouse simulation the 2-steps-lookahead search, deployed by the $AID2$ agent, significantly outperforms the 1-step-lookahead heuristic deployed

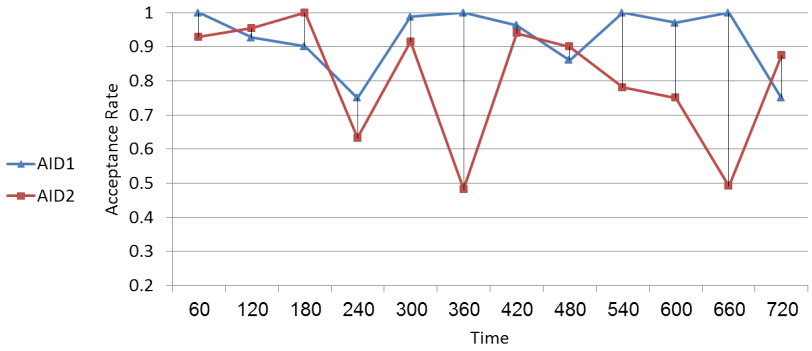

Fig. 4. Subjects' acceptance rate per 60 seconds for each of the tested agents. Results represent subjects' second trials.

by the $AID1$ agent and used in our previous study in SAR [21]. Note that in our experiments we were unable to run a 3-steps-lookahead heuristic in real-time due to the exponential search space. Our results indicate that subjects equipped with $AID2$ performed significantly better than those equipped with $AID1$. Nevertheless, the acceptance rate of advice suggested by $AID2$ was significantly lower than for the advice suggested by $AID1$. We hypothesize that in some cases subjects may not comprehend the reason for the suggested advice and therefore did not follow it. This property is disadvantageous as people are more likely to adhere to an agent's advice which they can understand [5].

We conclude that in complex environments, such as warehouse operation, k-steps-lookahead where $k > 1$ can enhance operators' performance significantly more than a simple 1-step-lookahead heuristic. However, additional factors such as the subjects' likelihood to accept different pieces of advice and the time-depth tradeoff have to be considered.

In future work we intend to examine our methodology's benefit in supporting larger teams of robots. Also, we intend to expand our methodology and design personalized agents. These agents could learn from previous interactions with the operator (if available) and estimate the subject's likelihood to accept different pieces of advice in order to tailor an advising policy for her.

References

1. Amos Azaria, Yaakov Gal, Sarit Kraus, and Claudia V Goldman. Strategic advice provision in repeated human-agent interactions. *Autonomous Agents and Multi-Agent Systems*, pages 1–26, 2015.
2. Jessie YC Chen, Michael J Barnes, and Caitlin Kenny. Effects of unreliable automation and individual differences on supervisory control of multiple ground robots. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 371–378. ACM, 2011.

