

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.Doi Number

Real-Time Scheduling with Independent Evaluators: Explainable Multi-agent approach

Artem O. Isakov¹, (Member IEEE), Danil E. Peregorodiev, Ivan A. Tomilov, and Natalia F. Gusarova

¹Department of Infocommunication Technologies, ITMO University, 197101, Saint-Petersburg, Russia

²World-Class Research Center for Personalized Medicine, Almazov National Medical Research Centre, 197341, Saint Petersburg, Russia

Corresponding author: Artem O. Isakov (e-mail: aoisakov@itmo.ru).

This work was supported by the Ministry of Science and Higher Education of the Russian Federation, Goszadanie (State Assignment) No. 2019-1339.

ABSTRACT This paper introduces a multi-agent reinforcement learning approach to address the challenges of real-time scheduling in dynamic environments, with a specific focus on healthcare operations. The proposed system integrates the Human-in-the-Loop (HITL) paradigm, allowing for continuous feedback from human evaluators, and employs heavy-tailed reward distributions to better handle outlier behaviors and emergent events. A novel mapping between reinforcement learning concepts and the Belief-Desire-Intention (BDI) framework is developed to enhance the explainability of the agents' decision-making processes. The system is designed to adapt to changes in patient conditions and preferences while minimizing disruption to existing schedules. Experimental results show significant improvements in patient waiting times and adherence to management constraints compared to traditional methods. The approach offers a robust, explainable, and adaptable solution for complex scheduling tasks in environments that require human-centered decision-making.

INDEX TERMS decision making, explainability, multi-agent systems, real time, reinforcement learning, scheduling

I. INTRODUCTION

The task of scheduling manufacturing production has a long history, but with the growth of the range and increasing diversification of production processes, its importance is only increasing. Scheduling production processes is shown to be NP-hard combinatorial problem [Zhang J.] and cannot be fulfilled in polynomial time; therefore, a plethora of approximate algorithms have been developed that reflect the content and specificity of individual production processes. In particular, the Multi-agent Reinforcement Learning (MARL) approach [Albrecht], in which a collective of agents learn to optimally interact in a shared environment during the game, allows one to take into account some of the uncertainties in the description of the production process. The evolution of the MARL approach to Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [Lowe] makes it possible to construct a schedule for a production process with a dynamically changing number of agents.

Today, production processes involving people are increasingly on the agenda. In this case, when drawing up

a schedule, it is necessary to take into account the opinion of the person participating in the process, including the implicit one, i.e., expressed only post factum (through assessments of proposals generated by the algorithm). That is, the scheduling problem moves from the class of computationally complex problems to the class of systemically complex problems: the NP-completeness of the scheduling problem is superimposed by the non-formalizability of a number of parameters associated with human behavior and with the online changes in the parameters of the process as a whole inspired by him. The main approach for such cases is Human-in-the-Loop (HITL) [Retzlaff]. In this paradigm, humans provide feedback in the form of preferences or scalar signals, which is then used to update the game parameters, primarily the reward function. Since it is often challenging or even impossible to define a reward function monotonically related to human preference, the Direct Preference Optimization (DPO) [Rafailov] approach is proposed as an alternative, which directly shapes the model's decisions based on positive or negative human feedback.

Within the framework of the HITL paradigm, a number of other reward-free approaches have been proposed [Wu X., Muslimani, Wu J.]. However, despite the differences in solutions, they all have a common limitation: to evaluate the solution proposed by the model, either the opinion of one person or the agreed opinion of several participants is used (in the latter case, they act as a distributed decision maker). In modern client-oriented processes, this limitation is often too strong: it is necessary to be able to take into account the individual desires and states of each client, including implicit ones and those that change during the process.

A typical example of such a scenario is the organization of the flow of high-tech medical interventions in a specialized clinic (surgery, dentistry, high-tech medical examinations, etc.). Patients who come to the clinic have a certain degree of severity of the disease, which determines the possible waiting time for the intervention, as well as personal preferences regarding the time of its implementation. The clinic, for its part, can provide a certain number of interventions each day. Based on this data, a schedule is drawn up, which is communicated to patients. However, during the wait, both the patient's condition and his preferences regarding the time of the intervention may change, which forces an ad-hoc reorganization of the entire schedule. As a result, in general, the waiting time of other clients increases and expensive workstations are underloaded.

To address these aforementioned challenges, this study proposes an algorithm that enables rapid rescheduling of medical interventions in response to changes in the state and wishes of individual patients, while minimizing disruption to schedules for already scheduled patients. Specific features of the proposed algorithm are as follows:

- Heavy-tailed distribution is used to enhance reinforcement learning algorithm ability to handle outlier user behavior;
 - Each human actor has two roles: through the intellectual agent representing him - as a participant in the game process; directly - as an independent evaluator.
- The main contributions of this paper are as follows:
- We devised a novel multi-agent environment to address the challenges of extending the planning horizon in dynamic scheduling;
 - We design a sophisticated reward function incorporating heavy-tailed distributions with adaptive alpha levels to attenuate the effect of human driven emergent events;
 - We propose a methodology to refine the decision-making process during the sequential rescheduling phase in presence of active human agents with the modified direct policy optimization technique;
 - We established a strong association between reinforcement learning nomenclature and Belief-Desire-Intention cognitive framework and suggested

LLM-based tools to elucidate intricacies of cooperative-competitive game interactions between agents for a supervising human expert;

- We transferred our findings to the problem of operation theatre rooms allocation.

The last of the paper is organized as follows. In the Related Works section, we present a review of recent literature on the main approaches used in our work. Particular attention is paid to methods for specifying parameters of human activity and behavior in the context of solving production process scheduling problems.

II. RELATED WORKS

The Multi-agent Reinforcement Learning approach and its various modifications, including Deep MARL (DMARL), Multi-Agent Actor-Critic (MAAC) and Multi-Agent Deep Deterministic Policy Gradient (MADDPG), have been widely presented in the literature in recent years; their advantages and applicability to production process scheduling problems has been summarized in reviews [Wong, Panzer, Al-Hamadani, Zhang K.].

In particular, the papers [Pu, Wan, Mangalampalli, Monacia] consider the problem of workflow scheduling under conditions of current changes in the states of the environment and agents in it. In [Mangalampalli], task priorities are calculated for individual users and workstations and then fed to a scheduler that dynamically schedules tasks using the Deep Q-Network model and a swarm algorithm. In [Pu], each workpiece is treated as an intelligent agent, and the representation of state, action, observation, and reward is introduced based on the Markov decision-making formula; in order to compute policies, a heterogeneous graph neural network based on graph node embedding is used. In [Wan, Monacia], actor-critic reinforcement learning method is used to solve the flexible job shop scheduling problem: the actor network is responsible for choosing the most suitable scheduling rule in different states, while the critic network is responsible for outputting the value function of the actions and providing feedback to the actor network to better adjust the scheduling strategy. The works [Wan, Monacia], along with others, has shown that the actor-critic approach is computationally efficient and flexible to the number of jobs, agents, and workstations compared to genetic [Mangalampalli] and graph [Pu] approaches.

It should be noted that a mandatory condition for the application of MAAC [Lowe] and MADDPG [Zhang K.] approaches is the possibility of agents grouping (typification) [Amir, Lage], which, at first glance, contradicts the principles of individualized medicine. However, in the review [Williams] the credibility and relevance of such typification for patients of medical institutions was confirmed. At the same time, the possibility of flexible changes in the preferences of individual agents during the process is not supported by all the indicated works.

The Human-in-the-Loop (HITL) paradigm aims to introduce weakly formalized parameters of human activity and behavior into models for production process scheduling. The large flow of literature in recent years, summarized in reviews [Mosqueira-Rey, Gómez-Carmona, Wu X., Retzlaff], testifies to the demand for this paradigm in real applications.

To reach the main goal of the HITL paradigm - increasing the level of formalization of the description of human activity - various approaches have been proposed.

One of the first is Inverse Reinforcement Learning algorithm [Abbeel] based on capturing domain-expert knowledge from demonstration, which derives an unknown reward function from expert's observed behavior. But as the number and complexity of tasks to be scheduled grows up, the size of the state space increases dramatically, and scheduling can quickly become computationally intractable. [Gombolay] uses action-driven learning to extract the scheduling strategies of domain experts, and training examples are created based on pairwise comparisons between scheduled and unscheduled tasks. A pairwise approach describes the behavior of experts in a more natural way and is less resource-intensive. A similar approach is presented in [Christiano]. In order to support reward learning from inconsistent and diverse human preferences, [Xue] propose to stabilize it by regularizing and correcting its predictions in a latent space. Other options for directly learning reward functions from human feedback are presented in [Hejna, Liang, Ge] etc.

