

Adaptive Task Allocation in Human-Machine Teams with Trust and Workload Cognitive Models

Clémence Dubois and Jerome Le Ny

Abstract—In mixed-initiative systems where teams of humans and automated agents collaborate to perform decision-making tasks, determining factors of joint performance include human cognitive workload and the level of trust placed by the operators in the automation. Both workload and trust are dynamic variables that change over time based on current task allocation and on the result of past interactions. In this paper, we propose a methodology leveraging quantitative models of trust and workload to automatically and dynamically suggest efficient task allocations in mixed human-machine systems. Our approach is based on a Markov decision process framework and is presented for concreteness in the context of a human-machine team performing repeated binary decision-making tasks. Simulation results show the emergence of interesting automation behaviors such as seeking trust, attempting to repair trust after an error and adjusting human workload for optimal performance. Overall, the human-aware dynamic task allocation strategy shows the potential of significant team performance improvement compared to a static task distribution, even in the presence of significant errors in the trust and workload models used.

I. INTRODUCTION

Designing automated agents that collaborate effectively with humans is a challenging problem with a variety of applications, e.g., the monitoring of industrial control systems or of groups of unmanned vehicles. Various approaches for control sharing in human-machine teams have been proposed and are surveyed in [1], [2]. Adaptive automation, adjustable automation and mixed-initiative systems are different paradigms proposed to orchestrate human-automation interactions [1], [3], [4], but all recognize the importance of taking into account several human characteristics, such as mental workload and the tendency of operators to trust more or less their automation [5], [6]. In this paper, we propose a methodology to automatically and dynamically suggest task allocations between humans and automation, by leveraging quantitative models of workload and trust to anticipate and optimize the joint team performance.

An example of workload-aware dynamic task allocation strategy is given in [7], where tasks described abstractly by their service time are released for completion by a human in order to maintain an optimal mental workload. Another approach in [8] includes the varying nature of agent performance in a group role assignment problem and proposes a dynamic role allocation strategy. Here we consider more

specifically tasks that involve decision-making, either by a human or the automation, such as classifying objects or events based on noisy sensor data. Signal detection theory [6], [9], i.e., the application in psychology of tools from statistical decision theory to the human decision-making process, is used as a framework to model the impact of workload on human performance. Based on this model, we then develop a baseline task assignment strategy optimizing steady state joint performance of a human-automation team.

Another critical factor influencing team performance is conflict between partners. Human operators typically have authority over the automation and can override its commands or suggestions. For example, if s/he does not trust the automation's capability, the operator could prefer manual control. Lee and See [5] discuss how mistrust or over-trust in automation translate into under- or over-reliance, impacting joint performance. Several factors can impact how trust evolves during a mission, e.g., automation performance, transparency [10] or cultural factors [11]. Despite this complexity and the difficulty of measuring trust online (through behavior, questionnaires or physiological measurements), dynamic models can make reasonable predictions. Quantitative models of trust dynamics, as proposed in [12]–[16], enable the design of trust-aware automation [16]–[18], i.e., automation that adapts its behavior to the current level of trust of the operator. Here we use a simplified version of the state-space model of trust dynamics of Gao and Lee [13], which has indirectly inspired task allocation strategies in [19], [20]. An advantage of such a model is its small number of parameters, which moreover can be interpreted, and hence it requires much less experimental data for calibration than black-box models such as neural networks.

Given the dynamic nature of the trust model, our task allocation problem is formulated as a Markov Decision Process (MDP). The MDP framework is particularly well suited when the state dynamics is stochastic and uncertain, which is the case for cognitive variables such as trust and workload. MDPs and their extension with partial state observations (POMDPs) are used for human modeling and the design of human-automation collaboration strategies in [16], [18], [21] for instance. Defining the allocation problem as an MDP requires modeling the human behavior (through trust and workload) and the environment. Depending on the scenario, defining and calibrating such models can be difficult. This motivates us to explore the compromise between using a simpler model leading to our baseline static policy or a more complex dynamic model incorporating both workload and trust, which leads to a dynamic adaptive task allocation

This work was supported by the Canadian Department of National Defence under the Innovation for Defence Excellence and Security (IDEaS) Micro-Net program.

