Received 20 December 2023, accepted 13 February 2024, date of publication 22 February 2024, date of current version 29 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3369041

RESEARCH ARTICLE

Dynamic Matching Optimization in Ridesharing System Based on Reinforcement Learning

HIBA ABDELMOUMÈNE[1,2], CHEMESSE ENNEHAR BENCHERIET[1,3], HABIBA BELLEILI[2], ISLEM TOUATI[1], AND CHAYMA ZEMOULI[1]

[1]Computer Science Department, Université 8 Mai 1945 Guelma, Guelma 24000, Algeria
[2]LabGED Laboratory, Badji Mokhtar University, Annaba 23000, Algeria
[3]LAIG Laboratory, Université 8 Mai 1945 Guelma, Guelma 24000, Algeria

Corresponding author: Hiba Abdelmoumène (abdelmoumene.hiba@univ-guelma.dz)

ABSTRACT Modern urban transportation, has concurrently posed environmental challenges such as traffic congestion and increased greenhouse gas emissions. In response to these issues, ridesharing systems have emerged as a viable solution. By fostering ridesharing among individuals with similar travel routes, ridesharing, effectively, optimizes vehicle utilization, offering a sustainable and practical alternative to address contemporary transportation challenges. In this work, we delve into intricacies of dynamic ridesharing systems. Focusing on the dynamic matching problem within ridesharing, we propose a solution leveraging reinforcement learning. Our contribution involves the distinct modeling of two scenarios: one-to-one and one-to-many ridesharing. In the one-to-one scenario, spatiotemporal constraints are considered with the objective of minimizing passengers' waiting times. In the more complex one-to-many scenario, additional constraints are introduced focusing on both minimizing passengers' waiting times and drivers' detour times. The proposed modeling is time-focused assuming that time is a cutting parameter in the decision-making. The results obtained through our experiments demonstrate the system's effectiveness, robustness and adaptability to diverse constraints.

INDEX TERMS Dynamic ridesharing, dynamic matching, reinforcement learning, spatiotemporal constraints, detour.

I. INTRODUCTION

The World's population is growing rapidly, especially in urban areas and citizens have an increasing need to move around and be mobile. However, this need for mobility is not without consequences. Indeed, today's cities are facing challenges in terms of congestion, lack of space, air pollution, climate change, etc.

To address these problems, various technological solutions have been proposed, including autonomous vehicles [1], vehicular networks [1], and the internet of things, etc. Furthermore, shared mobility [2] has gained popularity in recent years and has given rise to new transportation services such as ridesharing [2], [3], [4], [5], [6], carpooling [6], Dial a Ride Problem (DARP) [6], …

The ridesharing system [3], [4] is a mode of transportation that allows drivers and passengers with similar travel needs to make joint trips. This obviously has the potential to reduce traffic, fuel consumption and pollution. Therefore, this service is an important aspect of modern society (include smart cities). The emergence of technological advances in recent years (Global Position System GPS, powerful mobile applications, etc.), have called for dynamic ridesharing which must be able to respond to real-time requests and provide more sophisticated automatic matches than a simple radial search around origins and destinations. Ridesharing systems in modern times have adopted a dynamic allocation process, which considers times and routes to match passengers with drivers. The process of establishing a relationship between multiple users based on their preferences is called the ride-matching problem. Companies like Uber[1] and Lyft[2]

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chatterjee.

[1]https://www.uber.com/
[2]https://www.lyft.com/

provide current ridesharing systems that utilize a simple matching algorithm allocating users solely based on their travel routes. However, these commercial systems have limitations, as they employ a user-matching scheme that relies on two primary factors: (1) the passenger's destination: the driver's destination is dependent on the destination of the passenger; (2) the service's cost is calculated by the provider based on distances and times, without considering the cost-sharing of both the passenger and the driver. The dynamic ridesharing system is a solution that provides the flexibility and comfort of a private vehicle while offering rates and prices comparable to public transportation, distinguishing it from other conventional means of transport.

In this work, we consider the problem of dynamically matching drivers and passengers. The latter presents one of the necessary components for a successful dynamic rideshare system which needs deployment of effective and efficient optimization technology. This optimization problem entails efficiently managing a myriad of requests from drivers offering rides and passengers seeking transportation, ensuring optimal matches while respecting a set of constraints [7], [8], [9], [10].

The primary objective of our research is to develop a robust system capable of optimizing dynamic ridesharing solutions. This objective is two-fold: first to enhance the efficiency of traditional one-to-one dynamic ridesharing, where a single driver matches with a single passenger, and second, to delve into the intricacies of one-to-many dynamic ridesharing, specifically addressing detour considerations. The problem that we address encompass temporal considerations, spatial proximity, and the additional challenge posed by detours in one-to-many ridesharing scenarios. Striking a balance between the reduction of waiting times for passengers and the minimization of detour times for drivers is at the core of our optimization endeavor.

In approaching this optimization challenge, we explore the integration of Reinforcement Learning (RL) [11], [12], a methodology traditionally employed to solve sequential decision-making problems. Indeed, a key feature of the dynamic ridesharing problem is its spatiotemporal nature. The eligibility of a driver to match a passenger's request depends in part on his spatial proximity to the request. Moreover, responses to ride requests typically take a different amount of time to complete, and they change the spatial states of drivers, affecting the distribution of supply for a future match. Therefore, operational decisions in dynamic ridesharing are sequential in nature and have a strong spatiotemporal dependence, which offers excellent applications of reinforcement learning. Many works have considered reinforcement learning in response to this type of problem [12]. The major contribution of each work concerns the modeling of the different parameters of the problem by the different parameters of the theoretical foundation of RL, namely, Markov Decision Processes (MDPs) [13]. The aim of this work is to propose a good modeling of time and space for dynamic matching problem in order to minimize passengers' waiting times and drivers' detour times while respecting the time windows of the users, their locations and vehicles' capacity. Unlike other works [14], [15], [16], the modeling we propose is time-focused, having assumed that time is sufficient information for decision-making. Indeed, time is a cutting parameter in the decision: if time does not allow to take the passenger, it is useless to see the location.