In general, this group of approaches allows one to reproduce with a certain degree of accuracy the decisions made by an experienced dispatcher, but it cannot improve or explain them.

Another approach to the formalization of the description of human activity can be experts' typification, which aims to divide them into statistically homogeneous cohorts. In [Walsh, Arakawa], the parameter of typification is the decision-making scheme. Namely, authors [Arakawa] introduce five key characters to consider about human feedback when launching human-in-the-loop RL systems: Binary, Delay, Stochasticity, Unsustainability, and Natural Reaction. [Walsh] experimentally show that humans in decision-making follow three distinct decision strategies which demonstrated similar degrees of success as measured by task performance: one cluster utilized a more analytic approach to decision-making, while the other two clusters utilized more heuristic decision-making strategies. Such clustering allows the creation of software agents that more accurately represent human behavior. [Rzeszucinski] divides the humans involved in the scheduling by their place in the process timeline: Human in the Loop provide sole or collaborative decision-making, Human on the loop - supervisory oversight, Human above the loop - strategic governance, Human behind the loop - output analysis and improvement. However, such kinds of typification of experts exclude from consideration their implicit and spontaneous decisions, which sharply narrows the possibilities for generalizing the approach.

It seems more promising to take into account the distribution of process participants according to individual parameters that are most important when making a decision. In particular, for the scenario described in the Introduction to this article, such a parameter is human availability times. In [Varga] its distribution is considered Poisson: users initially propose a single starting time for each of their jobs, then scheduling approach based on integer linear programming proposes each user a small number of alternative time intervals, and then the user may accept or reject these. As the authors argue, an initial schedule can usually be quickly improved over a few interaction rounds. Meanwhile, the investigations [Barabási, Vázquez] show that when individuals execute tasks based on some perceived priority (which is just the case for decision-making), the timing of the tasks will follow heavy-tailed distributions. Accordingly, for processes involving a human as a decision-maker with spontaneous and implicitly inspired decisions, it seems organic to model reward distributions using asymmetric functions, including heavy-tailed ones. The literature [Zhu, Cayci, Yu Lu] presents RL algorithms using a non-Gaussian return function, but it is always considered as the same for all actors, and its parameterization for different actors is not provided.

Since the rewards that are artificially crafted may not accurately mirror human intentions, the reward-free RL models have become very popular. Direct Preference Optimization approach [Rafailov] allows organizing RL without explicitly setting the reward function, by using a change of variables to define the preference loss as a function of the policy directly. Preference-Based Reinforcement Learning [An] directly learns from preference without requiring any reward modeling. To achieve this, it adopts a contrastive learning framework to design a policy scoring metric that assigns a high score to policies that align with the given preferences. However, Direct Preference Optimization uses LLM as a domain model, and Preference-Based Reinforcement Learning is designed for offline RL tasks. Their adaptation for online scheduling tasks has not been presented in the literature.

AI systems for safety-critical applications must explain their decisions, actions, or predictions to ensure transparency, explainability, and accountability – this requirement, enshrined in governmental documents of a number of countries, has given rise to a vast stream of research. In reviews [Wells, Sado] for explainable AI (XAI) systems, a spectrum of classification features has been formed, which allows us to identify the areas of research that are closest to the problems of this article.

XAI systems are divided [Sado] into data-driven and goal-driven. Data-driven XAI algorithms aim to determine the input attributes that account for output predictions [Guidotti]. For example, the SHAP [Lundberg] and GradCAM [Selvaraju] algorithms highlight, respectively, those features of the input dataset or those pixels of the input image that had

the greatest impact on the classification result performed by the neural network. Goal-driven XAI [Anjomshoe] aims to create explainable robots or agents that can justify their behaviors to a lay user. Although there is a growing application of goal-driven XAI in the current AI-dependent world [Langley], the choice of algorithm type depends primarily on the target person to whom the explanation is provided [Brännström, Yan].

The paper [Yan] proposes to consider the XAI problem for a specific agent system from a multi-level perspective: the implementation level - for developers, the knowledge level - for designers, the domain level - for users. The authors emphasize that explanations for different addressees, even built on a single source of information, should not coincide.

The most commonly used sources of information for constructing explanations in agent systems are records of important system events, or characteristic trajectories, as well as the complete system log [Alelaimat, Dennis, Jan]. In the latter case, two modes of audit logging can be distinguished: behavior logs, and belief logs [Alelaimat].

Various agent models have been proposed that allow one to describe their behavioral manifestations with varying degrees of completeness. For example, reactive RL agents [Cruz, Sequeira] rely on a simple behavioral policy scheme, i.e., a state-to-action mapping via trial-and-error interaction with the environment. Such an agent model provides eXplainable Reinforcement Learning in terms of ongoing evaluation of individual agent parameters, first of all - reward functions [Zhang G]. But the most complete understanding of agent behavior is provided by cognitive models, primarily Belief-Desire-Intention (BDI) model [Bratman], which uses folk psychology ideas about human mental attitudes to implement rational agents. Here Belief is the agent's ideas about the current state of the environment, Desire is its ideas about the target state of the environment, Intention is the agent's actions for the current and next game steps. In the context of multi-agent systems, explaining emergent behaviors and complex agent interactions presents additional challenges [Ciatto et al.], and cognitive models best meet this challenge [Georgeff et al.]. The review [de Silva] confirms the justified popularity of the BDI model for constructing agent-based XAI.

Visual and iconic forms are widely used to represent explanations, such as highlighting significant pixels in the input image [Selvaraju], graphical representation of the relative importance of features [Lundberg], significant agent trajectories during Atari games, and significant samples to construct explanations (Zhang G), etc. However, in recent years, increasing attention has been paid to natural language explanations for reinforcement learning agents [Shu et al., Ehsan et al.]. The application of large language models (LLMs) in the field of XAI has opened new possibilities for generating more nuanced and context-aware explanations [Brown et al.]. Some researchers [Anderson] have explored the use of LLMs for dynamic role assignment in explanation

generation, allowing for adaptive and flexible interpretation of agent behaviors.

As noted in the review [Sado], the development of metrics for explainable behavior of multi-agent systems provided by XAI systems remains an open issue. The need for their development is stated in [Brännström], and some requirements for them are formulated in [Winikoff]. The comprehensive approach to developing a system of metrics for explainable behavior of multi-agent systems, which takes into account different levels and different addressees of explanation, presented in [Yan], seems conceptually important, but it has not been brought to practical implementation and has not been continued in other works. Examples of the practical implementation of such metric complexes in relation to textual representations of explanations were not found in the literature available to the authors.

The main challenges remain reconciling the spontaneous and implicitly justified decisions of individual users regarding the proposed schedule options, as well as implementation and metric support of the complex explainable multi-agent system.

IV. METHODS AND MATERIALS

A. ENVIRONMENT DESIGN

In the Preliminaries section we thoroughly discussed base ideas behind reinforcement learning. In brief every reinforcement learning environment defines state S and action spaces A . While operating in the environment, the intelligent agent observes the current state of environment or its subset O and draws an action. In response on this action environment changes state and provide a feedback signal called reward R .

To preserve experiments consistency and reproducibility we chose a Python 11.9 programming language and PettingZoo 1.24.3 framework with a standardized application program interfaces (APIs) namely Agent Environment Cycle (AEC) and Parallel to build a custom environment. On the one hand Parallel environment API is based around the paradigm of Partially Observable Stochastic Games (POSGs) on the other hand it provides agents with simultaneous actions, which meets the intricacies of the real-world resource allocation tasks.

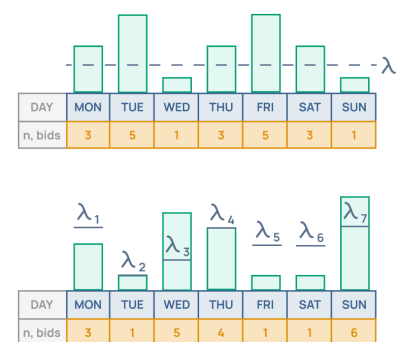


Fig. 1. State space

We defined environment state space as a timeline of a certain length called planning horizon with limited throughput capacity λ on each time slot. Fig. 1 (a) shows a special case with a uniform target distribution across the planning horizon with fixed λ . Fig. 1 (b) shows a general case with different target λ levels across the planning horizon.