C. Dubois and J. Le Ny are with the department of Electrical Engineering, Polytechnique Montreal and GERAD, QC H3T-1J4, Montreal, Canada. clemence.dubois, jerome.le-ny@polymtl.ca

strategy but would be more difficult to calibrate accurately. The simulation results for the MDP show the emergence of interesting automation behaviors that could be interpreted as seeking trust, repairing trust after an error and adjusting workload. Moreover the dynamic policy seems to bring real benefits compared to the static task distribution ignoring trust, even in the presence of significant modeling errors.

The organization of the paper and a summary of our contributions are as follows. Section II describes a general scenario with repeated decision-making tasks to be performed either by a human or automatically. Section III develops a model of human performance capturing the impact of workload, together with a method for statically assigning an optimal workload level. Section IV presents the dynamic trust model, which is then included with the workload model in an MDP framework in Section V. Finally, in Section VI we compare through simulations the static and dynamic policies, including the impact of modeling errors on performance.

II. SCENARIO DESCRIPTION

We focus on a team comprised of a human operator and an automated system performing a sequence of decision-making tasks. We assume that each new task is independent of the previous ones and requires a binary decision, made either by the human or the automation. Formally, each task can be viewed as a hypothesis testing (or classification) problem, where the decision-maker must distinguish for task k between hypothesis $H_{0,k}$ (the null hypothesis) and $H_{1,k}$ (an interesting event), based on evidence Y_k , a random variable. Let $D_k \in \{0, 1\}$ denote the decision for task k , with $D_k = 1$ corresponding to “deciding $H_{1,k}$ ”.

Whether the decision-maker is the human [9] or the automation [22], it can be abstracted by its resulting probability of true positive $P_{TP} := P(D_k = 1|H_{1,k})$ and false positive $P_{FP} := P(D_k = 1|H_{0,k})$, assumed here independent of time since the environment is stationary. We associate rewards to decisions, as follows. For a successfully performed task (D_k is a true positive or true negative) a reward R^1 is collected. When an error occurs (false positive or false negative), a value $R^0 < R^1$ is collected, which can be negative. A negative “reward” $R^m \leq 0$ is added when the task is completed by the human, to capture situations where the human operator would preferably focus on other tasks. The joint performance of the team is measured by the sum of the collected rewards over a sufficiently long horizon.

We divide time into periods of $N \in \mathbb{N}$ successive tasks. At the beginning of each period, the automation suggests a task distribution, i.e., the proportion of tasks to be performed by the operator during the period. Then, s/he can choose to follow the recommended assignment or to perform manually more or fewer tasks, e.g., because her/his trust in the capabilities of the automation to perform the tasks is low or high. We are looking for strategies that suggest task distributions at each period in order to maximize the overall team performance.

III. WORKLOAD-AWARE PERFORMANCE MODEL

In this section, we first present an analytical model capturing the impact of workload on steady-state performance for binary decision-making tasks performed by a human. This model is then used to propose a static workload-aware task assignment strategy that will serve as benchmark.

A. Workload-Performance Model

By considering both automation and human operator as binary decision makers, their Receiver Operating Characteristic (ROC) curves capture the relation between the true positive rate P_{TP} and the false positive rate P_{FP} [9], see Fig. 1 for an example. The automation is assumed to have a fixed and known ROC curve, whereas the human’s one is parametrized by the workload. Hence, we model the human performance by a family of ROC curves, which have to be determined.

Although relatively arbitrary ROC curves could be obtained from experimental data, it is useful to adopt a family of curves shaped by few parameters in order to simplify the model calibration and interpret the effect of workload. In particular, a central model of signal detection theory, often used to study human decision under uncertainty, is the equi-variance Gaussian model [9, Chapter 2]. In its simplest interpretation, it corresponds to the problem of deciding if $Y_k \sim \mathcal{N}(0, 1)$ (hypothesis H_0) or $Y_k \sim \mathcal{N}(d, 1)$ (hypothesis H_1), where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2 and $d \geq 0$ is called the sensitivity. However, this interpretation can be abstracted to retain only the ROC curves, which in this case take the form

$$P_{TP}(P_{FP}) = \Phi(d + \Phi^{-1}(P_{FP})), \quad (1)$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$ is the cumulative distribution function (cdf) of a standard normal distribution $\mathcal{N}(0, 1)$ and d can be interpreted as the intrinsic discrimination capability of the decision maker. Cognitive workload, denoted $W \in [0, 1]$, reduces this capability in general, as it reduces the time to study the evidence. One possible model is to assume that the sensitivity of a human decision-maker is

$$d = d(W) = d_0(1 - W). \quad (2)$$

In this case, there is only one parameter $d_0 \geq 0$ to identify experimentally, which is the maximum sensitivity of the human when the workload tends to zero and also the rate at which the sensitivity decreases with W . With (1)-(2), increasing W brings the ROC curve closer to the diagonal, which corresponds to a random classifier, choosing between hypotheses with equal probability (flipping a coin).

