Towards Explainable Recommendations of Resource Allocation Mechanisms in On-Demand Transport Fleets

Alaa DAOUD[1] Hiba ALQASIR[2] Yazan MUALLA[3]
Amro NAJJAR[4] Gauthier PICARD[5] and Flavien BALBO[1]

[1] Mines Saint-Étienne, CNRS, UMR 6158, LIMOS - Institut Henri Fayol, Saint-Étienne, France
[2] Université Lyon, UJM-Saint-Etienne, CNRS, Institut Optique Graduate School, Laboratoire Hubert Curien UMR 5516, Saint-Étienne, France
[3] CIAD, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France
[4] University of Luxembourg
[5] ONERA/DTIS, Université de Toulouse, France

alaa.daoud@emse.fr, h.alqasir@univ-st-etienne.fr, yazan.mualla@utbm.fr, amro.najjar@uni.lu, gauthier.picard@onera.fr, flavien.balbo@emse.fr

Abstract. Multi-agent systems can be considered a natural paradigm when modeling various transportation systems, whose management involves solving hard, dynamic, and distributed allocation problems. Such problems have been studied for decades, and various solutions have been proposed. However, even the most straightforward resource allocation mechanisms lead to debates on efficiency vs. fairness, business quality vs. user experience, or performance vs. robustness. We aim to design an analytical tool that functions as a recommendation system for on-demand transport (ODT) authorities. This tool recommends specific allocation mechanisms that match the authority's objectives and preferences to solve allocation problems for particular contextual scenarios. The paper emphasizes the need for transparency and explainability of resource allocation decisions in ODT systems to be understandable by humans and move toward a more controllable resource allocation. We propose a multi-agent architecture to meet these requirements.

Keywords: : Multi-agent Systems · Explainable Artificial Intelligence · Intelligent Transport Systems · Resource Allocation.

1 Introduction

Today's transport systems are constructed of complex, large-scale interactions in a dynamic environment. In on-demand transport (ODT) systems, a fleet of vehicles is distributed in an urban area to meet potential requests to transfer people or goods between origin and destination locations. Agent-based and multi-agent systems provide a suitable scheme to model such complexity. In multi-agent models of ODT systems, vehicles are represented by agents that are mobile in their spatial environment and may have communication abilities. The system's spatial environment consists of a network of

roads, facilities, and urban infrastructure artifacts. The agents may have the possibility to communicate with each other and with other system entities to share information and coordinate their actions [1].
The allocation problems are major issues in the management of ODT systems. They have been studied for decades, and various solutions have been proposed. However, even the most straightforward cases of resource allocation lead to debates on efficiency versus fairness [27], business quality versus user experience [35], and performance versus robustness [24].

We are interested in building an analytical tool that functions as a recommendation system for resource allocation methods for ODT scenarios. This tool takes as input the set of parameters for the scenario (vehicle fleet properties and request distribution model), user's objective function, and preferences, in addition to the environment model (road network and traffic model).
This system simulates the problem scenario and its solutions with different classes of AI methods, then produces to the user the recommended solution model (the solution method and its tuned parameters) that produce results matching the user objective and preferences for the input scenario.

In future Artificial Intelligence (AI) systems, it is vital to guarantee a smooth human-agent interaction, as it is not straightforward for humans to understand the agent's state of mind, and explainability is an indispensable ingredient for such interaction [29]. Recent works in the literature highlighted explainability as one of the cornerstones for building trustworthily responsible and acceptable AI systems [13, 26, 32, 34]. Consequently, the emerging research field of eXplainable Artificial Intelligence (XAI) gained momentum both in academia and industry [21, 4, 7]. XAI is allowing, through explanations, users to understand, trust, and effectively manage the next generation of AI solutions [22].

Providing users with some form of control over the recommendation process can be realized by allowing them to tell the system what they like or by engaging them in adjusting the recommendation profile to synthesize recommendations from different sources [39]. High-quality explanations allow a better understanding of the results and help the user to make the right decisions. Reliable answers increase confidence in the system, while explanations that reflect system inaccuracies allow the user to modify the system's reasoning or control the weighting parameter that reorganizes or regenerates recommendations.

2 About the Need for Explainability

The human perspective is what differentiates ODT from most routing and transport problems. In addition to the technical factors, the quality of the service is influenced by human satisfaction factors, including the stability of service quality, service availability, wait-time, information privacy, passengers' special constraints, and preferences [10].

The following examples show that global system decisions may not fit all stakeholders' preferences: a decision may make some people dissatisfied.

Scenario 1: Dial-a-ride in rush hours. At rush hours, taxi-ride demand is usually concentrated at specific parts of the city, e.g., city center and train stations, as seen in

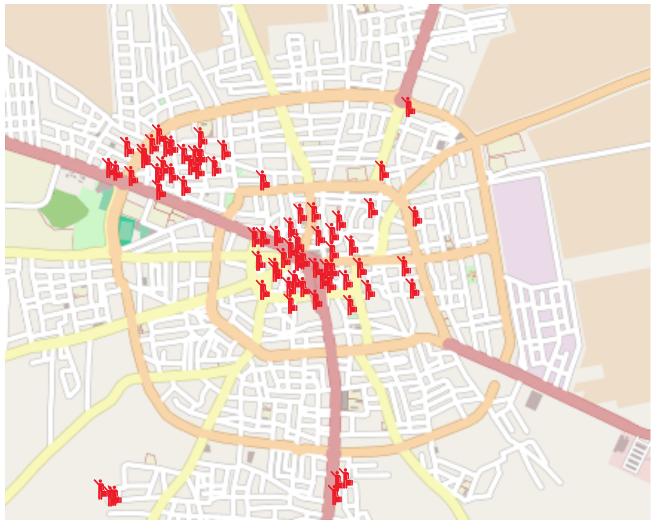


Fig. 1: Passenger request distribution at rush hours.