When it comes to real world application it's convenient to start with a uniform target distribution while training and fine tune with different target λ levels later.

Linearity in design allows us to shift the planning horizon over time, providing a flexible framework to work with continuous planning models (see Fig. 2).

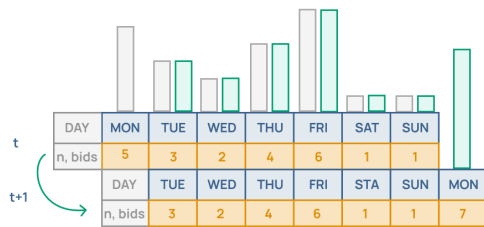


Fig. 2. Time shift in planning horizon

Left alone this state representation is prone to overfitting which in terms of reinforcement learning means instead of learning meaningful insights algorithm is perfectly adapted to a certain variant of gaming scenario. To mitigate the effects of overfitting, we need to train on multiple instances of the environment in parallel and limit the size of the agent's observation space and freedom of action.

Action space is designed in a way each agent can't learn to put its bid on a certain day rather to learn relative position compared to others. On each stage of the game agents can move their bids one day further or closer to the starting point of a planning horizon.

Fig. 3 (a) show action space for a one-week long planning horizon. Starting from Tuesday agent can move its bid on Monday, Wednesday or to leave it as is.

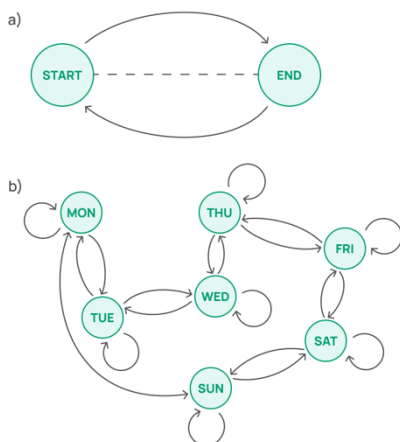


Fig. 3. Action space

Fig. 3 (b) shows the manner in which prohibited actions are addressed. Adhering to the same rules, staying on Monday will lead to a collision, because there is no room to move

towards the left border of the planning horizon. The same applies to the last day of the planning horizon.

There are many ways to resolve collisions in reinforcement learning, for instance we can mask agent action space when staying on boundaries, define a penalty to choose action that causes a collision or even go further and omit hard boundaries of a planning horizon.

Instead, we implemented a cyclic state space. That said if we start from Monday and choose to move left in a planning horizon, we will find ourselves on Sunday and vice versa. Given the limited number of game episodes it allows us to eliminate unnecessary steps for agents to come up with an optimal solution.

Observation space encompass agent knowledge about itself and the environment around it and depends on a window parameter $w \in \{2n + 2 \mid n \in \mathbb{N}, n \geq 2\}$ which limits agent vision. On Fig. 4 (a) w is set to 3, that said agent observes the environment state for three days in a row and its absolute position in the planning horizon.

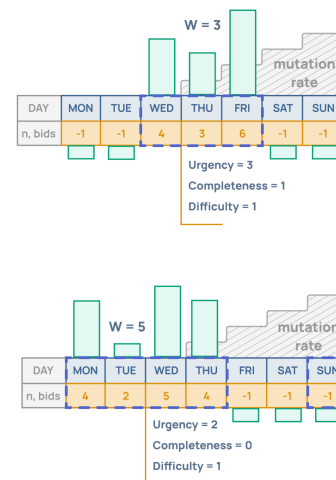


Fig. 4. Observation space

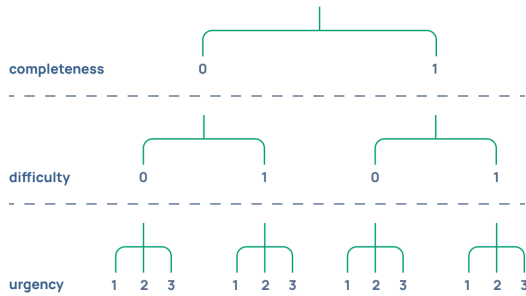
Some agents need more information to come up with an optimal policy, so we deliberately widen the agent's window size as depicted on Fig. 4 (b). The longer it takes to learn optimal policy the wider the window size. That brings agents more information but also higher penalties as we will see later.

In order to maintain the size of the observation space throughout the training procedure, we use the full size of the planning horizon and set an unknown number of bids on a day to a small negative value. This provides a reinforcement learning algorithm with a clear distinction between unknown, zero and non-zero number of bids on that day.

As previously stated, in addition to the knowledge about surrounding environment agents perceive themselves. In the context of our particular field that means agents assess the health status of the patients they serve. More broadly, this self-assessment could apply to any subject or entity.

The patient's health status (see Fig. 5) is a combination of data completeness, difficulty and urgency of medical

intervention. Data completeness evaluates the sufficiency of the available data necessary to provide the best treatment, $c \in \{0, 1\}$. Difficulty is an overall assessment of the complexity of a medical intervention, $d \in \{0, 1\}$. Urgency limits agent ability to wait, higher values indicating higher need to receive medical care, $u \in \{1, 2, 3\}$.



There is an association between the day on which the agent decided to bid and the patient's health status. Fig. 4 shows that the mutation rate slowly increases after the second half of the planning horizon at a rate of 5%. This mutation coefficient

Fig. 5. Patient's health state determines the probability of a change in the patient's health status and thereby affects agent observations.

The reward function is directly influenced by the selection of a termination rule. In a highly complex gaming scenario termination rule is supplemented by a truncation rule. The termination rule delineates the optimal outcome anticipated, while the truncation rule ensures that the game does not exceed a predefined number of episodes.

While effective this approach takes more time to train and tune later. Contrary we designed a termination rule with deliberate simplicity, considering the necessity for a system built upon a reinforcement learning algorithm to operate efficiently in real-time scenarios.

That said, we suggested setting a certain small number of steps up to a planning horizon length. Once this number is reached, the game will be over and the reward will be calculated as:

$$r = -\frac{b}{k} \sum_{d \in W} f(d) |s_o(d) - s_t(d)|$$

where b denotes a base unit reward adjusted with a scaling factor k , $f(d)$ is the value of the Lévy alpha-stable probability density function on d - a day within the observation window w , $s_o(d)$ and $s_t(d)$ are observed and target states of the environment on that day, respectively.

The scaling factor attenuates penalties for agents representing moderate to high-risk patients and calculated as:

$$\text{argmax}(1, ((d + (1 - c)) * u))$$

where c - data completeness, d - difficulty and u - urgency of medical intervention.

Initial α for the Lévy alpha-stable distribution is set to 2 which makes it close to a Gaussian distribution with $\mu = d$ and $\delta = 1$ as depicted on Fig. 6 (a). However, during the fine-tuning process, the window parameter w increases and at the same time α decreases, see Fig. 6 (b). Subtle decreasing α will slightly move distribution towards Cauchy distribution.

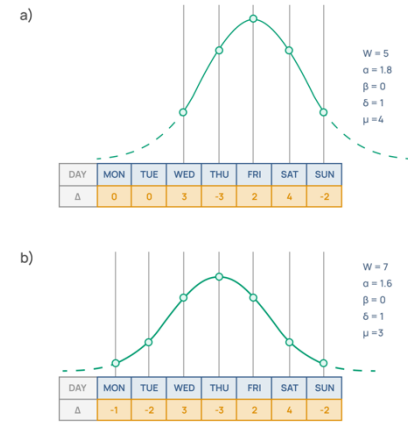


Fig. 6. Reward function design

Thus, by assigning a larger width to the observation window, we give agents more freedom and informed choice, but at the same time we penalise them more for deviations from the target distribution at the ends of the observation window by lowering α .

It should be noted that for certain reinforcement learning algorithms, such sparse reward design is considered to be disadvantageous. To mitigate the adverse effects of sparse rewards, a small negative value can be assigned to an agent at each step it takes. If we exclude penalties for a stay action, this strategy will also incentivise agents to minimise their actions to achieve the same goals, thus promoting more efficient behaviour.