The remainder of the paper is organized as follows. We classify and summarize related articles in section II. The problem description and base formulation are given in section III. Sections IV and V discuss our main contributions in modeling one-to-one and one-to-many ridesharing problems. The performance of our proposed approaches is evaluated in section VI. In Section VII, we provide a discussion of our work and the obtained results, positioning our study in relation to other works in the literature. Section VIII concludes and summarizes our work with future research directions.

II. LITERATURE REVIEW

A real-time ridesharing system aims to bring together travelers within a very short timeframe. Therefore, it may need to be reoptimized at regular intervals as new travelers enter or exit the system. Consequently, drivers and passengers already en route must be informed of any change in plans each time the system is reoptimized, as their original routes may be altered. This automated process requires efficient models and algorithms to match drivers and passengers in very short computation times. Several previous works addressed the online matching from exact methods [17], [18], [19] to heuristic and metaheuristic methods [20], [21] to reinforcement learning (RL) [12].

Wang et al. [14] model the ridesharing problem as a Markov Decision Process and proposed learning solutions based on deep Q-networks with action search to optimize the dispatching policy for drivers on ridesharing platforms. The authors introduced a new transfer learning method called Correlated Feature Progressive Transfer, along with two existing methods, to enable knowledge transfer in both spatial and temporal spaces, increasing learning adaptability and efficiency. Ke et al. [15] established a framework that combines deep reinforcement learning and multi-agent combinatorial optimization, in which the timing of each passenger request entering for matching is dynamically determined using multi-agent reinforcement learning techniques, while combinatorial optimization ensures bidirectional perfect matching between waiting passengers and inactive drivers. In [22], the authors addressed issues related to matching and repositioning, two key operations in ridesharing platforms. They suggest using a centralized value function as a foundation for learning and optimization to capture the interactions between these two tasks. In this perspective, they proposed an innovative approach based on a set of values, enabling fast online learning and large-scale offline training. Li et al. [23] addressed the ride-matching problem using multi-agent reinforcement learning, which follows the distributed nature of the peer-to-peer ridesharing problem and possesses the ability to capture the stochastic demand-supply dynamics.

The goal of [24] was to maximize transport efficiency so that fewer cars are needed to meet a given travel demand. To achieve this, they developed a spatiotemporal neural network (deep ST-NN: SpatioTemporal Neural Network) to predict travel time from GPS data. The constructed network learns travel time and distance from GPS coordinates of an origin and destination and the time of day. Subsequently, they developed a simulator for reinforcement learning using the outputs of the ST-NN. The objective is to build a policy that guides the driver on when to accept a ridesharing request to maximize long-term transport efficiency and reduce traffic congestion.

In reviewing the works addressing the problem with reinforcement learning, it is notable that a predominant approach involves treating the driver as the primary agent. The state representations commonly encompass details like the driver's location, departure and arrival times, and pertinent constraints, such as available seat count. Regarding environmental observations, the majority of studies factor in the coordinates of both the driver and passenger locations, along with temporal aspects, leading to an expanded state space. Rewards in this context typically function as optimization criteria, serving to guide the agent in making judicious decisions while steering clear of suboptimal ones.

III. PROBLEM DESCRIPTION

In this section, we present the relevant background knowledge and we formulate the online matching problem that we consider in this work.

Dynamic ridesharing is an automated process in which a service provider connects drivers and passengers with similar routes and schedules to share a ride on short notice in a personal vehicle. Ridesharing systems are inherently dynamic given the need for quick and responsive matching. The challenge of these systems lies in connecting individuals who must adhere to different constraints (spatiotemporal, seat vacancy, users' preferences, etc.). These constraints must be specified in advance by both drivers and passengers before the desired trip is defined and carried out

We consider, a dynamic matching problem in ridesharing system where the goal is to bring together drivers and passengers such that passengers' waiting time and drivers' detour time are minimized. Each driver and passenger carry a trip where he specifies his origin, his destination and time window that involves latest departure time and latest arrival time. Furthermore, we consider, in this work, two different online matching problems: one-to-one (one driver matched to one passenger) and one-to-many (one driver matched to many passengers).

Notation used to set our problem is summarized in table 1.

For the **one-to-one matching problem**, an *offer* of driver d, V_d, is represented by a quadruple $V_d = \left(o_d, ds_d, dp_d, ar_d\right)$, specifying, origin and destination points and latest departure and arrival times, respectively.

TABLE 1. Notation table.

Symbol	Description
D	A set of drivers indexed by d
P	A set of passengers indexed by p
V_d	A vehicle offer for a trip of driver d
R_p	A request for a trip of passenger p
o_d, ds_d	The origin and destination of driver d
o_p, ds_p	The origin and destination of passenger p
dp_d, ar_d	Latest departure time and latest arrival time of driver d
dp_p, ar_p	Latest departure time and latest arrival time of passenger p
sg_d	The number of available seats in the vehicle of driver d
$\max\_dt_d$	The maximum detour time of driver d
$\max\_wt_p$	The maximum waiting time of passenger p

A passenger's, p, *request*, R_p, is represented by a quintuple $R_p = \left(o_p, ds_p, dp_p, ar_p, \max\_wt_p\right)$ specifying origin and destination points, latest departure and arrival times and maximum waiting time, respectively.

For this instance of problem, we consider an inclusive ridesharing model [3]. This means that both the origin and destination of a passenger are located on the route of an original driver's itinerary.