Figure 1. The objective of the transport authority is to maximize the number of satisfied requests while reducing operational costs. An efficient allocation mechanism will dispatch as many vehicles as possible to the crowded areas to serve passengers, prioritizing the requests whose destinations are near other crowded areas. As a consequence, in this example, most of the vehicles move back and forth between the two areas, which reduces the chance of far passengers and makes them wait for a long time for being served, regardless of the urgency level of their requests that may be higher than those who do their ordinary work-home trips from the city center.

Scenario 2: Emergency management ODT. The example of Figure 2, introduced by [2] represents a disaster management situation. However, this kind of emergency transport can be modeled as an ODT system [36, 40, 5]. In this example, a failure in facility X leads to a leak of toxic substances. The leakage grows over time and threatens both communities A and B. The inhabitants of these communities need to be relocated to refuge R as soon as possible. A fleet of shuttles is available to relocate people. However, suppose that the fleet's size is not large enough to evacuate either community in one hour. Because of the wind direction, the time it takes for the substance to reach community A is double that of community B; however, community A's toxic density will be higher than in community B (assuming the density degrades with distance). Also, community B's population is three times the population of community A. The round-trip time from community A to the refuge is twice the round-trip time from community B to the refuge. In other words, a shuttle assigned to community B can carry twice the number of evacuees compared to the same shuttle assigned to community A. If the goal is to maximize the number of evacuees moved to the refuge within one hour, the answer would be to assign the entire fleet to community B since the round trip time is shorter for this community. However, if the goal is to evacuate high-risk individuals as quickly

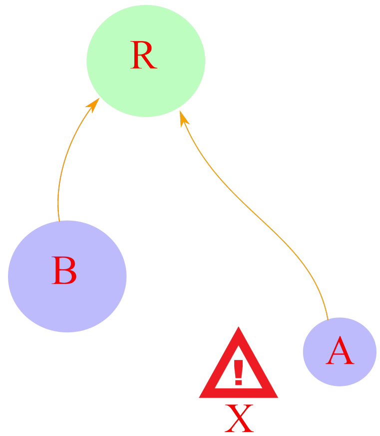


Fig. 2: An example of an emergency scenario.

as possible, the answer would be to assign the entire fleet to community A. While both of these responses seem correct for the corresponding objective, neither seems fair.

Providing explanations for the system decision may increase people's satisfaction [6], and maintain the AI system's acceptability. When a recommendation mechanism is too complicated for lay users, the system may need to justify why the recommendation has been made [14, 38]. The EU General Data Protection Regulation introduces a right of explanation for citizens to obtain "meaningful information about the logic involved" for automated decisions [19]. Generating explanations of autonomous decisions in multi-agent environments is even more difficult than providing explanations in other contexts [25]. In addition to identifying the technical reasons that led to the decision, it is necessary to convey the agents' preferences. It is necessary to decide what to reveal about other agents' preferences to increase user satisfaction while considering other agents' privacy, and how those features led to the final decision.

To provide useful explanations, it is necessary to identify the features of the context and decisions relevant to a specific user. Given these features, other relevant agents' preferences should be identified, and any relevant statements that touch on important concepts such as fairness should be generated. Using these features, preferences, and concepts, various explanations could be generated using subsets of them. The selected subset should be transferred in a certain communication form. The personalization of explanations could also be used at this stage since explanations are subjective and depend on multiple factors [37]. As to personalize explanations, there is a need to build a user, or mental model [23] that influences the generation of explanations.

In our resource allocation scenario in vehicle fleets, the allocation process can provide a set of constraints that lead to the proposed allocation. It will be necessary to identify the relevant constraints and generalize statements related to other agents' preferences and general system constraints related to fairness [28]. Then, we can use user satisfaction models to choose the best constraints, and generalized statements to present.

3 AV-OLRA Metamodel

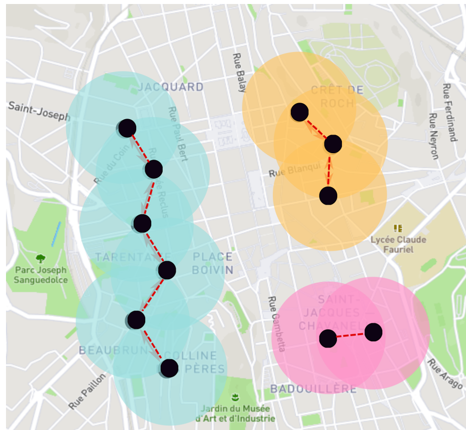


Fig. 3: AV-OLRA's dynamic composition of connected sets.

In previous work, we presented *AV-OLRA* a metamodel for resource allocation in autonomous vehicle fleets [12]. In this model, an autonomous vehicle is any vehicle that can make autonomous decisions, and interact with other entities in the surrounding environments, besides its self-driving capabilities. We consider vehicles communicate locally within limited ranges, and can pass transitive messages.

Connectivity between two vehicles is achieved if the distance between them is less than or equals their communication range. However, as the vehicles' communication range is limited, and to maximize their connectivity, two vehicles can be connected by transitivity if both are connected to another vehicle. We define the concept of *connected set* (CS) as a dynamic set of entities that can communicate with each other either directly or by transitive message passing. CSs are composed, split, and merged at run-time based on vehicles' movement as shown in Fig. 3. When the communication range is long enough, all vehicles in some urban area can communicate globally i.e. all the vehicles belong to one CS.