B. RE-SCHEDULING WITH HUMAN FEEDBACK

In order to reduce the computational costs associated with the re-scheduling it is proposed that a limited set of intelligent agents be trained to represent categories of end-users, in this case, patients. In order to facilitate efficient operation, the model has been pre-trained with twelve agents by twelve health states discusses earlier. During the inference stage, each patient is matched with one of these intelligent agents, which will represent their interests.

Fig. 7. shows a n -step re-scheduling process. At first stage a reinforcement learning algorithm propose a draft of the future schedule. Multiple human evaluators assess this proposal and lead expert receives their affirmative or negative answers. Based on provided human feedback lead expert refines schedule and process repeats.

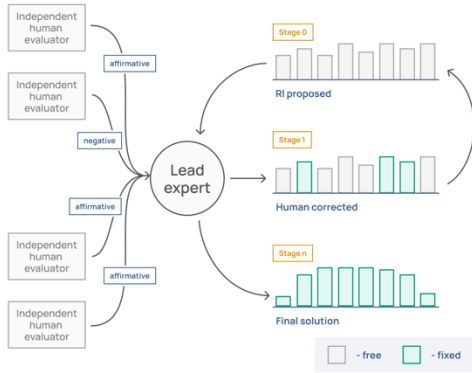


Fig. 7. Successive re-scheduling

Patients who have consented to the selected time slot are removed from the game. This results in the deactivation of the agent representing them, and the game continues as if they had consistently chosen the "stay" action. Once all patients have provided their consent, the game concludes, allowing for the evaluation of the quality of the solution provided.

C. EVALUATING THE QUALITY OF THE SCHEDULE

In order to evaluate the quality of the generated schedule, we employed the Kullback-Leibler (KL) divergence measure in conjunction with a number of less rigorous, empirically derived metrics. These supplementary metrics should assess the schedule from the standpoint of the end user, particularly the operator of the multi-aggregate system, such as a manager.

In order to apply the Kullback-Leibler divergence, we need to convert the obtained request frequencies into pseudo-probabilities. This is done by dividing the frequency of requests on the selected day by the sum of the frequencies over the entire planning horizon.

$$D_{KL}(P||Q) = \sum_{d \in H} P(d) \log_2 \frac{P(d)}{Q(d)}$$

where $P(d)$ represents the relative frequency of requests on day d within the simulated planning horizon of length H , and similarly, $Q(d)$ denotes the relative frequency of requests that correspond to the constraints set by the operator of the multi-agent system.

In order to assess the anticipated violations of the preferences established by the operator of the multi-agent system, the following expression is employed:

$$V_P = \frac{1}{H} \sum_{d \in H} \frac{s_o(d)}{s_t(d)}$$

where $s_o(d)$ and $s_t(d)$ represent the mean observed state assessed over multiple episodes and the target states of the environment, respectively, on day d within the planning horizon of length H .

Furthermore, we are interested in the mean position occupied by agents representing the interests of patients or in other words the average waiting time (AWT) of patients with a condition classified as relatively severe:

$$AWT(k > 3) = \frac{1}{n} \sum P(k > 3)$$

where $P(k > 3)$ represents the positions of agents with a scaling factor $k > 3$, obtained over the course of n episodes.

D. MAPPING BETWEEN BELIEF-DESIRE-INTENTION STRUCTURE AND REINFORCEMENT LEARNING TERMINOLOGY

In Figure 8, the conceptual transitions from classical reinforcement learning paradigms to the constituent elements of the Belief-Desire-Intention (BDI) framework are illustrated.

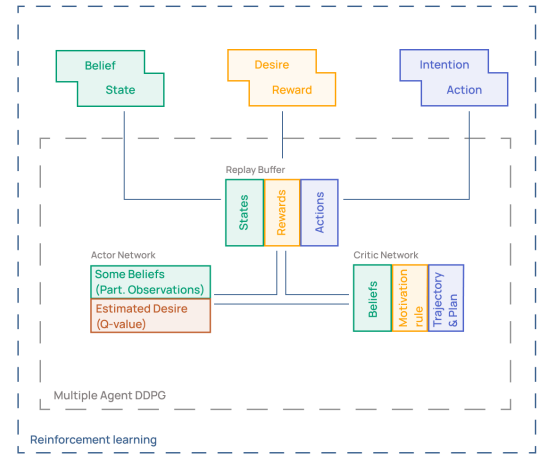


Fig. 8. Belief-Desire-Intention terminology mapping

Beliefs are constructed from the intelligent agent's internal representations of the current and target states of its environment. Within the context of reinforcement learning, beliefs are analogous to the state space. The agent's desires encapsulate the priorities established by the researcher or acquired through iterative interactions with the environment. The reward function serves as the closest analogous to the short-term desires. Intentions delineate the actions that the agent plans to execute in order to transition the observed state of the environment to the desired target state.

At the level of the conceptual architecture of the Actor-Critic model, all three components – beliefs, desires, and intentions – are collected within the Replay Buffer and are used in the training process of the Critic network. The derived values from these components subsequently contribute to the training of the Actor network.

The mapping of reinforcement learning concepts into the BDI paradigm facilitates the integration of BDI goal-setting methodologies and enables the development of explainable RL-agents, thereby foster the interpretability and transparency of autonomous decision-making processes.

E. MULTI-LEVEL EXPLAINABILITY

According to the concept on which our implementation of a multi-agent system is based, the elements of the system serve a single global goal but follow the interests of different human representatives, who have their own local and not necessarily strictly aligned interests. Therefore, following the work of [Young], we need different explainability for each end-user.

The resulting multi-agent system has three levels of explainability (see Fig. 9).

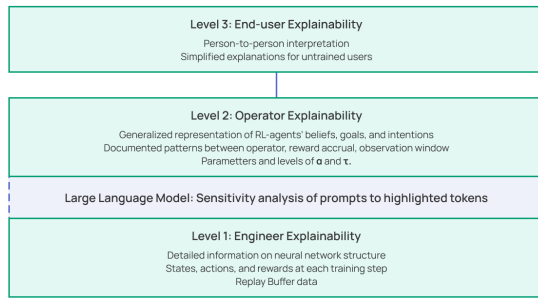


Fig. 9. Multi-level explainability in the Human-Agent Systems

At the first level, we have voluminous information that requires specialized engineering training about the structure of the neural networks that make up the RL-agents, as well as the states, actions, and rewards at each step of the training, stored in the Replay Buffer.

At the second level, we have a more generalized representation of the beliefs, goals, and intentions of the RL-agents, as well as documented patterns between the customized operator of the multi-agent system, the reward accrual process, observation window parameters, and the levels of α and τ .

The first and second levels are connected by the work of a large language model, so the sensitivity analysis of prompts to highlighted tokens can be useful and interpretable by both the engineer and the operator of the multi-agent system.

The third level of explainability implies an explanation that the end-user receives. Since the end-user generally has not undergone the required training, the interpretation at this level is from person to person.

F. NATURAL LANGUAGE BRIDGING

That said, language models can be used to generalize concepts of beliefs, desires, and intentions to provide a bidirectional interface between the levels of explainability of the engineer and the operator of a multi-agent system.

For interpreting system logs, we developed a series of prompts guided by the Chain of Thoughts (CoT) methodology [Wei] (see Figure 10).

In brief, CoT refers to a technique where the sequence of reasoning steps and intermediate thoughts is explicitly laid out in successive prompts. This method leverages the capabilities of language models to follow structured thinking patterns, breaking tasks down into manageable subtasks and thereby improving the quality of responses.

Appendix A presents a full listing of the devised prompts from the Claude Sonnet 3.5 language model we chose for experiments because of its excellent reasoning over the text according to [https://github.com/microsoft/promptbase].