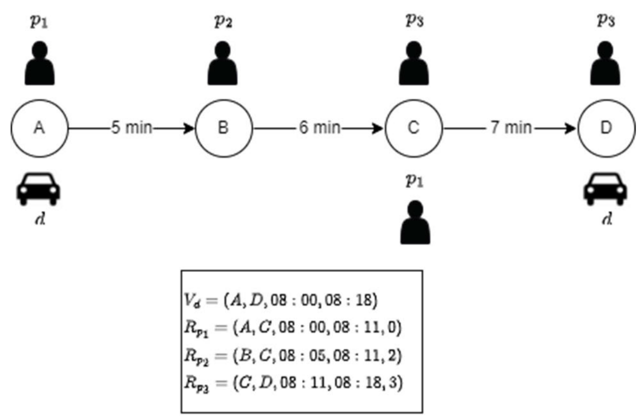

FIGURE 1. A scenario of one-to-one inclusive ridesharing model.

Fig. 1 presents an example for one-to-one inclusive ridesharing model. We consider a driver d that offer to share his vehicle on a route from point A to point D, a request from passenger p_1 has been matched with the offer of d. This passenger will share the journey from A to C with d (inclusive route). At point B, the system receives a new request from passenger p_2. This request cannot be fulfilled as the vehicle is already occupied. At point C, the driver drops off p_1, and the system receives a new request from p_3, whose route is included in d's route. This request is matched with d, who picks up p_3 at point C and drops him off at point D (his destination).

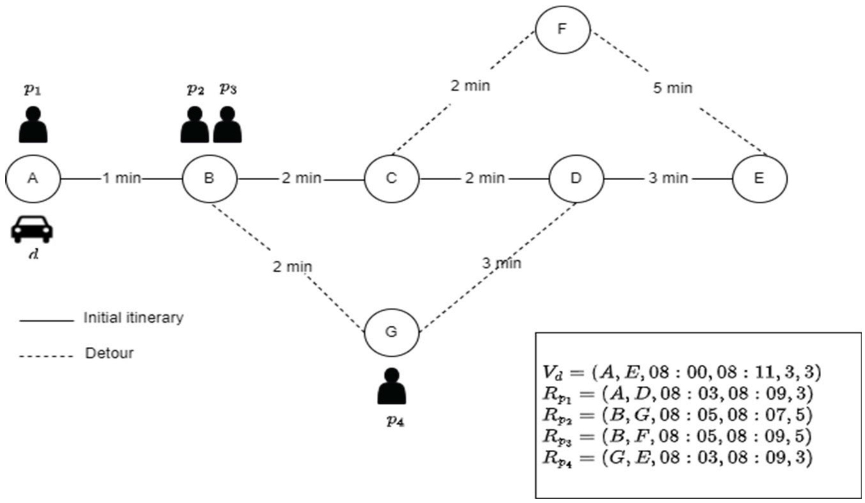

FIGURE 2. A scenario of one-to-many ridesharing model with detour.

For the **one-to-many matching problem**, an *offer* of driver d, V_d, is represented by a sextuple $V_d = (o_d, ds_d, dp_d, ar_d, sg_d, \max\_dt_d)$, specifying, origin and destination points, latest departure and arrival times, the number of available seats and maximum detour time, respectively.

A passenger's, p, *request*, R_p, is represented by a quintuple $R_p = (o_p, ds_p, dp_p, ar_p, \max\_wt_p)$ specifying origin and destination points, latest departure and arrival times and maximum waiting time, respectively.

For this instance of problem, we consider an exclusive ridesharing model [3], where a driver can take a detour (additional distance) in order to pick up or drop off passengers.

Fig. 2 describes an example for one-to-many ridesharing model with detour.

We propose addressing the dynamic matching problems considered through reinforcement learning (RL). The dynamic nature, spatiotemporal characteristics, decision-making aspects (whether to match or not), and environmental constraints—all these factors are inherent in reinforcement learning, especially in its sequential decision-making aspect.

In the next sections, we will present in detail the problem modeling approach we propose for the two instances (one-to-one and one-to-many).

IV. ONE-TO-ONE RIDESHARING MATCHING PROBLEM

Our challenge, in this work, is to model the aforementioned parameters (offers and requests) using the various components of reinforcement learning. RL, grounded in its theoretical foundation, relies on Markov Decision Processes (MDP) [13]. An MDP consists of a set of states, a set of actions, a transition function, and a reward function. Modeling the constrained dynamic matching decision problem using an MDP entails constructing each component of the MDP to integrate the necessary information for decision-making.

Our modeling centers around positioning the driver as the focal point, representing the agent. The driver, acting as the learning entity, makes decisions regarding whether to accept a passenger, considering specific locations and times. The ultimate aim is to optimize the passenger's waiting time within our system.

In the following, we describe the model construction of our constrained ridesharing decision problem.

A. PROBLEM MODELING

1) STATE

The state s_t is captured by the geographical coordinates of the driver, the passenger, the next destination point, and the time. s_t is time stamped and is defined by $s_t = (C_d, C_p, C_{nd}, t)$, where:

- C_d is the pair of GPS coordinates (latitude, longitude) of the driver's origin and destination points,
- C_p is the pair of GPS coordinates (latitude, longitude) of the passenger's departure and destination points,
- C_{nd} is the pair of GPS coordinates (latitude, longitude) of the next destination point in the driver's itinerary.

2) OBSERVATION

At every time step, the driver receives an observation from the environment, representing the current state of the system. This observation provides information about the dynamic conditions, including the driver's location, passenger requests, and other relevant factors. The driver then draws an observation that is correlated with the state of the environment. This observed information serves as the basis for the driver's decision-making process, allowing them to assess the surrounding conditions and make informed choices regarding whether to accept passengers or not.