Considering the transport requests as dynamic resources that can be consumed by or allocated to vehicles, the *AV-OLRA* metamodel is formulated as:

$$\textit{AV-OLRA} := \left(\mathcal{R}, \mathcal{V}, \mathcal{G}, \mathcal{T}\right) \tag{1}$$

where $\mathcal{R}$ defines a dynamic set of resources that occur to be available for a specific time window at the time of execution, representing passengers' requests; the set of consumers $\mathcal{V}$ represent a fleet of m autonomous vehicles that are mobile and can only communicate within a limited range; $\mathcal{G}$ is a directed graph representing the urban infrastructure network that defines the problem spatial environment, with $\mathcal{N}$ the set of nodes, and $\mathcal{E}$ the set of edges, $e_{ij} \in \mathcal{E}$ is the edge between the nodes i and j, ω is a valuation function that associates each edge $e \in \mathcal{E}$ with the value ω_e based on a temporal distance measure (e.g., average driving time in minutes), which will be used to calculate the operational costs of vehicle trips; $\mathcal{T}$ defines the temporal dimension of the problem as a discrete-time horizon.

Instantiating this metamodel by defining the feature model of these components results in an *AV-OLRA* problem model while defining these features' exact values leads to an *AV-OLRA* problem instance. A problem model can be solved with different solution models. A solution model defines the strategy by which the allocation is computed. Applying a strategy X to a problem instance I results in assigning values to allocation variables, which means achieving a feasible solution if it exists.

Example. The dial-a-ride problem (DARP) model can be defined in some urban area u by defining the u's urban network features (number of nodes, edges, facilities, etc.). The fleet's vehicles are taxis with a set of attributes for capacity as the number of seats, average speed, energy consumption, and communication range. The requests are trip requests with attributes for some passengers, pick-up and drop-off locations, time-window, and budget. An instance of this problem model is defined by the exact values of node locations, edge distances, number of vehicles in the fleet with their capacities speed, range, and initial locations, in addition to the set of passenger requests and the time slot in which the scenario takes place (the time horizon of the problem instance). Here the allocation variables are the vehicles' schedules; for each vehicle, we have a schedule as a list of couples $(location, time)$ defining the locations that the vehicle needs to visit (for pick-up or delivery) and their potential visit time.

4 Explainable MAS for AV-OLRA recommendation

In this section, we introduce *EX-AV-OLRA*, an extension to *AV-OLRA* metamodel with explainability-related components. We present a multi-agent model for an explainable recommendation system that realizes the *EX-AV-OLRA* model. *EX-AV-OLRA* model is formulated as:

$$\textit{EX-AV-OLRA} := \big(\mathcal{R}, \mathcal{V}, \mathcal{G}, \mathcal{T}, \mathcal{X}\big) \tag{2}$$

Where $\big(\mathcal{R}, \mathcal{V}, \mathcal{G}, \mathcal{T}\big)$ define an *AV-OLRA* and $\mathcal{X}$ defines the explaining mechanism.

We aim to design a recommender system in which a human user sets the scenario parameters to create an AV-OLRA instance, setting objective and utility preferences. The system's output is a recommendation to use the solution method that is the best match to user preferences, supported with multi-level explanations of why particular methods are recommended and why others are discouraged.

The generic multi-agent model of *AV-OLRA* consists basically of *Autonomous Vehicle (AV)* agents who are mobile in their spatial environment to serve trip requests and may communicate within a limited range with other agents and surrounding artifacts. We can distinguish three different *sub-behaviors* (*acting*, *communicating*, and *planning*) shown in Fig. 4.

The multi-agent model for the explainable recommender system extends the previous model. An additional agent type *Monitor Agent (MA)* plays the role of proxy for*AV*s to produce human-readable personalized explanations for the recommended methods.

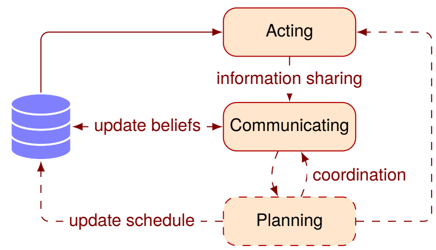


Fig. 4: Generic AV agent behavior in *AV-OLRA*.

Unlike the inter-*AV*s' limited-range communication model, the *MA* can interact with *AV*s globally (See Fig. 5).

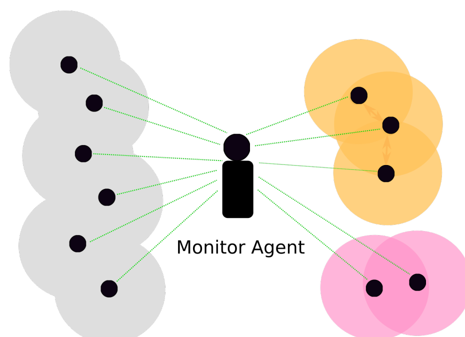


Fig. 5: *MA* and *AV* agents interaction.

This interaction is only to monitor the performance of *AV* agents and logging the explanations of their actions during the simulation. To reflect the behavior of *AV*s in real scenarios, this global interaction means should never be used for communication between *AV*s.

4.1 AV Agents' Behavior

An *AV* agent repeatedly performs the following actions representing the behavior of a vehicle in the system: 1. read the received messages and update the context (*communicating* sub-behavior), 2. choose the locations to visit (*planning* sub-behavior), 3. act by performing a driving action (*acting* sub-behavior), 4. broadcast context information (*communicating* sub-behavior).

The *acting* and *communicating* sub-behaviors are always the same whatever the problem instance and whatever the chosen solution model. The *AV* agent can perform four actions (moving, waiting-for/marauding requests, picking-up, and dropping off) as a transport vehicle. As a communicating agent, an *AV* can join/leave a connected set and send, receive, or broadcast messages. The communication behavior depends mainly on the value of the communication range, which is an attribute of the scenario. The *planning* sub-behavior represents how an *AV* obtains its dynamic schedule. This behavior depends on the allocation mechanism, which is specific to each coordination mechanism that defines the solution model. A *coordination mechanism* is defined by three components $\langle DA, AC, AM \rangle$, where DA denotes the level of decision autonomy which is either centralized (C) or decentralized (D); AC denotes the agents' cooperativeness level with (S) or without (N) sharing of schedule information, and AM is the chosen allocation mechanism (e.g. "Auctions", "Greedy", "DCOP", etc).