G. PROMPT SENSITIVITY ANALYSIS

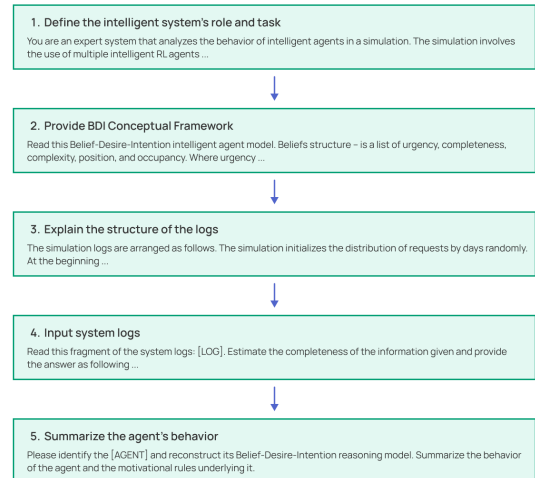


Fig. 10. Chain of Thoughts for interpreting system logs

In order to ensure the reliability of the outputs produced by a language model, it is crucial to identify the tokens to which the model assigns greater significance during the process of response generation.

It is hypothesised that the model assigns greater significance to tokens whose alterations result in a significant variation in the model's output. It is therefore essential to evaluate the extent to which the model's response is sensitive to the input prompt.

To achieve this, we proposed using the same Claude Sonnet 3.5 to mask key concepts and related words in the prompt, as illustrated in Figure 11. Appendix B presents a full listing of the devised prompts

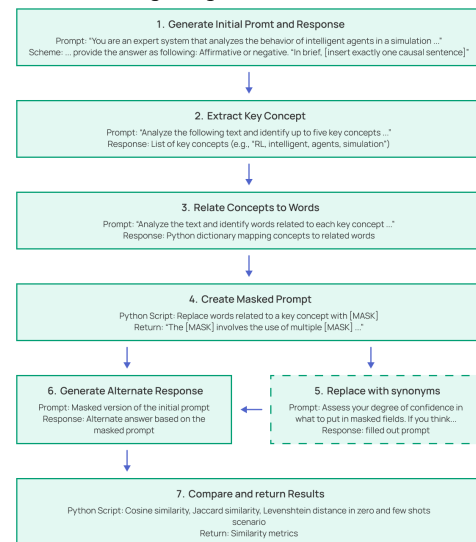


Fig. 11. Chain of thoughts to analyze prompts sensitivity

By inputting the masked prompt into the model, we suggest evaluating the difference in responses using cosine similarity provided by scikit-learn 1.5.1 scientific computing framework. The measure is useful for comparing texts in terms of their overall meaning or in other words their semantic similarity. The calculation is as follows:

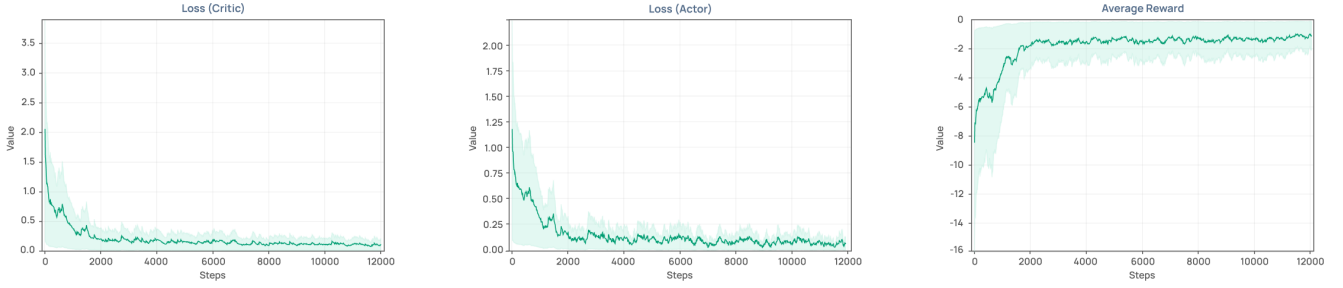


Fig. 12. Error metrics of actor networks (Fig. 12 – a), critic networks (Fig. 12 – b) and average episodic error (Fig. 12 – c) during model training

$$\cos(\theta) = \frac{a \cdot b}{\|a\| \|b\|}$$

where a and b – the first and second responses, respectively, θ – angle between representations of responses in vector space.

The Jaccard distance measure is a more straightforward approach that circumvents the necessity for a translation of the original language model responses into numerical form, which would otherwise require a TF-IDF tokenization operation. Moreover, Jaccard's measure is more granular in its approach, focusing on the presence of individual words as the fundamental unit of comparison. The calculation is as follows:

$$J(a, b) = \frac{|a \cap b|}{|a \cup b|}$$

where a and b – the first and second responses, respectively.

In order to further examine the concept of granularity in the context of comparing two texts, we will now turn our attention to the Levenshtein distance measure. This particular measure has been designed with the specific purpose of detecting the smallest character-by-character differences, and it is therefore particularly well suited for use in situations where the language in question is relatively short and structured. The calculation is as follows:

$$L(a_i, b_j) = \begin{cases} \max(i, j) & \text{If } \min(i, j) = 0 \\ \begin{cases} L(a_{i-1}, b_j) + 1 \\ L(a_i, b_{j-1}) + 1 \end{cases} & \text{otherwise} \\ L(a_{i-1}, b_{j-1}) + 1_{(a_i \neq b_j)} \end{cases}$$

where a and b – the first and second responses, respectively, i and j – are the current indices being compared.

V. RESULTS

Fig. 12 illustrates a gradual decline in actor and critic network errors following approximately 2,000 steps. The mean episodic return increases rapidly, approaching a value of zero.

The distinctive nature of the game environment proposed for the scheduling problem precludes the use of conventional benchmarks for direct comparison. In lieu of these, we have conducted a theoretical experiment.

The performance of the resulting algorithm is compared with the simple empirical sorting logic. We propose the

assignment of twelve patients, with the internal parameters of each randomly selected in accordance with the constraints of the queue data structure. The first step will be to allocate patients using the first-in-first-out (FIFO) method. Once the first day is filled, the scheduling horizon will be shifted forward on each subsequent day, in accordance with the constraints set by the manager on the λ levels for each day.

Table 1 shows the best achieved results of our experiments in one and few shots scenarios.

Table 1. Experiment results

Algorithm	Metrics	Zero-shot	Few-shots average (n=10000)
Basic heuristic	D_{KL}	0.0	0.0
	$V_p, \%$	0.0	0.0
	AWT ($k > 3$), days	2.0	2.26
MAPPO	D_{KL}	0.0025	0.0062
	$V_p, \%$	2.86	9.96
	AWT ($k > 3$), days	0.9	1.75

As anticipated, the results demonstrate a strict adherence to the constraints imposed by the manager on the maximum lambda throughput of each day. However, the mean waiting time for patients with scaling factor $k > 3$, who conditionally correspond to patients with severe health conditions, is approximately equal to the weighted mean position of the agent in the planning horizon. Conversely, MAPPO's performance was noticeably better: for a minor concession on the part of the manager (five-shot average deviation from manager preferences $< 10\%$), the waiting time for a sick patient to undergo a scheduled surgery was reduced by 22.5%.

Let's increase the number of patients to one hundred. In this instance, each patient will be assigned one of the twelve health categories and will be treated using the same pre-trained models (see Fig. 13). To accommodate the inherent unpredictability of real-world scenarios, we will permit patients, with a probability of 30 to 70%, to decline the manager's proposed schedule. In such instances, the intelligent agents aligned with their interests will be randomly

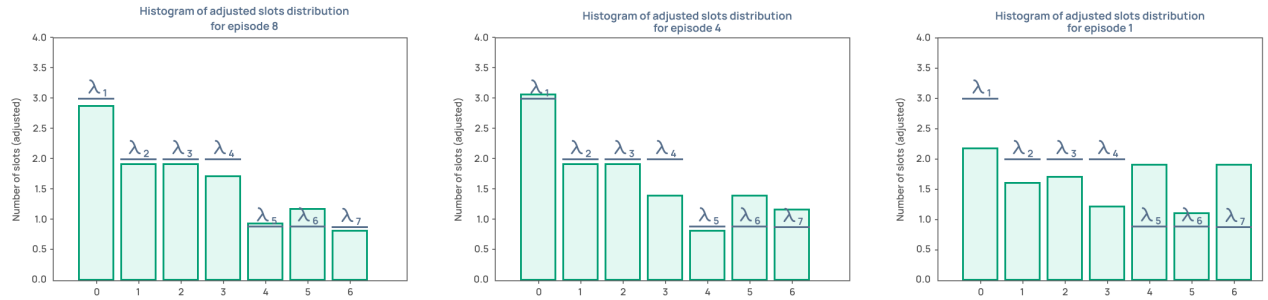


Fig. 13. Patient allocation during the iterative rescheduling portion of the schedule

reinitialized within the environment, and the game will resume as if other agents had been frozen.