We define the observation within our model as a tuple $o_t = (t_{nd}, Wt_p, max\_wt_p)$, where:

- t_{nd} is the time to reach the next destination,
- Wt_p is the current waiting time of the passenger at time t,

- max_wt_p is the maximum waiting time of the passenger.
Our choice of these three parameters is justified by:

(1) Our overall objective, which is the minimization of passenger waiting time. Therefore, time represents necessary information for decision-making and dynamic accuracy, ensuring stability and a quick response from our system.

(2) After reviewing existing works that utilized RL to solve similar problems, we observed that all these works transmit the locations of drivers and passengers to the agent to make a decision, thereby increasing the space. However, in our work, we assume that time is sufficient information for decision-making and closely reflects reality. Indeed, time is a decisive factor in the decision: if the time does not allow picking up the passenger, knowing the location becomes irrelevant.

3) ACTION
Two actions are available for the driver, reflecting his decision:

$$\begin{cases} a = 0, & \textit{decline the request} \\ a = 1, & \textit{accept the request} \end{cases}$$

4) REWARD
The reward signal in our problem must provide information that guides the agent's behavior to achieve the main objective of the system, which is the minimization of passengers' waiting time. Therefore, we propose that this signal, r, corresponds to the waiting time saved by passengers on a trip and is a function of the current state (time to the next destination, current waiting time of the passenger, and the maximum waiting time of the passenger) and is defined by:

$$r = \max\_wt_p - \left(Wt_p + t_{nd}\right)$$

Our proposed reward signal signifies that a shorter waiting time corresponds to a higher reward. The agent is required to act in a way that maximizes this reward, aligning with our optimization objective.

B. AGENT DECISION
Upon receiving a new request, the driver processes it by considering waiting time, current time, and the time to reach the next destination. Subsequently, the agent makes a decision on whether to accept the request.

If accepted, the driver either picks up the passenger or provides notification. Following the reward calculation, if the driver is already en route or has an existing request to fulfill, they proceed to the next destination. Alternatively, if no active tasks are present, the driver checks whether it's time to depart; if not, they wait.

Upon completion of this decision-making process, the environment updates and returns the new state based on the revised information.

V. ONE-TO-MANY RIDESHARING MATCHING PROBLEM
The objective in this second contribution is to broaden the scope of the problem addressed above by considering multiple passengers in a single assignment. By integrating this new dimension, we aim to account for partial trips and introduce the detour constraint. Thus, our goal is to optimize two objectives: minimize passengers' waiting time and reduce drivers' detour time. This approach allows us to meet the needs of both parties involved in the ridesharing system.

In the following sections, we present the problem modeling as an MDP and the agent decision for the one-to-many ridesharing problem with detour. Again, our modeling revolves around placing the driver at the core, serving as the agent. In this learning framework, the driver, as the decision-maker evaluates whether to accept a request, taking into account, spatiotemporal constraints, vacant seats, passengers' waiting time and the possibility of making a detour.

A. PROBLEM MODELING
1) STATE
State s_t is time stamped and is defined by a tuple: $s_t = \left(C_d, C_p, C_{nd}, sg, C_{ds_P}, ar_P, C_{dt} t\right)$, where:

- C_d is the pair of GPS coordinates (latitude, longitude) of the driver's origin and destination points,
- C_p is the pair of GPS coordinates (latitude, longitude) of the passenger's departure and destination points,
- C_{nd} is the pair of GPS coordinates (latitude, longitude) of the next destination point
- sg is the number of available seats at time t,
- C_{ds_P} is the set of GPS coordinates (latitude, longitude) of arrival points of the set P of passengers on board,
- ar_P is the set of latest arrival times of the set P of passengers on board,
- $\boldsymbol{C_{dt}}$ is the set of GPS coordinates of the points on the detour itinerary.

2) OBSERVATION
We define the observation as $o_t = \left(t_{nd}, Wt_p, max\_wt_p, sg_t, max\_\boldsymbol{dt}, \boldsymbol{\Delta}_{dt}\right)$, where:

- t_{nd} is the time to reach the next destination,
- Wt_p is the current waiting time of passenger p at time t,
- max_wt_p is the maximum waiting time of passenger p,
- sg_t is the number of available seats at time t,
- $max\_\boldsymbol{dt}$ is the maximum detour time of the driver,
- $\boldsymbol{\Delta}_{dt}$ is the difference between the durations of the initial itinerary with and without detour.

Our choice may be motivated by the fact that our goal is to minimize passenger waiting time and driver detour time. Thus, providing the agent with a comprehensive set of information related to making the right decision regarding detour and passenger waiting is necessary. We assume that temporal parameters are decisive factors for the agent's decision before even considering locations.

3) ACTION
Two actions are available for the driver, reflecting his decision:

$$\begin{cases} a = 0, & \textit{decline the request} \\ a = 1, & \textit{accept the request} \end{cases}$$

4) REWARD

To achieve our primary system goals of reducing waiting time and detour time, it is essential that the reward signal in our problem contains information that guides the agent's action.

Therefore, we propose defining the reward signal (r) based on the reduction in passenger waiting time and driver detour time during a specific trip. This reward signal will measure the gain in terms of time from both the passenger and driver perspectives. We define the reward as follows:

$$r = \max\_wt_p - \left(Wt_p + t_{nd}\right) + \max\_dt - \boldsymbol{\Delta}_{dt}$$

where, max_wt_p is the maximum waiting time of the passenger, Wt_p, t_{nd} are the current waiting time of the passenger and the next destination time respectively and max_dt, Δ_{dt} are the maximum detour time and the difference between the durations of the initial itinerary with and without detour of the driver, respectively.

Example: According to the scenario presented in Fig.2, we assume that the driver is at point B. At the same point, the driver has received two requests R_{p_2} and R_{p_3}, knowing that there is only one available seat. The respective destinations of p_2 and p_3 are points G and F. Both destinations require a detour from the driver.