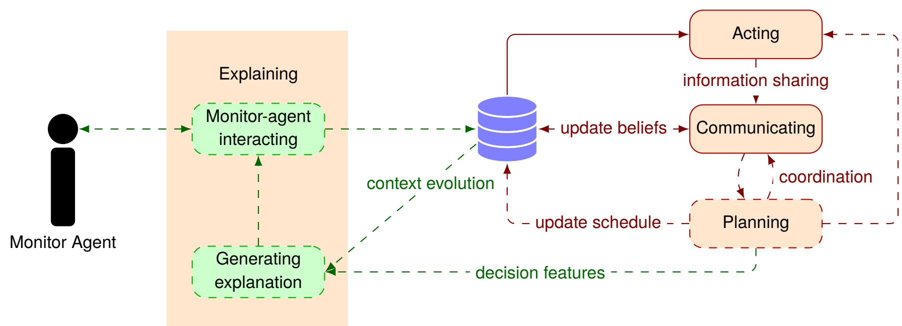


Fig. 6: Explainable AV agent behavior in *EX-AV-OLRA*

In this work, we propose to add another sub-behavior (the *explaining* sub-behavior) to the *AV* model. This sub-behavior consists of two phases *generating explanation* and *monitor-agent interacting* as shown in Fig. 6.

Generating explanation phase is triggered whenever a decision is taken (in *planning* sub-behavior). The *AV* gathers all information related to the taken decision (the leading constraints, context information, potential improvement in the solution quality, etc.), in addition to the changed decision variables and their values. This information, together with the contextual data gathered in the previous steps from the agent belief base, are used to generate an understandable explanation for the taken decision. When the explanation is generated, the agent moves to the *monitor-agent interacting* phase. In the *monitor-agent interacting* phase, the generated explanations are sent to the monitor agent and stored in the *AV* belief base. To reflect the behavior of *AV*s in real scenarios, the *MA* should never play the role of communication mediator between *AV*s.

The set of possibly explainable actions and decisions depend basically on the chosen solution model. Table 1. lists some examples of solution models in line with their possibly explainable decisions.

4.2 *Monitor Agent*'s Behavior

The role of the *MA* is to be a proxy between *AV*s and the user. It interacts with the user via dialogues to build a user profile that simulates the user preferences and objectives. *MA*

Solution	DA	AC	AM	**Explanation examples**
Selfish	D	N	Greedy	Why prioritizing a specific request?
Dispatching	C	S	MILP	Which constraints are violated?
Market	D	S	Auctions	How winner determination computed?
				Why accepting some trade options?
Cooperative	D	S	DCOP	What are individual costs and utilities?

Table 1: Examples of solution models and what should be explained.

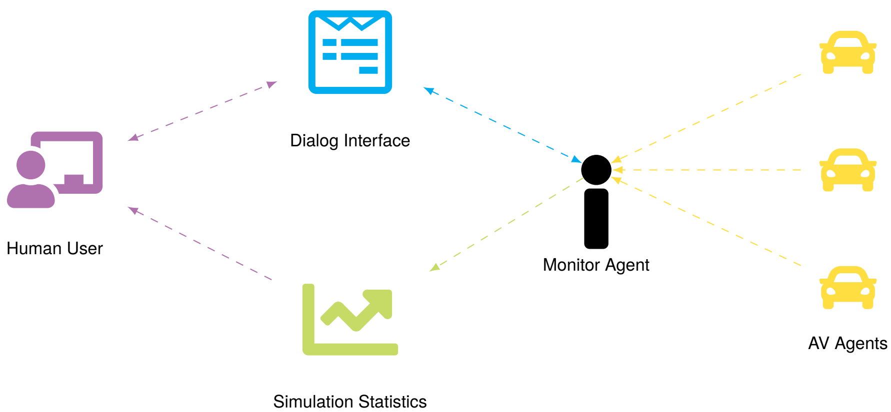


Fig. 7: The interacting behavior of the *Monitor Agent*.

could be formed in a group of agents for fault tolerance and backup reasons and to avoid having a bottleneck in the model. Additionally, members inside this group may execute different explanations behaviors and interact/cooperate to provide the explanation to the human. The most important point is to have one interface with the human user to avoid overwhelming her/him with many interfaces. We can look at the assistant agent as the personal assistant of the human that could be embedded in his/her smartphone for example. Therefore, and even with a group of assistant agents, the interface with the human is preferably unified through one agent as a representative of the group.

MA gathers the statistics of decisions and their explanations from *AV*s. It aggregates these explanations in several abstraction levels. Following a similar approach of [33], the *MA* builds a multilevel explanation tree. The leaves of this tree correspond to particular agent's actions explanations. The root corresponds to the global abstract explanation for the final recommendation, and intermediate levels correspond to explanations for the evolution of evaluation metrics. At the end of the simulation scenario, it ranks the different solution methods based on their matching to the user profile providing a summary explanation for the ranking decision. The user could ask for a detailed explanation –to handle this, the *MA* defines a new grain size for selecting the right level of explanation that is communicated to the user. While the user is asking for more details, *MA* proceeds from the root to leaves gradually, providing at every step the corresponding level of explanations. It stops when the user stops asking or reaching the leaves representing the atomic details that can not be expanded. The next section discusses how an *MA* computes its recommendation.