B. Static Workload-Aware Task Assignment

Once a family of workload-dependent ROC curves is chosen, we can formulate a task allocation problem. Here we assume that the false positive rate P_{FP} allowed for the task is fixed, communicated to the human operator and included in the automated classifier’s algorithm. For many tasks, this requirement can be fulfilled by following threshold-based decision rules, which are naturally implemented by humans.

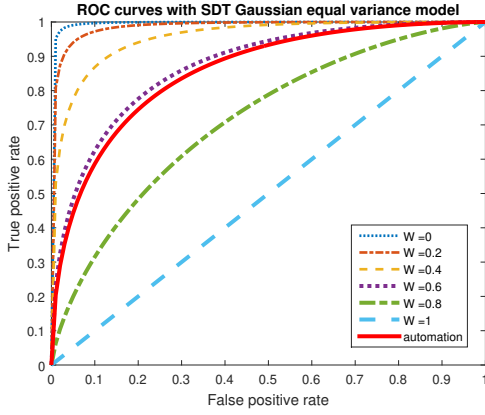


Fig. 1. A possible family of ROC curves for a human classifier, based on the equal variance Gaussian model. Here the human ROC curves are drawn with (1) and (2) for $d_0 = 4$. The automation is also associated to a ROC curve drawn with (1) where $d^a = 1.5$.

We denote P_s^h (resp. P_s^a) the correct classification probability of the human classifier (resp. the automation), i.e.,

$$\begin{aligned} P_s^h(W) &= pP_{TP}^h(P_{FP}; W) + (1-p)(1-P_{FP}), \\ P_s^a &= pP_{TP}^a(P_{FP}) + (1-p)(1-P_{FP}), \end{aligned} \quad (3)$$

where $p := P(H_{1,k})$, the probability of having an “interesting” event, is known. Since P_{FP} is fixed, we omit it as an argument of $P_{TP}^{a/h}$, and P_{TP}^h becomes a function of W only. To define a static workload-aware task distribution strategy, we find the workload W^* (i.e., the proportion of tasks assigned to the human) maximizing the overall reward

$$\begin{aligned} \max_{W \in [0,1]} & \left((1-W)[R^1 P_s^a + R^0(1-P_s^a)] + \right. \\ & \left. W[R^m + R^1 P_s^h(W) + R^0(1-P_s^h(W))] \right). \end{aligned} \quad (4)$$

For the scenario of Section II, the static policy simply assigns W^*N tasks to the human operator and $(1-W^*)N$ tasks to the automation at the beginning of each period.

After expanding (4), the problem can be written as

$$\begin{aligned} \max_{W \in [0,1]} & f(W), \quad \text{with} \\ f(W) &= W[p(R^1 - R^0)(P_{TP}^h(W) - P_{TP}^a) + R^m]. \end{aligned} \quad (5)$$

The following theorem gives conditions under which the human operator is beneficial to the team, i.e., under which the optimal workload is not zero.

Theorem 1: Fix a probability of false positives P_{FP} for the tasks. If P_{TP}^h is a continuous function of W , the function f in (5) reaches a maximum at some $W^* \in [0, 1]$. Moreover, assume that $R^0 < R^1$, $p > 0$, $R^m \leq 0$ and P_{TP}^h is differentiable at $W = 0$. Then we must have $W^* > 0$ if

$$P_{TP}^h(W=0) - P_{TP}^a > \frac{|R^m|}{p(R^1 - R^0)}, \quad (6)$$

and otherwise $W^* = 0$. Moreover, if $P_{TP}(W=1) < P_{TP}^a$, or if $P_{TP}(W=1) = P_{TP}^a$ and $R^m < 0$, then $W^* \in (0, 1)$.