The agent's (driver) decision at this moment will involve accepting either passenger p_2 or passenger p_3. The observations received by the agent at this moment are:

- For p_2: $o_t = (0, 1, 5, 1, 3, 1)$
- For p_3: $o_t = (0, 1, 5, 1, 3, 2)$

Let's compute the reward:

- If the driver decides to accept p_2, the reward will be: $r = 5 - (1 + 0) + 3 - 1 = 6$.
- Il the driver decides to accept p_3, the reward will be: $r = 5 - (1 + 0) + 3 - 2 = 5$.

In this case, the driver will choose to take passenger p_2, as it saves more time than taking p_3.

B. AGENT DECISION

In the decision-making process, the driver may receive two types of requests: those that align with his initial itinerary and those that require a detour.

Requests are evaluated based on various criteria, such as the passenger's current waiting time, their maximum waiting time, the remaining time before their next destination, and the number of available seats in the vehicle. For detour requests, additional factors come into play, including the driver's latest arrival time, his maximum detour time, the detour destination, and the latest arrival time of passengers already on board. Based on these criteria, the agent decides whether to accept or reject the request.

Once an action is selected, the reward is calculated based on the agent's current state. If the driver has already started the journey or all seats are occupied, he proceeds to his next destination. Otherwise, he assesses whether it's time to depart or if he still needs to wait to maximize ridesharing efficiency. After making a decision and completing the task, the environment returns a new state reflecting the updated information.

This iterative decision-making process continues until the driver reaches his final destination. At each iteration, the agent evaluates the actions taken and learns from the outcomes. If a decision proves unfavorable, the agent considers this experience when faced with similar situations in the future, aiming to avoid repeating the same mistakes. Through this iterative learning, the agent progressively improves his choices and makes more advantageous decisions over time.

The pseudo-code of agent decision algorithm is given below.

Algorithm Agent Decision

Input: V_d, R_{p_i}, t
Output: $\left(V_d, R_{p_i}\right)$
If $sg_d \neq 0$ **then**
  If $o_{p_i} \subset C_{dt}$ *or* $ds_{p_i} \subset C_{dt}$ **then**
    If $\boldsymbol{\Delta}_{dt} \leq max\_\boldsymbol{dt}$ **then**
      If $C_{ds_P} \subset C_{dt}$ **then**
      While $(ar_i \in ar_P)$ **do**
        If $t + \boldsymbol{\Delta}_{dt} + t_{nd} > ar_{p_i}$ **then**
          Go to next destination
      Pick-up the passenger
    Else Go to next destination
  Else Go to next destination
 Else Pick-up the passenger
Else Go to next destination
Return $\left(\boldsymbol{V_d}, \boldsymbol{R_{p_i}}\right)$

VI. EXPERIMENTAL RESULTS

To implement our system, two distinct stages are involved. The first is a learning phase aimed at constructing the above-presented model. In this phase, the agent learns behavior that effectively deals with the various constraints, such as minimizing waiting times for passengers and reducing detour times for drivers, and optimizes the considered objectives.

The second involves deploying this model in a simulator. In this section, we will discuss these two phases within the two problem instances considered.

A. LEARNING

An important aspect of our work is to construct a model, inject it into the agent, and enable it to learn a decision-making behavior that maximizes efficiency within the constraints and objectives of the problem.

To build this model, we need a dataset containing the necessary information corresponding to our problem description for the directions of each trip.

To achieve this, we will utilize the Uber Pickup[3] database in the city of New York. This repository contains data on over 4.5 million Uber pickups in New York City from April to September 2014.

[3] https://www.kaggle.com/datasets/fivethirtyeight/uber-pickups-in-new-york-city

We use a specific file from this directory called *"federal 02216"*, which contains raw data on pickups from a non-uber FHV company. It includes information such as *Date* (the date of the request), *Time* (the time of the request), *PU_Address* (the pick-up address), *DO_Address* (the drop-off address), *Routing Details*, and *status* (request status).

To conduct our experiments, we randomly selected two data samples from Federal_02216. The first one, consists of 93 trips (for the one-to-one problem), and the second one, consists of 48 trips (for the one-to-many- problem). Then, we used the Google Directions API to obtain direction details.

In order to train our agent, we have chosen to use Proximal Policy Optimization (PPO) [25]. PPO is considered one of the simplest and most popular algorithms for reinforcement learning. This technique involves interacting with the environment to collect data and then optimizing an objective function using the ascent gradient.

To evaluate the learning results, we considered a complete driver's journey as an episode to calculate the reward curve. The total duration of the learning process amounts to 800000 episodes with a discount factor ($\gamma = 0,9$), during which the agent aims to maximize cumulative rewards.

Before proceeding to real-time simulation, to evaluate the agent's reactions, we conducted preliminary tests on several scenarios. The objective was to determine whether the agent consistently makes the correct choice. To do this, we subjected the model to various observations to assess its decisions in diverse contexts.

The different observations provided to the agent encompassed a set of crucial factors influencing its decisions. This included departure and arrival times of the trips, origin and destination points, the number of available seats, passenger waiting times, the driver's maximum detour time, as well as itineraries whether they required a detour or not. Analyzing these different variables allowed us to evaluate the agent's ability to make relevant decisions in a variety of situations.

B. SIMULATOR

In this section, we perform a series of numerical experiments to validate our proposed models using metrics crucial to the ridesharing problem. We design a simulator that helps us understand how our models perform in different scenarios, providing insights into their effectiveness and adaptability in an online matching system.

1) ONE-TO-ONE MODEL

To initialize the environment, we have launched the simulation from 08:00 am to 9:00 pm. The initial locations of offers and requests are randomly selected from the dataset.

We recall that, in this model, we consider inclusive itineraries, where the trip of the passenger is included in the trip of the driver.