4.3 Computing the recommendations

The objective of *MA* is to assign values to its decision variable by the end of the scenario execution. *MA* has three sets of variables: profiles, recommendation and explanation variables. The recommendation variables are the ranking values for the different candidate methods. The explanation variables aggregate individual *AV*s' explanations and *MA*'s reasoning on the evolution of the evaluation metrics during the execution. The profile variables define a model based on the available features of allocation methods that match the user-defined features profile. If we manage to get such a model, then making recommendations for a user is relatively easy. We need to look at the user profile and compute its similarity to the different candidate methods. The candidates are then ranked based on their similarity value.

The user profile u in the set of user profiles U is represented by a vector of n features $u = [u_1, ..., u_n]$ defines the user's preferred values for the different evaluation metrics. Given that the system implements k solution methods that are potential candidates, this set of candidate methods is represented by $M = \{m_1, ..., m_k\}$; $m \in M$ is the feature vector of the candidate $m = [f_1^m, ..., f_n^m]$ we define a distance $dist$ function to calculate the n-dimensional euclidean distance between feature vectors:

$$dist : U \times M \to \mathbb{R}+$$

$$dist(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

A perfect-match method m' to user profile u if exists, will have $dist(u, m') = 0$ otherwise the following similarity function will be used to rank the candidate methods:

$$sim : U \times M \to [0, 1]$$

In its simplest form, sim function is the inverse of $dist$.

$$sim(x, y) = \frac{1}{dist(x, y)}$$

so that the highest recommended method m' to user u is the one with higher value of $sim(u, m')$.

4.4 Creating and communicating the explanations

As seen before, we need explanations that are scalable for multiple levels, we can distinguish two types of actions to be explained: the *AV*'s individual decisions and the aggregated decision by *MA*.

The classical approach in XAI is the straightforward design of interpretable models on the original data to reveal the logic behind actions proposed by the system. State-of-the-art interpretable models, including decision trees, rules, and linear models; are considered to be understandable and readable by humans. This applies to the individual decisions of *AV*s in our model, every *AV* is an autonomous agent having predefined interpretable behavior, and can justify his decisions with their technical and social reasons (based on its believes of itself and the context).

Another XAI approach is the post hoc interpretability, given the decisions made by the system, the problem consists of reconstructing an explanation to make the system intelligible without exposing or modifying the underlying model internally. The generation of explanations is an epistemic and computational action, carried out on-demand according to the current state of a model, and meta-knowledge on the functionalities of the system. It is intended to produce a trustworthy model based on features or exemplars. This applies to the aggregation of decisions made by *MA*, to the statistics-based matching, and to the recommendations.

An explanation can indeed be in any type of interaction. The advantage of human-like interaction is that it provides to the user higher levels of satisfaction, trust, confidence, and willingness to use autonomous systems. For this reason, many techniques have been developed to generate natural language (NL) descriptions of agent behavior and detected outliers or anomalies. This entails answering questions such as, why did an agent choose a particular action? Or what training data were most responsible for that choice? The internal state and action representations of the system are translated into NL by techniques such as recurrent neural networks [17], rationale generation [18], adding explanations to a supervised training set such that a model learns to output a prediction as well as an explanation [9].

5 ExDARP: a Use Case of Explainable Decisions in Decentralized DARP

In DARP scenarios, a fleet of vehicles is distributed through an urban network to satisfy passenger dynamic trip requests. A human *User* sets the scenario parameters, including fleet size, vehicle characteristics, request distribution model in addition to the user-defined objective, and utility preferences.

The *User* of the system is a representative of the transport authority; he wants to know the best solution method for solving the problem regarding the authority's preferences and the actual context parameters. The *Monitor Agent* uses this information to build the user profile and the scenario profile, which is then passed to the simulator. The system runs simulations with several solution methods; at each simulation tick, a snapshot of the problem context is solved with the different solution methods, explanations for local decisions are logged, and statistics for evaluation metrics are computed. At the end of the scenario execution, the results from different solution methods are compared, assessed in line with the statistics and user preferences.

Solution methods with the highest match to user preferences are recommended to the *User* (abstractly, macro-level) explaining the features of these methods –e.g., greedy method favors closer requests with short distances, which means lower operational cost.

The *User* can also ask the system to monitor why some solution method is not suitable for his scenario –e.g., centralized dispatching requires continuous communication between vehicles and the dispatching portal, this consumes bandwidth in dynamic settings, making it unsuitable for scenarios with limited communication. – or, centralized dispatching requires exponential execution time to reach an optimal solution, it is unsuitable for emergency response scenarios. If the *User* requires fine-grained details, he can trace the evolution of evaluation metrics and ask for explanations for remarkable spots –e.g., a valley in the QoS chart followed by a peak can be explained as follows: at that time slot, 70% of vehicles were carrying passengers on the route to their far destinations, so for a while, only a low number of requests is satisfied, which means low values of QoS for a while; when these long trips ended one by one, the number of satisfied requests increases rapidly causing the peak in QoS. The *User* can continue asking for finer-grained details until reaching an explanation for individual vehicle actions.

Fig. 8 illustrates a simple instance of DARP in a part of an urban network where the two vehicles V_1 and V_2 located in A and B are available and aware of a passenger request d_1 to travel from C to H which is announced at t_1. Considering the symmetric weights on edges represents the edge crossing operational cost in terms of the average time that a vehicle needs to move between its two ends. In the absence of central authority, vehicle agents should act autonomously and make decentralized decisions to find a solution in which the request is allocated to one of them. One of the common decentralized solution methods to such allocation problems is the market-based allocation mechanism [3, 11, 15] in which vehicle agents coordinate their decisions via auctions. Both V_1 and V_2 can