Condition (6) implies that with a human intervention cost $R_m < 0$, the human operator’s best performance (when the

workload is low, i.e., $W = 0$) has to be sufficiently higher than the automation’s capability in order to benefit from the human assistance, increasingly so as $|R_m|$ increases.

Proof: To simplify the notation, we can rewrite (5) as

$$f(W) = \alpha WP(W) - \beta W$$

with $\alpha := p(R^1 - R^0) > 0$, $\beta := p(R^1 - R^0)P_{TP}^a - R^m$ and $P(W) := P_{TP}^h(W)$. The function f is a continuous real valued function, hence attains its maximum on the closed interval $[0, 1]$. Moreover,

$$f'(W) = \alpha(P(W) + WP'(W)) - \beta.$$

If $f'(0) > 0$, then the maximum of f in $[0, 1]$ is not zero. Noting that $f'(0) = \alpha P(0) - \beta$, this gives (6). Moreover, we have $f(0) = 0$ and $f(1) = \alpha P(1) - \beta = \alpha(P(1) - P_{TP}^a) + R^m$. The second part of the theorem corresponds to $f(1) < f(0)$, in which case $W^* \in (0, 1)$. ■

C. Computing W^* for the Equal Variance Gaussian Model

The following proposition ensures that finding numerically a global optimum W^* for (5) (by dichotomy or by solving $f'(W) = 0$) can be done efficiently for the equal-variance Gaussian model.

Proposition 1: Assume that the ROC curves are given by (1) with workload model (2). Under the conditions of Theorem 1, f has a unique local maximum W^* in $(0, 1)$ and W^* is the unique solution of $f'(W) = 0$.

Proof: Using the notation of the proof of Theorem 1, from (2) and (1) we have $P(W) = \Phi(-d_0W + K)$, for $K = d_0 + q$ and $q = \Phi^{-1}(P_{FP})$. We know from the previous proof that $f'(0) > 0$. Now we can show that $f'(1) < 0$:

$$f'(1) = \alpha(P(1) + P'(1)) - \beta = f(1) + \alpha P'(1)$$

We have already shown $f(1) < 0$ and $P'(1) = -\frac{d_0}{\sqrt{2\pi}}e^{-\frac{d_0^2}{2}} < 0$, hence $f'(1) < 0$. Moreover f' is differentiable and

$$\begin{aligned} f''(W) &= \alpha(2P'(W) + WP''(W)) \\ &= \frac{d_0\alpha}{\sqrt{2\pi}} \exp\left(-\frac{(-d_0W + K)^2}{2}\right) [d_0^2W^2 - Kd_0W - 2] \\ &=: g(W)(d_0^2W^2 - Kd_0W - 2) \end{aligned}$$

with $g(W) > 0$ always. The polynomial function $l(W) = d_0^2W^2 - Kd_0W - 2$ has two real roots W_1 and W_2 (with $W_1 < W_2$) because of its discriminant $\Delta = d_0^2(K^2 + 8) > 0$. We have $W_1 < 0$ as $l(0) < 0$ and the coefficient d_0^2 is positive. Hence either f'' is always negative and so f' is decreasing (if $W_2 > 1$), or f'' is negative until W_2 and then positive (if $W_2 \leq 1$) and so f' is decreasing and then increasing. But since $f'(0) > 0$ and $f'(1) < 0$, in both cases necessarily there must be a unique root of $f'(W) = 0$. ■

IV. MODELING HUMAN TRUST IN AUTOMATION

For the type of decision-making scenarios described in Section II, the automation would only make suggestions about the task distribution at each period, and the final control would remain to the human operator. In that case, one

can expect that s/he would decide to complete manually more tasks (resp. fewer tasks) than suggested by the automation if s/he does not trust (resp. overtrusts) its capabilities. To avoid such sub-optimal situations, we consider the trust dynamics in the design of the task allocation strategy.

Gao and Lee [13] proposed a quantitative model of operator reliance on automation, based on the level of trust in the automation’s capabilities. The model captures the impact of the history of the human-machine interactions and the dynamics of the underlying psychological process. Reliance, i.e., the decision to use automated control over manual control, is seen as a function of the difference between trust and self confidence. Here, for simplicity, we include only trust in our model of reliance, however it is possible to extend the model with the self-confidence dynamics.