We set the following parameters (Table 2). Δt is the time interval for receiving new offers and requests by the system.

TABLE 2. Initialization parameters.

Δt	Offers	Requests	max_wt_p
4 min	[1,5]	[1,5]	[15,30]

Thus, every 4 minutes, the system receives a random number between 1 and 5 of offers and requests.

TABLE 3. Simulation results from 08:00 am to 09:00 pm.

Simulation results	
Offers	4254
Requests	3772
Accepted requests	2514
Total reward	8145
Total waiting time	5115

In table 3, we provide a comprehensive overview of key performance metrics for the simulation period from 08:00 am to 09:00 pm. The table includes the total number of offers and requests received during this timeframe, shedding light on the demand and supply dynamics within the system. Additionally, we outline the total number of accepted requests (successfully completed trips), the total reward, representing the cumulative profitable time, and the total waiting time, offering insights into the efficiency and user experience of the system throughout the simulation duration.

The presented table highlights a notable outcome wherein the total system reward for the simulated time interval reaches 8145 minutes. This reward signifies the accumulated time gain, often referred to as profitable time. It's noteworthy that this gain substantially exceeds the total waiting time, which amounts to 5115 minutes. This discrepancy underscores the system's efficiency in optimizing the overall experience, ensuring that the accumulated profitable time significantly outweighs the waiting periods.

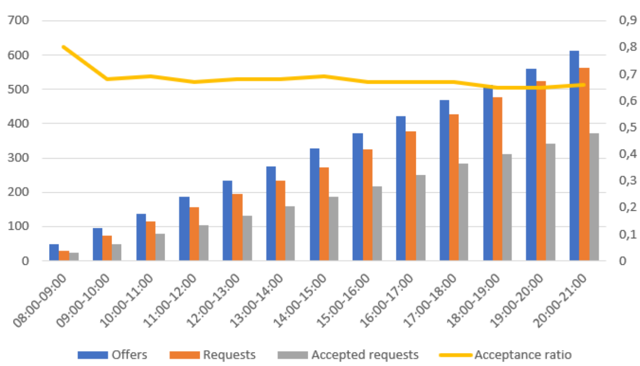

FIGURE 3. Simulation results per hour.

Fig. 3 presents details of simulation results per hour. Furthermore, we assess the acceptance rate by calculating the ratio of accepted requests to the total number of requests per hour. Notably, the observed acceptance rate consistently falls within the range of 0.65 to 0.80. This indicates that, during the simulation, a substantial proportion of requests were successfully accommodated, aligning with an acceptance rate that remained consistently high, demonstrating the effectiveness of the system in efficiently handling incoming requests.

To provide a more comprehensive understanding of the effectiveness of the proposed method, we evaluated our system across four environments corresponding to different parameterizations. All scenarios start at 08:00 am and end at 11:00 am. The initial locations of offers and requests are randomly chosen from the dataset. We present in table 4 the parameters of the various tested scenarios. The results of the execution of these scenarios are presented in table 5.

TABLE 4. Parameters of the tested scenarios.

	Δt	Offers	Requests	max_wt_p
Scenario 1	4 min	5	[1,5]	[15,30]
Scenario 2	1 min	[1,5]	[1,5]	[15,30]
Scenario 3	4 min	[1,5]	[1,5]	15 min
Scenario 4	4 min	[1,5]	[1,5]	30 min

TABLE 5. Simulation results for different scenarios.

	Offers	Requests	Accepted requests	Total reward	Total waiting time
Scenario 1	680	375	307	4305	785
Scenario 2	1594	1499	465	6990	4730
Scenario 3	379	378	217	1121	1024
Scenario 4	412	323	231	2417	1423

In all these scenarios, our system consistently demonstrates that the total reward, representing profitable time, outweighs the waiting time. In scenario 1, the total waiting time is 785 minutes, while the total reward is 4305 minutes, accounting for 85% of the time compared to 15% for waiting time. In scenario 2, 60% of the time is considered profitable, and 40% is waiting time. Scenarios 3 and 4 yield total rewards of 52% and 63%, respectively, with waiting times at 48% and 37% for each scenario. This pattern underscores the effectiveness of our system in minimizing waiting time which leads to maximizing profitable time.

2) ONE-TO-MANY MODEL

The second problem considered, in our work, concerns matching one driver to multiple passengers taking into accounts seats availability and the possibility of making a detour in order to pick-up or drop-off passengers.

In order to evaluate this model, we have launched simulation from 08:00 am to 09:00 pm with $\Delta t = 5$ *minutes*, the number of offers is between 4 and 10, and the number of requests is between 10 and 18. The initial locations of offers and requests are randomly selected from the dataset. Table 6 presents simulation parameters.

TABLE 6. Initialization parameters.

Δt	Offers	Requests	max_wt_p	max_dt_d
5 min	[4,10]	[10,18]	[10,30]	[5,15]

TABLE 7. Simulation results from 08:00 am to 09:00 pm.

Simulation results	
Offers	1000
Requests	2960
Accepted requests	1167
Accepted requests with detour	202
Accepted requests without detour	965
Total reward	20000
Total waiting and detour times	16803

The data obtained from the simulation, as presented in Table 7 highlight the effectiveness of our approach. The total reward achieved, interpreting the gain in time resulting from the drivers' informed decisions regarding detours and passenger waiting times, amounts to 20000 minutes for a simulation from 8:00 am to 9:00 pm. This remarkable figure clearly demonstrates the significant benefits of our application.

By optimizing detour decisions, our model has enabled drivers to minimize the additional time spent on the road while still meeting the needs of passengers. Furthermore, by optimizing passenger waiting times, our application has greatly enhanced their experience by reducing the overall waiting time. These results underscore the effectiveness of our approach in making optimal decisions for drivers and improving the passenger experience.