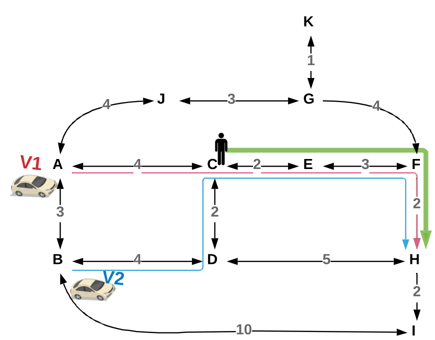
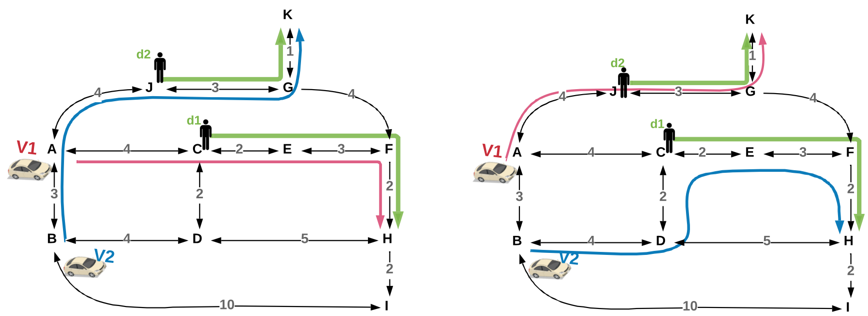


Fig. 8: A simple instance of DARP

serve d_1 without violating its constraints. To decide who serve it, vehicles instantiate an auction on d_1 where V_1's offer Bid_{V_1} is to serve d_1 with total operational cost of 11, while V_2 offers to serve it with 13. V_1 can explain its offer as "Serving d_1 costs me 11 time units because: Following the shortest paths, reaching C (the pick-up location of d_1) from my current location A requires 4 time units, and reaching H (the drop-off location of d_1) from C requires at least 7 time units". In the same way V_2 can explain: "Serving d_1 costs me 13 time units because: Following the shortest paths, reaching C (the pick-up location of d_1) from my current location B requires 6 time units, and reaching H (the drop-off location of d_1) from C requires at least 7 time units".

V_1 wins the auction on d_1, Explaining “how winner determination computed?" (See line 3 in Table 1.) can be: “V_1's offer has a lower cost than V_2's. The lower the cost is, the better the QoB achieved." Or, “V_1 reaches pick-up location 2 time units earlier than V_2 which means lower waiting time so better QoS".



(a) strait-forward auction (b) schedule exchange auction

Fig. 9: Combinatorial auctions

After d_1 is added to V_1's schedule, a new request d_2 for a passenger trip from J to K specifying the pick-up time-window at J as $tw_{d_2} = [5, 15]$ is announced. V_2 can offer to serve it with 11 time unit (pick-up in 7 time units and delivery after 4), as shown in Figure 9a. While V_1 cannot offer a reasonable bid, it explains: “I am committed to d_1, the earliest pick-up I can offer for d_2 is t_{27} which violates d_2's time-window constraint".

Some run-time optimization protocols such as ORNInA [11] and the sliding-horizon method [3] allow vehicles to exchange their scheduled requests if this exchange improves the global quality of the solution. V_2 can offer to pull d_1 from V_1's schedule for extra cost of 2 time units, so that V_1 can bid for d_2 with only 8 time units without violating its time-window constraint Figure 9b.
This exchange should be explained because it increases d_1's waiting time by 2 time units. In ORNInA optimization protocol, V_1 should accept V_2's pull bid, and then an explanation to “why accepting the exchange?" would be: “Abandoning d_1 in favor of V_2 decreases the global operational cost value by 1. It also decreases the accumulated waiting time by 1”

6 Related Work

Basically, one can distinguish between post hoc interpretability and the design of the explanation. In the former, the task is to create an explanation of the decision made by the system. While in the latter, the task is to design an interpretable model with its explanations.

Post-hoc interpretability approaches are divided into three categories depending on the motivation for having an explanation: model explanation, outcome explanation, and model inspection. The model explanation problem is to understand the reasoning of the system as a whole, while, the outcome explanation problem is to provide an explanation of the output of an intelligent system on a given input instance. Finally, the model inspection problem lies between the two previous problems [21].

We are interested in the outcome explanation problem because in this type it is not necessary to explain all the logic behind the system, only explaining why a certain decision has been returned for a particular input. Approaches that solve the outcome explanation problem yield a locally interpretable model that can explain the system output for a specific input in human-readable terms. For example, the locally interpretable model might be a decision tree constructed from a neighborhood of the instance in question, and an explanation might be the path of the decision tree followed by the attribute values of the instance. The proposed agnostic solutions to the problem of outcome explanation are generalizable by definition. Thus, in some cases, they can also be used for diverse data types. A recent proposal is LORE (LOcal Rule-based Explanations) [20], which overcomes previous solutions in terms of performance and clarity of the explanations. LORE first relies on a local interpretable predictor learned on a synthetic neighborhood generated by a genetic algorithm. Then, it generates an explanation which comprises (i) a decision rule, which explains the reasons for the decision; (ii) a set of counterfactual rules, suggesting changes in the functionality of the instance that lead to a different outcome. Another agnostic explanation method is MAME (Model Agnostic Multilevel Explanations) [33], which takes as input a post-hoc local explainability technique and an unlabeled dataset. Then generates multiple explanations for each of the examples corresponding to different levels of cohesion between explanations of the examples. As mentioned earlier, this type of methods could be applied to explain the the aggregation of decisions made by *MA* .