To facilitate the tracking of their numbers in relation to the throughput levels indicated, we scaled them to the desired level by dividing the number of patients on the selected day by the total number of patients and multiplying by twelve. Consequently, 25 patients who scheduled their visit on Monday would be $(24/100) * 12 \approx 2.9$.

As illustrated in Fig. 13-a, the agents were initialized at random. Subsequently, in the fourth iteration of the replanning process (see Figure 13-b), the schedule began to demonstrate a favorable improvement. In the final iteration of the planning process (see Figure 13-c), the schedule exhibited minimal movement, approaching the theoretically anticipated outcome.

Since the replanning process uses the same models used to estimate the behavior of the twelve base agents, the D_{KL} , V_P , and AWT ($k > 3$) measures remain at the same level.

The explainability of the system under consideration is achieved through log analysis by large language models, as detailed in the Methods section (Sections D-F). The following example illustrates the behavior summarization of agent number 10 during one of our experiments:

«Agent_10 starts with a moderately urgent and complex task scheduled for Monday. Initially, with incomplete information, the agent assesses its situation as moderately unfavorable and decides to move forward in the week (from Monday to Tuesday). Upon moving to Tuesday, the agent reevaluates its beliefs, showing an updated but still incomplete understanding of the situation. Despite a slight improvement in its assessment (desire becoming less negative), the agent chooses to move backward, returning to Monday».

A comprehensive listing of all responses generated by the large language model can be found in Appendix C.

In order to enhance the reliability of responses from large language models and streamline the prompt engineering task, we devised a method to assess the susceptibility of prompts to arbitrary alterations in the input tokens. The findings of this investigation are documented in Appendix D.

VI. DISCUSSION

A. DIRECT PREFERENCE OPTIMIZATION

It is important to note that once trained, a multi-agent system must be able to adapt over time to new preferences set

by the manager. To do this, we suggest following the Direct Preference Optimization cycle (see Fig. 14).

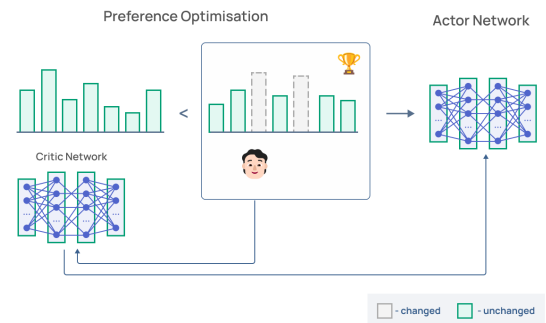


Fig. 14. Direct preference optimization

Firstly, it is necessary to determine the additional penalty value, which determines the punishment for agents for the discrepancy between the proposed distribution and the updated requests of a human in the loop. Secondly, it is possible to reuse the usually discarded critic model in the process of ever continuous fine tuning.

B. DAY AND TIME MANAGEMENT

So far, we have only planned the day of the visit for the patient. However, in more complex scenarios it is necessary to take into account the day and hour of the visit (see Fig. 15).

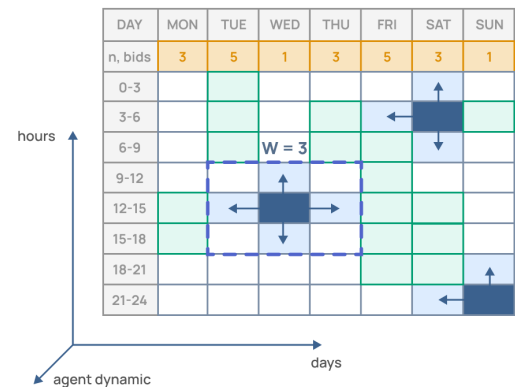


Fig. 15. Day and time exploration space

In this case, the game takes on a form reminiscent of classic maze pathfinding problems, where the free cells are the path and the occupied ones are the walls. The observation window in this case becomes a 3 by 3 square, and the number of actions increases to 5.

VII. CONCLUSION AND FUTURE WORKS

In conclusion, this study presents a novel multi-agent reinforcement learning approach designed to address the challenges of real-time scheduling in dynamic environments, particularly within the context of healthcare operations. The proposed system enhances both the efficiency and adaptability of scheduling processes by integrating human feedback through the Human-in-the-Loop paradigm and leveraging heavy-tailed reward distributions. The integration of the Belief-Desire-Intention (BDI) framework enhances the explainability of agent behaviors, thereby increasing the transparency and user-friendliness of the system. The experimental results illustrate the superiority of this approach over traditional methods, particularly in reducing patient waiting times and aligning with managerial preferences. Future work will focus on refining the explainability mechanisms and extending the applicability of the model to other domains requiring dynamic, human-centered scheduling solutions.

APPENDIX A

1.1 You are an expert system that analyzes the behavior of intelligent agents in a simulation. The simulation involves the use of multiple intelligent RL agents, each of which observes changes in the state of the environment and performs a transition action. The ultimate goal of the agents is to distribute themselves along a line representing the planning horizon in order to maximize the use of the clinic's resources while taking into account the needs of the patients. Estimate your understanding of the information given and provide the answer as following: "Affirmative or negative. In brief, [insert exactly one causal sentence]".

1.2 Read this Belief-Desire-Intention intelligent agent model. Beliefs structure – is a list of urgency, completeness, complexity, position, and occupancy. Where urgency of a surgery is varying from 1 (low) to 3 (high), completeness of information is varying from 0 (incomplete) to 1 (complete), complexity of a surgery varying from 0 (low) to 1 (high). Current agent position in the planning horizon of a one week, is varying from 0 (Monday) to 6 (Sunday). The masked occupancy of days is a dictionary in which each key for the number of days of the week is mapped to a discrete value that varying from a small negative value of -1 (unknown number of agents) to 12 (maximum number of agents). An instance of beliefs: 'agent_8': array([3, 1, 0, 2, -1, 4, 2, 2, -1, -1, -1]). Desires structure – is a continuous negative value. Represents the agent assessment of whether this new position is favorable or not. An instance of desires 'agent_0': -0.96. Intentions structure – is a discrete value from 0 to 2, where 0 – movement forward in the planning horizon (by Sunday), 1 – movement backward in the planning horizon (by Monday), 2 – no changes. An instance of intentions: 'agent_0': 1. Estimate your understanding of the information given and provide the answer as following: "Affirmative or negative. In brief, [insert exactly one causal sentence]".

1.3 The simulation logs are arranged as follows. The simulation initializes the distribution of requests by days randomly. At the beginning of each subsequent step of the simulation, the objective distribution of bids by day, as well as agents' beliefs, desires, and intentions, is initialized. Following the classical reinforcement learning cycle: first observations (beliefs) are formed, actions (intentions) are chosen, rewards are received (evaluating the desirability of the outcome), next observations (beliefs) are formed. The last step involves only observations, since it is the last step of the game. Estimate your understanding of the information given and provide the answer as following: "Affirmative or negative. In brief, [insert exactly one causal sentence]".

1.4 Read this fragment of the system logs: [LOG]. Estimate the completeness of the information given and provide the answer as following: "Complete or incomplete. In brief, [insert exactly one causal sentence]".

1.5 Please identify the [AGENT] and reconstruct its Belief-Desire-Intention reasoning model. Summarize the behavior of the agent and the motivational rules underlying it.

APPENDIX B

1.1 Analyze the following text and identify up to five key concepts. Present your answer as a comma-separated list. Each concept should be expressed in one or two words or be a specific named entity. Focus on the most essential ideas, terms, or themes: [PROMPT]

1.2 Analyze the text and identify words related to each key concept. Provide the output as a Python dictionary. Each key is one of the given concepts. The corresponding value is a list of words or phrases from the text that are closely related to or exemplify that concept. Key concepts: [CONCEPTS]. Text to analyze: [PROMPT]. Include only words and phrases that appear in the given text. If a concept has no related words in the text, use an empty list as its value. Ensure the dictionary is properly formatted and can be directly executed as Python code.

1.3 Assess your degree of confidence in what to put in masked fields. If you think there is more than one appropriate choice, choose a generalizing form. Start filling in the words at the end. Use only common lexicon. Provide only the filled text without any additional explanations. Text to analyze: [MASKED PROMPT].