To more effectively evaluate the proposed method, experiments were conducted in different environments and configurations, primarily focusing on the number of requests and offers to assess scalability. Additionally, we varied the maximum detour time to observe the decision-making behavior of the agent. Three distinct scenarios were created, each covering a time range from 7:00 am to 12:00 pm. The initial locations of ridesharing offers and requests were randomly selected from our dataset. For each driver, we set the number of available seats to 3. Details of the three scenarios parametrization are given in table 8

The results of the execution of these scenarios are presented in table 9 and illustrated in Fig. 4.

The experiments we conducted aim to assess the adaptability of our system in situations where drivers may face

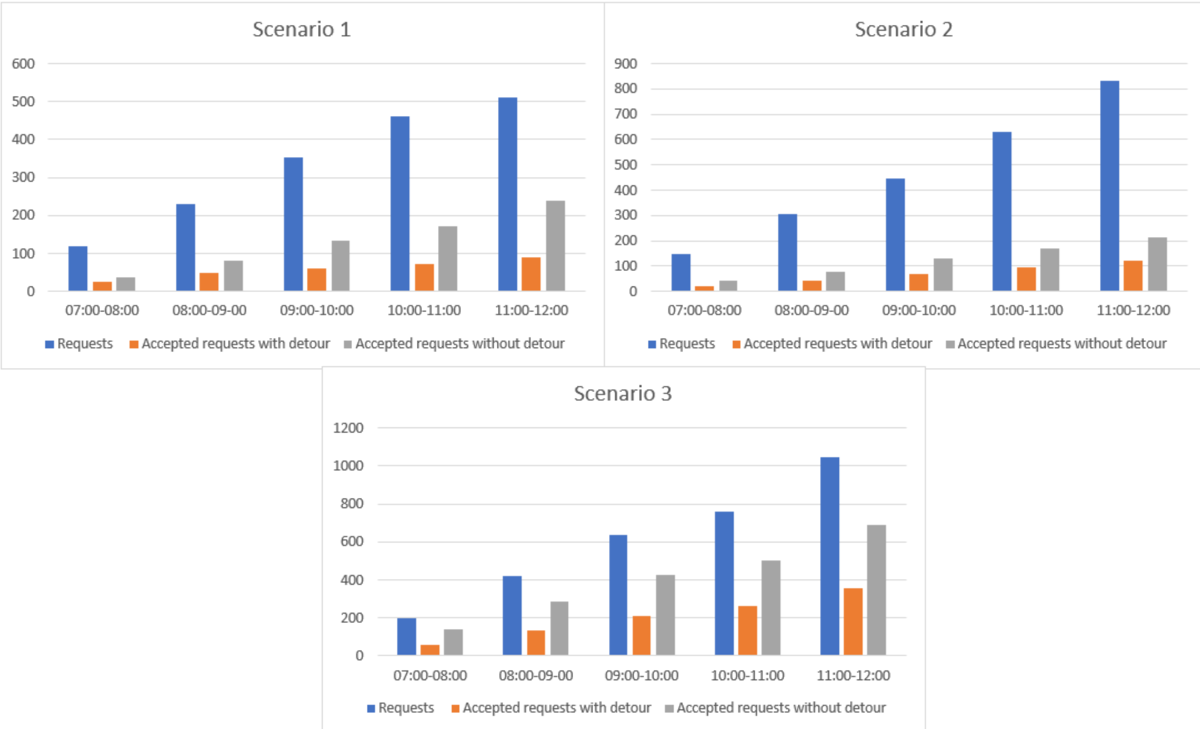

FIGURE 4. Simulation results for the three scenarios.

TABLE 8. Parameters of the tested scenarios.

	Δt	Offers	Requests	$\max\_wt_p$	$\max\_dt_d$	sg_d
Scenario 1	5 min	[3,10]	[9,15]	[10,25]	[5,10]	3
Scenario 2	5 min	[3,10]	[9,15]	[10,25]	20	3
Scenario 3	5 min	[15,30]	[20,50]	[10,25]	20	3

TABLE 9. Simulation results for different scenarios.

	Scenario 1	Scenario 2	Scenario 3
Requests	1673	2362	3062
Accepted requests with detour	298	349	1021
Accepted requests without detour	662	636	2041
Total reward	20734	27185	98059
Total waiting and detour times	11744	20244	58114

detours of varying durations. By comparing results for different intervals of maximum detour time, we observed that the number of accepted requests with and without detour does not reveal a significant difference. This finding attests to the effectiveness of our decision algorithm and the reasoning of the agent. Our system can make balanced decisions, considering requests requiring a detour and those that can be integrated into the initial route and passenger waiting time. Additionally, varying the number of offers and requests allowed us to explore different workloads and test our system's ability to efficiently handle an increasing volume of real-time data. By analyzing the system's performance under these different conditions, we were able to evaluate its robustness and adaptability to various demand situations.

VII. DISCUSSION

This work aims to address the dynamic ridesharing optimization problem with a primary focus on minimizing passenger waiting time and driver detour time. The obtained results demonstrate the effectiveness and the robustness of our system in making balanced decisions between drivers' constraints and passengers' constraints. This balanced decision-making capability contributes to the development of a robust dynamic ridesharing system which is adaptable and responsive in real-world scenarios.

In a departure from existing literature, our proposed modeling approach uniquely prioritizes temporal dimension, emphasizing time-related factors in the decision-making process. Unlike other works that integrate both temporal and spatial information into observations, a practice that significantly inflates the state space, our approach focuses on temporal aspects. Specifically, the information captured in the observation pertains exclusively to time-related factors, encompassing passenger waiting times and driver detour durations. Central to our framework is the assumption that time serves as a critical delineating parameter for decision-making, a departure that simplifies the problem considered. Furthermore, our work addresses two instances of the dynamic ridesharing optimization problem: the one-to-one ridesharing matching problem and the one-to-many ridesharing matching problem. This dual perspective offers a more nuanced understanding of the system dynamics, enhancing the adaptability and applicability of our proposed model across diverse ridesharing scenarios.