The work of [8] argues that explainable planning can be designed as a wrapper around an existing planning system that uses the existing planner to answer challenging questions. To do so, they presented a prototype framework for explainable planning as a service.
In the task allocation domain, [41] introduces *AlgoCrowd* that offers efficient and explainable AI task allocation optimization designed to emphasize on fair treatment of workers, whiling reducing managers' workload to find suitable workers for tasks. they aimed at guaranteeing that workers of similar capability and productivity receive equitable incomes in the long run. This objective is translated into minimizing the workers' regret when their incomes are compared to similar peers.
In the domain of passenger transport, Ehmke and Horstmannshoff [16] propose the idea of personalized creation of multi-modal travel itineraries. They demonstrate the potential and limitations of "black box" mathematical optimization and discuss how to include more complex passenger preferences. Using solution sampling, they present a simple idea to identify the solution space's characteristics and allow travelers to restrict or improve their preferences interactively.
These works consider central "black-box" AI tools and provide explanations produced by another external or wrapper AI tool. We consider in this paper a decentralized

“transparent-box" multi-agent model for which the required explanation should not only focus on technical reasons but also it is necessary to convey the user’s preferences. It is necessary to decide what to reveal about agents’ preferences to increase user satisfaction, trust, and control while considering how those features led to the final decision.

Agent-based approaches have been employed in the literature to provide explanations to humans in intelligent transport systems. For example, Mualla. et al. [31] proposed a Human-Agent Explainability Architecture (HAExA) to formulate context-aware explanations for remote robots represented as agents [31, 29]. Considering that the human understandability of AI is subjective, they conducted empirical human-computer interaction studies employing Agent-based Simulation (ABS). The experiment scenario was about investigating the role of XAI in the communication between Unmanned Aerial Vehicles (UAVs) and humans in the context of package delivery in a smart city [30]. Their results showed that a balance between the simplicity of explanations and adequacy of the information contained in the explanations is needed. One interesting research direction, which we considered in this paper, is the importance of user-aware explanations [37]. Additionally, The results showed that ABS offers a test-bed environment to conduct human studies that facilitate the explanations reception by the human and visualize the behavior of the remote robots, represented as agents.

7 Conclusion

In this work, we explore the direction of explaining planning decisions in multi-agent resource allocation for ODT scenarios. Because there exists a huge variety of methods for resource allocation, the choice between these methods cannot be considered a straightforward decision. Moreover, these cannot be only seen as technical issues. The need for matching human satisfaction and controllable decisions requires these decisions to be transparent and self-explainable.

In this work, we aimed at conceptually designing a multi-agent model for an explainable recommendation of resource allocation mechanisms that match the user preferences and objectives in solving ODT scenarios, in particular contexts. We defined the system’s main components and how the explanations are generated and aggregated in multiple levels of abstraction, to scale for the user-defined level. We defined some general guidelines and assumptions for generating the explanations and communicating them to human users. We illustrated this proposal through some case study examples.

However, to bring this model into reality many open questions should be addressed, including: How much generic is this model to explain the variety of allocation mechanisms? How to define the right levels of detail that match the user needs, cognitive capacities, and competencies? How can we assess the quality of explanation and could that be automatic? We propose to deploy a proof-of-concept implementation of this model to address these challenges.

Bibliography

[1] Simulating Demand-responsive Transportation: A Review of Agent-based Approaches. 35. ISSN 0144-1647, 1464-5327. https://doi.org/10.1080/01441647.2015.1017749. URL http://www.tandfonline.com/doi/full/10.1080/01441647.2015.1017749.

[2] S. Aalami and L. Kattan. Fair dynamic resource allocation in transit-based evacuation planning. *Transportation Research Part C: Emerging Technologies*, 94:307–322, 2018. ISSN 0968-090X. https://doi.org/https://doi.org/10.1016/j.trc.2017.10.018. URL https://www.sciencedirect.com/science/article/pii/S0968090X17302917. ISTTT22.

[3] N. A. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang. Dynamic ride-sharing: A simulation study in metro atlanta. *Transportation Research Part B: Methodological*, 45(9):1450 – 1464, 2011. ISSN 0191-2615. https://doi.org/https://doi.org/10.1016/j.trb.2011.05.017. URL http://www.sciencedirect.com/science/article/pii/S0191261511000671. Select Papers from the 19th ISTTT.

[4] S. Anjomshoae, A. Najjar, D. Calvaresi, and K. Främling. Explainable agents and robots: Results from a systematic literature review. In *Proc. of 18th Int. Conf. on Autonomous Agents and MultiAgent Systems*, pages 1078–1088. Int. Foundation for Autonomous Agents and Multiagent Systems, 2019.

[5] E. Borowski and A. Stathopoulos. On-demand ridesourcing for urban emergency evacuation events: An exploration of message content, emotionality, and intersectionality. *International Journal of Disaster Risk Reduction*, 44:101406, Apr. 2020. ISSN 2212-4209. https://doi.org/10.1016/j.ijdrr.2019.101406. URL https://www.sciencedirect.com/science/article/pii/S221242091930799X.

[6] G. L. Bradley and B. A. Sparks. Dealing with service failures: The use of explanations. *Journal of Travel & Tourism Marketing*, 26(2):129–143, 2009.

[7] D. Calvaresi, Y. Mualla, A. Najjar, S. Galland, and M. Schumacher. Explainable multi-agent systems through blockchain technology. In *Proc. of 1st Int. Workshop on eXplainable TRansparent Autonomous Agents and Multi-Agent Systems (EXTRAAMAS 2019)*, 2019.

[8] M. Cashmore, A. Collins, B. Krarup, S. Krivic, D. Magazzeni, and D. Smith. Towards Explainable AI Planning as a Service. *arXiv:1908.05059 [cs]*, Aug. 2019. URL http://arxiv.org/abs/1908.05059. arXiv: 1908.05059.

[9] N. C. Codella, M. Hind, K. N. Ramamurthy, M. Campbell, A. Dhurandhar, K. R. Varshney, D. Wei, and A. Mojsilovic. Teaching meaningful explanations. *arXiv preprint arXiv:1805.11648*, 2018.

[10] J.-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, June 2007. ISSN 0254-5330, 1572-9338. https://doi.org/10.1007/s10479-007-0170-8. URL http://link.springer.com/10.1007/s10479-007-0170-8.