The human trust state is represented by two variables denoted $x_t^h = (T_t, B_t)^T \in \mathbb{R}^2$, where T_t represents the trust level, B_t the belief in the automation capability and t denotes the period index. In other words, we update the trust dynamics only every N tasks, as described in Section II. These dynamics are influenced by two input variables, the environment state x_t^e and the automation action a_t^a , as follows

$$B_{t+1} = B_t + \eta (C(x_t^e, x_t^h, a_t^a) - B_t) + w_t^B \quad (7)$$

$$T_{t+1} = (1 - \mu)T_t + \mu (B_t + \eta (C(x_t^e, x_t^h, a_t^a) - B_t) + w_t^B) + w_t^T. \quad (8)$$

Here $\mu \in [0, 1]$ is a growth-decay parameter representing the inertia of trust, so that T_{t+1} is influenced by both T_t and B_{t+1} . The parameter $\eta \in [0, 1]$ represents the “transparency” of the automation interface, i.e., how well the true capability of the system is conveyed to the user. The two sequences $w_t^h = [w_t^B, w_t^T]^T$ are independent and identically distributed (iid) random variables with mean 0 and standard deviation σ_B and σ_T . Finally, $C(x_t^e, x_t^h, a_t^a) \in \mathbb{R}$ represents the true capability of the automation during the period t , which is a function of the current environment state, the human state and the automation action, and can be interpreted as a success rate, the precision demonstrated by the automation when completing its task, etc. The definition of C depends on the application and can be a delicate issue requiring preliminary experimental calibration studies, because it assumes that the human sees the automation’s capability as a one-dimensional variable. In our scenario, C could depend simply on the automation’s correct classification rate, or could also include the automation’s ability to suggest a “good” task distribution, see Sections V and VI for examples. In addition to the function C , calibration of this model requires identifying the four parameters $\mu, \eta, \sigma_B, \sigma_T$, based on experimental data. Calibration is, to some degree, user dependent. In [13], the model demonstrates good prediction performance but some errors can be attributed to individual behavioral differences.

V. DYNAMIC TASK ALLOCATION INCLUDING WORKLOAD AND TRUST

The dynamic and uncertain nature of trust as a cognitive state variable leads us to formulate the trust-aware task

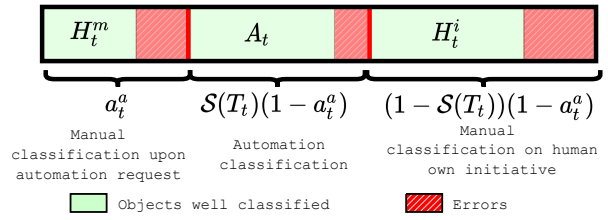


Fig. 2. Distribution and outcomes for tasks during period t .

allocation problem in the form of a Markov Decision Process (MDP), integrating the workload model of Section III as well. For simplicity, we consider here a situation where the human operator cannot perform fewer tasks than suggested by the automation but can only increase her workload.

At the beginning of period t , the automation suggests that a fraction $a_t^a \in [0, 1]$ of the N tasks be carried out by the human. This decision is allowed to depend on the state up to period $t - 1$. During this period and up to the beginning of period $t + 1$, the cognitive state of the operator is denoted $x_t^h = [T_t, B_t]^T$, which follow the dynamics (7)-(8), and we assume that the human actually performs a fraction

$$W_t := a_t^a + (1 - a_t^a)(1 - \mathcal{S}(T_t)) \quad (9)$$

of the tasks during period t , where \mathcal{S} is an increasing function mapping trust $T_t \in \mathbb{R}$ to $(0, 1)$ and assumed to be a sigmoid $x \mapsto 1/(1 + \exp(-5(x - 0.5)))$. In other words, an operator with a higher level of trust takes a smaller share of the fraction $(1 - a_t^a)$ of tasks initially assigned to the automation.