APPENDIX C

1.1 Affirmative. In brief, the intelligent agents optimize resource allocation by positioning themselves along a planning horizon to balance clinic resources with patient needs.

1.2 Affirmative. In brief, the agent's beliefs, desires, and intentions guide its movement within the planning horizon to optimize its position based on surgery urgency, information completeness, complexity, and occupancy.

1.3 Affirmative. In brief, the simulation follows a reinforcement learning cycle where agents' actions are

influenced by their initialized beliefs, desires, and intentions, leading to observations and rewards until the final step.

1.4 Complete. In brief, the logs provide the initial and subsequent environment states, along with the beliefs, desires, and intentions of each agent, allowing for a full understanding of the simulation's progress.

1.5 Agent_10 starts with a moderately urgent and complex task scheduled for Monday. Initially, with incomplete

information, the agent assesses its situation as moderately unfavorable and decides to move forward in the week (from Monday to Tuesday). Upon moving to Tuesday, the agent reevaluates its beliefs, showing an updated but still incomplete understanding of the situation. Despite a slight improvement in its assessment (desire becoming less negative), the agent chooses to move backward, returning to Monday.

APPENDIX D

Prompt	Concepts	Key Words	Cosine Similarity				Jaccard similarity		Levenshtein similarity	
			Zero-shot		Few-shots		Zero-shot		Zero-shot	
			Hard	Soft	Hard	Soft	Hard	Soft	Hard	Soft
Appendix A Prompt 1.1	Intelligent agents	'RL', 'intelligent', 'agents'	0.3214	0.6237	0.3111	0.3984	0.3630	0.6207	0.3832	0.6771
	Reinforcement learning	'RL', 'agents'	0.3361	0.6237	0.7688	0.3984	0.3143	0.6207	0.5689	0.6771
	Simulation	'simulation'	0.5473	0.5840	0.3190	0.2621	0.5333	0.5938	0.5689	0.6911
	State transitions	'transition', 'state', 'action'	0.2748	0.8149	0.4486	0.3382	0.2647	0.8148	0.4192	0.7869
	Resource optimization	'themselves', 'line', 'representing', 'use', 'maximize', 'along', 'resources', 'planning', 'the', 'of', 'a', 'horizon', 'distribute', "clinic's"	0.3641	0.3150	0.2626	0.5183	0.3429	0.3077	0.3643	0.4792
Appendix A Prompt 1.2	Belief-Desire-Intention model	'model', 'agent', 'intelligent', 'Belief-Desire-Intention'	0.2841	0.5722	0.2841	0.5474	0.3243	0.5000	0.3321	0.4542
	agent beliefs	'structure', 'completeness', 'complexity', 'occupancy', 'Beliefs', 'position', 'masked', 'of', 'days', 'urgency'	0.4513	0.6708	0.4513	0.5518	0.4146	0.6111	0.5214	0.6044
	agent desires	'value', 'Desires', 'structure', 'negative', 'continuous', 'assessment', 'agent'	0.4885	0.8622	0.4885	0.8921	0.4103	0.8571	0.4786	0.9000
	agent intentions	'no', 'backward', 'structure', 'forward', 'changes', 'movement', 'Intentions'	0.6294	0.6852	0.6294	0.6852	0.5455	0.7105	0.4643	0.8071
	planning horizon	'week', 'planning', 'horizon', 'one', 'Sunday', 'Monday'	0.4138	0.5179	0.4202	0.4830	0.4000	0.4634	0.3643	0.5018
Appendix A Prompt 1.3	simulation	'simulation', 'of', 'step', 'the', 'initializes', 'logs'	0.2617	0.2369	0.0506	0.2635	0.2813	0.2727	0.3698	0.3708
	reinforcement learning	'learning', 'cycle', 'reinforcement'	0.2826	0.2470	0.5624	0.2420	0.2500	0.2727	0.4688	0.4944
	beliefs	'observations', 'beliefs'	0.5121	0.3298	0.3639	0.2187	0.4828	0.4138	0.5885	0.6461
	intentions	'actions', 'intentions'	0.2989	0.2462	0.5482	0.3797	0.3429	0.2432	0.5156	0.3539
	rewards	'rewards'	0.3063	0.1858	0.2299	0.3360	0.3125	0.2105	0.5052	0.3708
Appendix A Prompt 1.4	System logs	'[LOG]', 'logs', 'system'	0.5612	0.2060	0.5612	0.2060	0.4348	0.0893	0.6111	0.2310
	Log analysis	'Read'	0.2049	0.3584	0.2049	0.3584	0.0656	0.3043	0.1941	0.4196

Appendix A Prompt 1.5	Belief-Desire-Intention model	'Belief-Desire-Intention'	0.2950	0.7727	0.3618	0.2539	0.2237	0.6034	0.2978	0.5559
	agent analysis	'AGENT'	0.9780	0.8950	0.8521	0.6940	0.9608	0.8113	0.9801	0.7224
	reasoning reconstruction	'reconstruct'	0.3038	0.7133	0.2899	0.6464	0.1719	0.6071	0.3052	0.7093

ACKNOWLEDGMENT

This work was supported by the Ministry of Science and Higher Education of the Russian Federation, Goszadanie (State Assignment) No. 2019-1339.