[11] A. Daoud, F. Balbo, P. Gianessi, and G. Picard. Ornina: A decentralized, auction-based multi-agent coordination in odt systems. *AI Communications*, pages 1–17, 2020.

[12] A. Daoud, F. Balbo, P. Gianessi, and G. Picard. A generic multi-agent model for resource allocation strategies in online on-demand transport with autonomous vehicles. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, page 3, 2021.

[13] A. Dhurandhar, V. Iyengar, R. Luss, and K. Shanmugam. TIP: typifying the interpretability of procedures. *CoRR*, abs/1706.02952, 2017. URL http://arxiv.org/abs/1706.02952.

[14] C. di Sciascio, P. Brusilovsky, and E. Veas. A study on user-controllable social exploratory search. In *23rd International conference on intelligent user interfaces*, pages 353–364, 2018.

[15] M. Egan and M. Jakob. Market mechanism design for profitable on-demand transport services. *Transportation Research Part B: Methodological*, 89:178–195, 2016.

[16] J. Ehmke and T. Horstmannshoff. Position Paper: Explainable Search of Multimodal Itineraries. In *Modellierung*, 2020.

[17] U. Ehsan, B. Harrison, L. Chan, and M. O. Riedl. Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 81–87, 2018.

[18] U. Ehsan, P. Tambwekar, L. Chan, B. Harrison, and M. O. Riedl. 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*, pages 263–274, 2019.

[19] B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

[20] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.

[21] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):93, 2018.

[22] D. Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2(2), 2017.

[23] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

[24] Y. Jin and B. Sendhoff. Trade-off between performance and robustness: An evolutionary multiobjective approach. In *International conference on evolutionary multi-criterion optimization*, pages 237–251. Springer, 2003.

[25] S. Kraus, A. Azaria, J. Fiosina, M. Greve, N. Hazon, L. Kolbe, T.-B. Lembcke, J. P. Muller, S. Schleibaum, and M. Vollrath. Ai for explaining decisions in multi-agent environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13534–13538, 2020.

[26] Z. C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018. https://doi.org/10.1145/3233231. URL https://doi.org/10.1145/3233231.

[27] Y. Liu, Z. Li, J. Liu, and H. Patel. A double standard model for allocating limited emergency medical service vehicle resources ensuring service reliability. *Transportation Research Part C: Emerging Technologies*, 69:120–133, 2016. ISSN 0968-090X. https://doi.org/https://doi.org/10.1016/j.trc.2016.05.023. URL https://www.sciencedirect.com/science/article/pii/S0968090X16300602.

[28] J. Ludwig, A. Kalton, and R. Stottler. Explaining complex scheduling decisions. In *IUI Workshops*, 2018.

[29] Y. Mualla. *Explaining the Behavior of Remote Robots to Humans: An Agent-based Approach*. PhD thesis, Belfort, France, 2020. 2020UBFCA023.

[30] Y. Mualla, A. Najjar, T. Kampik, I. Tchappi, S. Galland, and C. Nicolle. Towards explainability for a civilian uav fleet management using an agent-based approach. *1st Workshop on Explainable AI in Automated Driving: A User-Centered Interaction Approach, Utrecht, Netherland. arXiv preprint arXiv:1909.10090*, 2019.

[31] Y. Mualla., I. H. Tchappi., A. Najjar., T. Kampik., S. Galland., and C. Nicolle. Human-agent explainability: An experimental case study on the filtering of explanations. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: HAMT,*, pages 378–385. INSTICC, SciTePress, 2020. ISBN 978-989-758-395-7. https://doi.org/10.5220/0009382903780385.

[32] A. Preece. Asking 'Why'in AI: Explainability of intelligent systems–perspectives and challenges. *Intelligent Systems in Accounting, Finance and Management*, 25 (2):63–72, 2018.

[33] K. N. Ramamurthy, B. Vinzamuri, Y. Zhang, and A. Dhurandhar. Model agnostic multilevel explanations. *arXiv preprint arXiv:2003.06005*, 2020.

[34] A. Rosenfeld and A. Richardson. Explainability in human–agent systems. *Autonomous Agents and Multi-Agent Systems*, pages 1–33, 2019.

[35] M. Schilde, K. Doerner, and R. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research*, 38(12):1719–1730, 2011. ISSN 0305-0548. https://doi.org/https://doi.org/10.1016/j.cor.2011.02.006. URL https://www.sciencedirect.com/science/article/pii/S0305054811000475.

[36] J. L. Schofer, T. C. R. Program, and N. R. C. U. S. . T. R. Board. *Resource Requirements for Demand-responsive Transportation Services*. Transportation Research Board, 2003. ISBN 978-0-309-08778-0. Google-Books-ID: RG9wnNBKCy4C.

[37] R. Singh, P. Dourish, P. Howe, T. Miller, L. Sonenberg, E. Velloso, and F. Vetere. Directive explanations for actionable explainability in machine learning applications. *arXiv preprint arXiv:2102.02671*, 2021.

[38] N. Tintarev and J. Masthoff. Designing and evaluating explanations for recommender systems. In *Recommender systems handbook*, pages 479–510. Springer, 2011.

[39] C.-H. Tsai. *Controllability and Explainability IN A HYBRID SOCIAL RECOMMENDER SYSTEM*. PhD thesis, 2019.

[40] D. Yankov. Discrete Event System Modeling Of Demand Responsive Transportation Systems Operating In Real Time. *Graduate Theses and Dissertations*, Mar. 2008. URL https://scholarcommons.usf.edu/etd/575.

[41] H. Yu, Y. Liu, X. Wei, C. Zheng, T. Chen, Q. Yang, and X. Peng. Fair and Explainable Dynamic Engagement of Crowd Workers. page 3.