We define W_t in (9) to be the workload of the operator during period t . This workload affects the performance $P_s^h(W_t)$ of the operator during period t , according to the model (3). We also define A_t , the number of correct decisions made by the automation during period t divided by N , while H_t^m and H_t^i are similar ratios for the number of correct decisions made by the human for tasks: i) suggested by the automation; and ii) taken on her own initiative. We have

$$A_t = \mathcal{S}(T_t)(1 - a_t^a)P_s^a, \quad H_t^m = a_t^a P_s^h(W_t) \\ H_t^i = (1 - \mathcal{S}(T_t))(1 - a_t^a)P_s^h(W_t).$$

Fig. 2 provides a graphical representation of these different quantities. The overall system state is simply x_t^h , i.e., in this scenario there is no external environment state to remember. This state is assumed to be perfectly observable by the automation but with one period delay, in order to make the new recommendation $a_{t+1}^a \in [0, 1]$. This is the case in scenarios where the correctness of decisions can be evaluated online. Since \mathcal{S} is one-to-one and $\mathcal{S}(T_t)$ is observed through the operator’s actions during period t , the model essentially assumes that T_t has been observed by the end of period t . The variable B_t can be observed for example by asking the operator to rate at each period the automation’s capabilities. Nonetheless, in future work we plan to extend our model to a partially observed MDP (POMDP), to model situations where part of the state is not directly observable, i.e., we might only have noisy observations of x_t^h and perhaps x_t^e .

The following reward function is defined at the beginning of period t (see Fig. 2 for the interpretation of the terms)

$$R(x_t^h, a_t^a) = (R^1 + R^m)(H_t^m + H_t^i) + (R^0 + R^m) \left((a_t^a - H_t^m) + (1 - \mathcal{S}(T_t))(1 - a_t^a) - H_t^i \right) + R^1 A_t + R^0 (\mathcal{S}(T_t)(1 - a_t^a) - A_t),$$

with R^1 , R^0 and R^m defined in Section II. We can then search for an automation's policy, i.e., a sequence of functions π_t mapping x_{t-1}^h to a_t^a , maximizing the expected discounted reward over an infinite horizon

$$\max_{\pi=(\pi_0, \pi_1, \dots)} \mathbb{E} \left[\sum_{t=1}^{\infty} \alpha^t R(x_t^h, \pi_t(x_{t-1}^h)) \right],$$

with $\alpha \in [0, 1)$ a discount factor.

Finally, we need to define the automation's true capability function C entering the dynamics (7)-(8). We consider that the human measures the automation's capability to both making correct decisions and suggesting "good" task distributions. Concretely, here C is the rate of correct decisions made by automation and by the human in her assigned tasks

$$C(x_t^h, a_t^a) = \frac{H_t^m + A_t}{a_t^a + \mathcal{S}(T_t)(1 - a_t^a)}. \quad (10)$$

VI. SIMULATION RESULTS

The MDP model defined in Section V was implemented in the Julia programming language using the software package for MDPs [23], and approximately optimal policies were computed with its Local Approximation Value Iteration solver. This algorithm allows a continuous state representation, but the action space had to be discretized (with a stepsize of 0.05). For the simulations, we chose the following parameter values: $\eta = 0.5$, $\mu = 0.5$, $\sigma_B = 0$, $\sigma_T = 0.2$, $R^1 = 100$, $R^0 = -100$, $p = 0.5$ and $P_{FP} = 0.1$. The intervention cost R^m is set to zero, so that the only impact of a human intervention is via an increase of her workload. The values for the detection performance model are shown in Fig. 1 and the discount factor is $\alpha = 0.98$.

The policy computed from the MDP is compared to the static strategy of Section III based only on the detection performance model, without taking into account the effect of trust, which in all cases evolves according to (7)-(8) and the workload according to (9). Fig. 3 illustrates the typical behavior of the two strategies in simulations. In both simulations the trust and belief states start at a low level $T_0 = 0$ and $B_0 = 0$. On Fig. 3, one can see that the MDP strategy gives more tasks to the human when her trust level is high (e.g., at period 13), anticipating the low rate of intervention, in order to set her workload at the correct value. However, it still gives some tasks to the human when her trust level is low (e.g., at period 2 or 24), in order to increase her trust level, e.g., through the influence of H_t^m on the function C in (10).

Fig. 4 shows the distribution over 10,000 simulations of the total discounted reward collected by each policy. It suggests that the MDP policy brings a real benefit to the long term team performance compared to the static policy.