REFERENCES

- [1] Abbeel P., Ng A.Y. Apprenticeship learning via inverse reinforcement learning. Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.
- [2] Albrecht S.V., Christianos F., Schäfer L. Multi-Agent Reinforcement Learning: Foundations and Modern Approaches. The MIT Press, 2024. ISBN: 9780262049375
- [3] Alelaimat A., Ghose A., Hoa Khanh Dam. Mining and Validating Belief-based Agent Explanations. University of Wollongong, 2023
- [4] Al-Hamadani M.N., et al. Reinforcement Learning Algorithms and Applications in Healthcare and Robotics: A Comprehensive and Systematic Review. Sensors. V.24 (8). 2461. (2024)
- [5] Amir O., Doshi-Velez F., Sarne D. Summarizing agent strategies. Autonomous Agents and Multi-Agent Systems. (2019) 33:628–644. <https://doi.org/10.1007/s10458-019-09418-w>
- [6] An G., et al. Direct Preference-based Policy Optimization without Reward Modeling. NeurIPS, 2023.
- [7] Anderson A., et al. Explaining Reinforcement Learning to Mere Mortals: An Empirical Study. arXiv:1903.09708v2 [cs.HC] 18 Jun 2019
- [8] Anjomshoe S., et al. 2019. Explainable agents and robots: Results from a systematic literature review. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, United States, 1078–1088.
- [9] Arakawa R., et al. DQN-TAMER: Human-in-the-Loop Reinforcement Learning with Intractable Feedback. arXiv:1810.11748. 28 Oct 2018.
- [10] arXiv:2310.18715v2 [cs.LG] 30 Mar 2024
- [11] Barabási, AL. The origin of bursts and heavy tails in human dynamics. Nature 435, 207–211 (2005). <https://doi.org/10.1038/nature03459>
- [12] Brännström M., et al. Metrics for Explainable Agent Behaviour. Micro-project WP5 AI Ethics and Responsible AI. HumanE-AI-Net, 2019.
- [13] Bratman M.E., Israel D.J., Pollack M.E. Plans and resource-bounded practical reasoning. Computational Intelligence 4(4), 349–355 (1988)
- [14] Brown T.B., et al. Language Models are Few-Shot Learners. arXiv:2005.14165v4 [cs.CL] 22 Jul 2020
- [15] Cayci S., Eryilmaz A. Provably Robust Temporal Difference Learning for Heavy-Tailed Rewards. 37th Conference on Neural Information Processing Systems (NeurIPS 2023).
- [16] Christiano P.F., et al. Deep Reinforcement Learning from Human Preferences. arXiv:1706.03741v4 [stat.ML] 17 Feb 2023
- [17] Ciatto G., et al. Towards XMAS: eXplainability through Multi-Agent Systems. CEUR workshop proceedings 2502, 40-53, 2019. 27, 2019.
- [18] Cruz F., Dazeley R., Vamplew P.. 2019. Memory-based explainable reinforcement learning. Australasian Joint Conference on Artificial Intelligence. Springer, 66–77.
- [19] de Silva L., Meneguzzi F., Logan B. BDI agent architectures: A survey. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. V. 7. 2020. pp. 4914–4921.
- [20] Dennis L.A., Oren N. Explaining BDI agent behaviour through dialogue. Auton Agent Multi-Agent Syst 36, 29 (2022). <https://doi.org/10.1007/s10458-022-09556-8>
- [21] Ehsan U., et al. Automated rationale generation: a technique for explainable AI and its effects on human perceptions. In Proceedings of the 24th international conference on intelligent user interfaces, 263-274. (2019)
- [22] Ge L., Zhou X., Li X. Designing Reward Functions Using Active Preference Learning for Reinforcement Learning in Autonomous Driving Navigation. Appl. Sci. 2024, 14(11), 4845; <https://doi.org/10.3390/app14114845>
- [23] Georgeff M., et al. (1999). The Belief-Desire-Intention Model of Agency. In: Müller, J.P., Rao, A.S., Singh, M.P. (eds) Intelligent Agents V: Agents Theories, Architectures, and Languages. ATAL 1998. Lecture Notes in Computer Science, vol 1555. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-49057-4_1
- [24] Gombolay M., et al. Apprenticeship Scheduling: Learning to Schedule from Human Experts. IJCAI, 2016
- [25] Gómez-Carmona O., et al. Human-in-the-loop machine learning: Reconceptualizing the role of the user in interactive approaches. Internet of Things. Volume 25, April 2024, 101048
- [26] Guidotti R., et al. 2018. A survey of methods for explaining black box models. Comput. Surveys 51, 5 (2018), 42 pages.
- [27] Hejna J., Sadigh D. Few-Shot Preference Learning for Human-in-the-Loop RL. 6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.
- [28] Lage I., et al. Toward robust policy summarization. AAMAS '19: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. Pp. 2081 - 2083
- [29] Langley P., et al. 2017. Explainable agency for intelligent autonomous systems. In Proceedings of the 29th Innovative Applications of Artificial Intelligence Conference.
- [30] Liang X.-X., et al. Reward Uncertainty for Exploration in Preference-based Reinforcement Learning. International Conference on Learning Representations. 2022
- [31] Lowe R., et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. arXiv:1706.02275v4 [cs.LG] 14 Mar 2020
- [32] Lundberg S., Su-In Lee (2017). A unified approach to interpreting model predictions, Advances in neural information processing systems 30.
- [33] Mangalampalli S., et al. Multi Objective Prioritized Workflow Scheduling Using Deep Reinforcement Based Learning in Cloud Computing. IEEEAccess. 8 January 2024
- [34] Monacia M., Agasuccia V., Grani G. An actor-critic algorithm with policy gradients to solve the job shop scheduling problem using deep double recurrent agents. <https://arxiv.org/pdf/2110.09076> 21 Nov 2023
- [35] Mosqueira-Rey E., Hernández-Pereira E., Alonso-Ríos D. et al. Human-in-the-loop machine learning: a state of the art. Artif Intell Rev 56, 3005–3054 (2023). <https://doi.org/10.1007/s10462-022-10246-w>
- [36] Muslimani C., Taylor M.E. Leveraging Sub-Optimal Data for Human-in-the-Loop Reinforcement Learning. arXiv:2405.00746v1 [cs.LG] 30 Apr 2024
- [37] Panzer M., Bender B. Deep reinforcement learning in production systems: a systematic literature review. International Journal of Production Research. V. 60 (13), 4316-4341. (2022)
- [38] Pu Y., Li F., Rahimifard S. Multi-Agent Reinforcement Learning for Job Shop Scheduling in Dynamic Environments. Sustainability 2024, 16(8), 3234; <https://doi.org/10.3390/su16083234>
- [39] Rafailov R., et al. (2023) Direct Preference Optimization: Your Language Model is Secretly a Reward Model". arXiv:2305.18290 [cs.LG].
- [40] Retzlaff C.O., et al. Human-in-the-Loop Reinforcement Learning: A Survey and Position on Requirements, Challenges, and Opportunities. JAIR, vol.79(2024)

- [41] Rzeszucinski P. AI, humans and loops. Feb 29, 2024. https://medium.com/@pawel.rzeszucinski_55101/ai-humans-and-loops-04ee67ac820b
- [42] Sado F., et al. 2023. Explainable Goal-driven Agents and Robots - A Comprehensive Review. *ACM Comput. Surv.* 55, 10, Article 211 (February 2023), 41 p. <https://doi.org/10.1145/3564240>
- [43] Selvaraju R.R., et al. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 618–626.
- [44] Sequeira P., Gervasio M. 2019. Interestingness Elements for Explainable Reinforcement Learning: Understanding Agents' Capabilities and Limitations.
- [45] Shu T., Xiong C., Socher R. Hierarchical and Interpretable Skill Acquisition in Multi-task Reinforcement Learning. *arXiv:1712.07294v1 [cs.AI]* 20 Dec 2017
- [46] Varga J., et al. Scheduling jobs using queries to interactively learn human availability times. *Computers & Operations Research* Volume 167, July 2024, 106648. <https://doi.org/10.1016/j.cor.2024.106648>
- [47] Vázquez A., et al. Modeling bursts and heavy tails in human dynamics. *Phys Rev E Stat Nonlin Soft Matter Phys.* 2006 Mar; 73(3 Pt 2):036127.
- [48] Walsh S.E., Feigh K.M. Differentiating 'Human in the Loop' Decision Process. 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 2021, pp. 3129-3133, doi: 10.1109/SMC52423.2021.9658802
- [49] Wan L., et al. An effective deep actor-critic reinforcement learning method for solving the flexible job shop scheduling problem. *Neural Computing and Applications* 36(20):1-23. April 2024
- [50] Wells L., Bednarz T. Explainable ai and reinforcement learning—a systematic review of current approaches and trends. *Frontiers in artificial intelligence*, 550030. (2021)
- [51] Williams S., Crouch R. Emergency department patient classification systems: A systematic review. *Accident and Emergency Nursing*. Volume 14, Issue 3, July 2006, Pages 160-170
- [52] Winikoff M., Sidorenko G. Evaluating a Mechanism for Explaining BDI Agent Behaviour. *Explainable and Transparent AI and Multi-Agent Systems: 5th International Workshop, EXTRAAMAS 2023, London, UK, May 29, 2023, Revised Selected Papers*. Pages 18 – 37. https://doi.org/10.1007/978-3-031-40878-6_2
- [53] Wong, A., Bäck, T., Kononova, A.V. et al. Deep multiagent reinforcement learning: challenges and directions. *Artif Intell Rev* 56, 5023–5056 (2023). <https://doi.org/10.1007/s10462-022-10299-x>
- [54] Wu J., et al. Toward Human-in-the-Loop AI: Enhancing Deep Reinforcement Learning via Real-Time Human Guidance for Autonomous Driving. *Engineering*. V. 21, February 2023, Pp. 75-91
- [55] Wu X., et al. A survey of human-in-the-loop for machine learning. *arXiv:2108.00941v3 [cs.LG]* 26 Apr 2022
- [56] Xue W., et al. Reinforcement Learning from Diverse Human Preferences. *IJCAI*, 2024.
- [57] Yan E., et al. Towards a Multi-Level Explainability Framework for Engineering and Understanding BDI Agent Systems. *WOA2023: 24th Workshop From Objects to Agents*, November 6–8, Rome, Italy
- [58] Yu Lu et al. Deep reinforcement learning based optimal scheduling of active distribution system considering distributed generation, energy storage and flexible load. *Energy*, V. 271, 15 May 2023, 127087
- [59] Zhang G., Kashima H. Learning state importance for preference-based reinforcement learning. *Mach Learn* 113, 1885–1901 (2024). <https://doi.org/10.1007/s10994-022-06295-5>
- [60] Zhang K., Yang Z., Basar T. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *arXiv:1911.10635v2 [cs.LG]* 28 Apr 2021
- [61] Zhang, J.; Ding, G.; Zou, Y.; Qin, S.; Fu, J. Review of job shop scheduling research and its new perspectives under Industry 4.0. *J. Intell. Manuf.* 2019, 30, 1809–1830
- [62] Zhu J., et al. Robust Offline Reinforcement Learning with Heavy-Tailed Rewards. *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, PMLR 238:541-549, 2024.