3. Aaron St Clair and Maja Mataric. How robot verbal feedback can improve team performance in human-robot task collaborations. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 213–220. ACM, 2015.
4. ML Cummings, S Bruni, S Mercier, and PJ Mitchell. Automation architecture for single operator, multiple UAV command and control. Technical report, DTIC Document, 2007.
5. Avshalom Elmalech, David Sarne, Avi Rosenfeld, and Eden Shalom Erez. When suboptimal rules. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
6. Shaheen Fatima, Sarit Kraus, and Michael Wooldridge. *Principles of Automated Negotiation*. Cambridge University Press, 2014.
7. Fiona Graham. Logistics: Rise of the warehouse robots, December 2012. [Online; posted 18-December-2012].
8. Eric Guizzo. Three engineers, hundreds of robots, one warehouse. *IEEE spectrum*, 7(45):26–34, 2008.
9. Othar Hansson and Andy Mayer. The optimality of satisficing solutions. *CoRR*, 2013.
10. Eugene Kim. Amazon is now using a whole lot more of the robots from the company it bought for $775 million, October 2015. [Online; posted 22-October-2015].
11. G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512, 2013.
12. Michael Lewis. Human interaction with multiple remote robots. *Reviews of Human Factors and Ergonomics*, 9(1):131–174, 2013.
13. Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, 72(2):147–165, 2013.
14. C Miller. Modeling human workload limitations on multiple uav control. In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting*, pages 526–527, 2004.
15. Vahab Mirrokni, Nithum Thain, and Adrian Vetta. A theoretical examination of practical game playing: Lookahead search. In *Algorithmic Game Theory*, pages 251–262. Springer, 2012.
16. April Rose Panganiban. *Task load and evaluative stress in a multiple UAV control simulation: The protective effect of executive functioning ability*. PhD thesis, University of Cincinnati, 2013.
17. Judea Pearl. Heuristics: intelligent search strategies for computer problem solving. 1984.
18. Shokoofeh Pourmehr, Valiallah Mani Monajjemi, Seyed Abbas Sadat, Fei Zhan, Jens Wawerla, Greg Mori, and Richard Vaughan. You are green: a touch-to-name interaction in an integrated multi-modal multi-robot hri system. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 266–267. ACM, 2014.
19. S. D. Ramchurn, Joel E. Fischer, Yuki Ikuno, Feng Wu, Jack Flann, and Antony Waldock. A study of human-agent collaboration for multi-UAV. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1184–1192, 2015.
20. Ariel Rosenfeld. Automated agents for advice provision. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 4391–4392. AAAI Press, 2015.

21. Ariel Rosenfeld, Noa Agmon, Azaria Amos, Maksimov, Oleg, and Sarit Kraus. Intelligent agent supporting human-multi-robot team collaboration. In *Proceedings of the Twenty-Fouth International Joint Conference on Artificial Intelligence*, 2015.
22. Ariel Rosenfeld, Amos Azaria, Sarit Kraus, Claudia V. Goldman, and Omer Tsimhoni. Adaptive advice in automobile climate control systems. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 543–551, 2015.
23. Ariel Rosenfeld and Sarit Kraus. Providing arguments in discussions based on the prediction of human argumentative behavior. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1320–1327, 2015.
24. Stephanie Rosenthal and Manuela Veloso. Using symbiotic relationships with humans to help robots overcome limitations. In *Workshop for Collaborative Human/AI Control for Interactive Experiences*, 2010.
25. Jonas Lye Scheie, Jens Petter Røyseth, Preben Grøssereid, Lee Zheng, and Geir Fjermstad Rolandsen. Autostore, 2012. BachelorThesis.
26. Dominik Sieber, Selma Music, and Sandra Hirche. Multi-robot manipulation controlled by a human with haptic feedback. In *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE/RSJ, pages 2440–2446. IEEE, 2015.
27. Gordon K Smyth. Nonlinear regression. *Encyclopedia of environmetrics*, 2002.
28. PN Squire and R Parasuraman. Effects of automation and task load on task switching during human supervision of multiple semi-autonomous robots in a dynamic environment. *Ergonomics*, 53(8):951–961, 2010.
29. Atanasia Stoica, Theodoros Theodoridis, Huosheng Hu, Klaus McDonald-Maier, and David F Barrero. Towards human-friendly efficient control of multi-robot teams. In *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS)*, pages 226–231. IEEE, 2013.
30. Katia Sycara and Michael Lewis. Human control strategies for multi-robot teams. In *Proceedings of the 16th WSEAS International Conference on Computers*, pages 149–154. World Scientific and Engineering Academy and Society, 2012.
31. Prasanna Velagapudi and Paul Scerri. Scaling human-robot systems. In *CHI*, 2009.
32. Huadong Wang, Michael Lewis, Prasanna Velagapudi, Paul Scerri, and Katia Sycara. How search and its subtasks scale in n robots. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 141–148. ACM, 2009.
33. Chelsea C White III and Douglas J White. Markov decision processes. *European Journal of Operational Research*, 39(1):1–16, 1989.