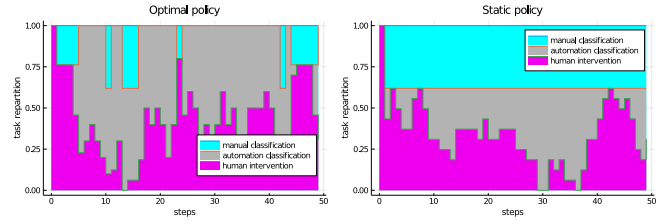


Fig. 3. Allocation of the tasks during each period of time of a 50 steps simulation. The fraction of tasks a_t^a assigned to manual classification by the automation are in cyan, the additional tasks taken by the human are in magenta. The tasks performed by the automation are in gray. For instance, in the left simulation, during period 13, the automation suggests that 40% of the tasks should be completed by the human and the operator accepts this distribution. However, in period 14, the automation suggestion is the same but the operator decides to classify an additional tenth of the tasks. In the left figure, the policy used is computed from the MDP, in the right one, the policy is static with $a_t^a \equiv 0.38$ for all t .

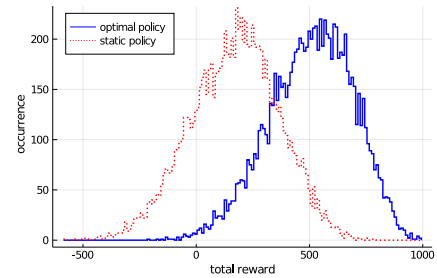


Fig. 4. Reward occurrence for optimal (blue) policy and static (red) policy collected during 10,000 simulations of 50 steps each.

A. Robustness to Errors in the Human Performance Model

In the following, we discuss the performance of the MDP policy in the presence of discrepancies between the models used to compute it and the simulation model. This is intended to capture, to some extent, the effect of errors that will occur when calibrating models in the real world. First, suppose that the workload effect on sensitivity is in fact $d(W) = (d_0 + e)(1 - W)$ instead of (2) used in the MDP, thereby changing the operator's ROC curves. When the error e is positive (resp. negative) the human is more (resp. less) efficient than expected. Fig. 5 shows that the MDP strategy still performs better than the static strategy when both are using the erroneous model. The performance gap between both strategies appears to decrease as e grows however.

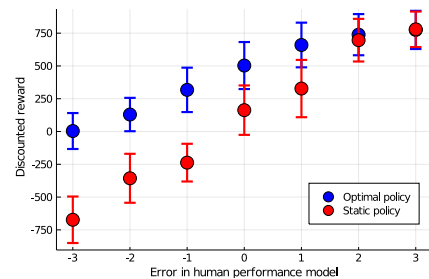


Fig. 5. Mean reward of the dynamic and static policies for a constant error in the human performance model, computed over 1000 simulations. Bars indicate one standard deviation on each side.

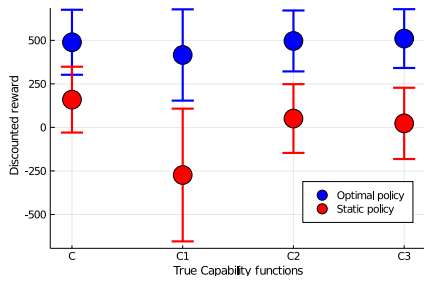


Fig. 6. Mean reward (for 1000 simulations) for the dynamic and static policies for different true capability functions ((10), (11), (12) and (13)) used in the simulation. The MDP assumes (10) to compute the policy.

B. Robustness to Errors in the Trust Model

The definition of the function C defining the automation's true capability from the human's point of view is generally a difficult issue [13]. To test robustness against errors in modeling C , we compute the MDP policy using (10) but we choose another function in the simulation and performance evaluation. As a result, the trust evolves differently than expected when computing the policy. The following candidate functions were tested

- Strict automation responsibility: only the automation's successes in completing the tasks are considered

$$C_1(x_t^h, a_t^a) = \frac{A_t}{\mathcal{S}(T_t)(1 - a_t^a)}. \quad (11)$$

- Biased team responsibility: the overall team success is considered but the payoffs R^0, R^1 and R^m are ignored.

$$C_2(x_t^h, a_t^a) = A_t + H_t^m + H_t^i. \quad (12)$$

- Objective team responsibility: the payoffs are considered (almost the same definition as the MDP)

$$C_3(x_t^h, a_t^a) = \frac{R(x_t^h, a_t^a) - (R^0 + R^m)}{R^1 - (R^0 + R^m)}. \quad (13)$$

Fig. 6 shows that for each of the functions above, the dynamic policy still performs better than the static policy. Future work will consider other robustness tests, e.g., to errors in the parameters η and σ_T .

VII. CONCLUSION

This paper has presented two strategies to allocate decision-making tasks in human-machine teams. The first, static one, takes into account the effect of workload on human performance, while the second strategy also includes a dynamic trust model to predict reliance on automation. The MDP-based approach improves joint long term performance and can handle human or environmental changes more gracefully since it produces a closed-loop strategy. Simulation results show that this performance improvement holds even in the presence of significant mismatches between the anticipated models and the models actually used in simulations. Future work will focus on translating these ideas to real-world human-automation interaction scenarios and identifying parameters based on experimental data.

REFERENCES

- [1] J. Y. C. Chen and M. J. Barnes, "Human-agent teaming for multirobot control: A review of human factors issues," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 1, pp. 13–29, Feb 2014.
- [2] S. Musić and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342–354, 2017.
- [3] R. Parasuraman, K. A. Cosenzo, and E. De Visser, "Adaptive automation for human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and mental workload," *Military Psychology*, vol. 21, no. 2, pp. 270–297, 2009.
- [4] S. Jiang and R. C. Arkin, "Mixed-initiative human-robot interaction: definition, taxonomy, and survey," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 954–961.
- [5] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human factors*, vol. 46, no. 1, pp. 50–80, 2004.
- [6] C. D. Wickens, J. G. Hollands, S. Banbury, and R. Parasuraman, *Engineering psychology and human performance*. Psych. Press, 2015.
- [7] K. Savla and E. Frazzoli, "A dynamical queue approach to intelligent task management for human operators," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 672–686, 2011.
- [8] Y. Sheng, H. Zhu, X. Zhou, and W. Hu, "Effective approaches to adaptive collaboration via dynamic role assignment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 1, pp. 76–92, 2015.
- [9] T. D. Wickens, *Elementary signal detection theory*. Oxford University Press, USA, 2002.
- [10] J. Y. C. Chen, M. J. Barnes, A. R. Selkowitz, and K. Stowers, "Effects of agent transparency on human-autonomy teaming effectiveness," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, October 2016.
- [11] S. Chien, M. Lewis, K. Sycara, A. Kumru, and J. Liu, "Influence of culture, transparency, trust, and degree of automation on automation use," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 3, pp. 205–214, 2020.
- [12] J. Lee and N. Moray, "Trust, control strategies and allocation of function in human-machine systems," *Ergonomics*, vol. 35, no. 10, pp. 1243–1270, 1992.
- [13] J. Gao and J. D. Lee, "Extending the decision field theory to model operators' reliance on automation in supervisory control situations," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 36, no. 5, pp. 943–959, 2006.
- [14] S. Lee, Y.-J. Son, and J. Jin, "An integrated human decision making model for evacuation scenarios under a BDI framework," *ACM Transactions on Modeling and Computer Simulation*, vol. 20, no. 4, 2010.
- [15] A. Xu and G. Dudek, "OPTIMO: Online probabilistic trust inference model for asymmetric human-robot collaborations," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2015.
- [16] C. Nam, P. Walker, H. Li, M. Lewis, and K. Sycara, "Models of trust in human control of swarms with varied levels of autonomy," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 3, pp. 194–204, 2020.
- [17] A. Xu and G. Dudek, "Maintaining efficient collaboration with trust-seeking robots," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3312–3319.
- [18] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Trust-aware decision making for human-robot collaboration: Model learning and planning," *ACM Transactions on Human-Robot Interaction*, vol. 9, no. 2, February 2020.
- [19] H. Saeidi and Y. Wang, "Trust and self-confidence based autonomy allocation for robotic systems," in *IEEE Conference on Decision and Control (CDC)*, Melbourne, Australia, December 2015.
- [20] H. Saeidi, J. R. Wagner, and Y. Wang, "A mixed-initiative haptic tele-operation strategy for mobile robotic systems based on bidirectional computational trust analysis," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1500–1507, 2017.
- [21] L. M. Hiatt, C. Narber, E. Bekele, S. S. Khemlani, and J. G. Trafton, "Human modeling for human-robot collaboration," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 580–596, 2017.
- [22] P. J. Bickel and K. A. Doksum, *Mathematical statistics: basic ideas and selected topics, volume 1*, 2nd ed. CRC Press, 2015.
- [23] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017.