Based on the promising results obtained and the extensive review of related literature, our approach stands as a noteworthy contribution to the field of dynamic ridesharing optimization. The balanced decision-making capability, coupled with a focus on the temporal dimension, positions our system as an efficient and adaptable solution, offering substantial benefits to both passengers and drivers in dynamic ridesharing environments.

VIII. CONCLUSION AND FUTURE WORK

Dynamic ridesharing offers great flexibility of use, allowing for the swift identification of a shared ride based on the positions of potential drivers and passengers along a given route. Ridesharing systems are inherently dynamic and complex. Their complexity primarily lying in the matching of users subject to several constraints: spatiotemporal, waiting times, detour times, etc.

In this work, our modeling comprises two key aspects. Initially, we focus on modeling the one-to-one dynamic ridesharing problem, incorporating inherent constraints and the objective of minimizing passengers' waiting time. Subsequently, we extend our modeling to address the one-to-many dynamic ridesharing scenario, introducing additional inherent constraints and objectives that involve minimizing both passengers' waiting times and drivers' detour times. This comprehensive approach, based on components of the Markov decision process and rooted in reinforcement learning theory, primarily centers on the temporal dimension. The overarching goal is to optimize the dynamic constrained matching process, ultimately leading to a reduction in passenger waiting times and minimizing detour times for drivers, thereby enhancing the overall efficiency of the ridesharing system and increasing the number of successfully completed requests.

The proposed modeling was validated on the basis of a simulator developed to ensure the dynamic aspect of the system. The results obtained through the various experiments demonstrated the effectiveness, robustness, and ability of our system to respond to diverse requests subject to various constraints. These outcomes confirmed our hypothesis that time plays a crucial role in decision-making. It is encouraging to note that our model successfully reduced passenger waiting times and minimized driver detour times, providing an efficient and beneficial solution for all parties involved in the ridesharing process.

This work can be extended in many different directions. One potential is to address conflicting objectives. The inherent tension between minimizing passenger waiting times and reducing driver detour times necessitates a more nuanced approach. One avenue worth exploring is the application of multi-objective reinforcement learning, allowing the system to find optimal solutions that balance these conflicting objectives. Scalability is another important perspective to consider. As ridesharing systems evolve and cater to an increasing number of users, it is imperative to assess the scalability of our proposed model.

User preferences play a pivotal role in the success of ridesharing systems. Future work should delve into incorporating personalized preferences and constraints, offering users a more tailored experience.

REFERENCES

[1] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. P. Joshi, and W. Cho, "A review on autonomous vehicles: Progress, methods and challenges," *Electronics*, vol. 11, no. 14, p. 2162, Jul. 2022.

[2] K. H. Ting, L. S. Lee, S. Pickl, and H.-V. Seow, "Shared mobility problems: A systematic review on types, variants, characteristics, and solution approaches," *Appl. Sci.*, vol. 11, no. 17, p. 7996, Aug. 2021.

[3] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transp. Res. B, Methodol.*, vol. 57, pp. 28–46, Nov. 2013.

[4] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *Eur. J. Oper. Res.*, vol. 223, no. 2, pp. 295–303, Dec. 2012.

[5] L. D. C. Martins, R. de la Torre, C. G. Corlu, A. A. Juan, and M. A. Masmoudi, "Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms," *Comput. Ind. Eng.*, vol. 153, Mar. 2021, Art. no. 107080.

[6] A. Mourad, J. Puchinger, and C. Chu, "A survey of models and algorithms for optimizing shared mobility," *Transp. Res. B, Methodol.*, vol. 123, pp. 323–346, May 2019.

[7] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "Making dynamic ride-sharing work: The impact of driver and rider flexibility," *Transp. Res. E, Logistics Transp. Rev.*, vol. 91, pp. 190–207, Jul. 2016.

[8] X. Li, J. Gao, C. Wang, X. Huang, and Y. Nie, "Ride-sharing matching under travel time uncertainty through data-driven robust optimization," *IEEE Access*, vol. 10, pp. 116931–116941, 2022.

[9] V. Armant and K. N. Brown, "Fast optimised ridesharing: Objectives, reformulations and driver flexibility," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112914.

[10] W. Zeng, Y. Han, W. Sun, and S. Xie, "Exploring the ridesharing efficiency of taxi services," *IEEE Access*, vol. 8, pp. 160396–160406, 2020.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[12] Z. Qin, H. Zhu, and J. Ye, "Reinforcement learning for ridesharing: An extended survey," *Transp. Res. C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103852.

[13] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[14] Z. Wang, Z. Qin, X. Tang, J. Ye, and H. Zhu, "Deep reinforcement learning with knowledge transfer for online rides order dispatching," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 617–626.

[15] J. Ke, F. Xiao, H. Yang, and J. Ye, "Learning to delay in ride-sourcing systems: A multi-agent deep reinforcement learning framework," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2280–2292, May 2022.

[16] N. D. Kullman, M. Cousineau, J. C. Goodson, and J. E. Mendoza, "Dynamic ride-hailing with electric vehicles," *Transp. Sci.*, vol. 56, no. 3, pp. 775–794, May 2022.

[17] N. Alisoltani, L. Leclercq, and M. Zargayouna, "Real-time ride-sharing systems performance considering network congestion," in *Proc. 8th Symp. Eur. Assoc. Res. Transp.*, Sep. 2019, p. 6.

[18] X. Wang, N. Agatz, and A. Erera, "Stable matching for dynamic ride-sharing systems," *Transp. Sci.*, vol. 52, no. 4, pp. 850–867, Aug. 2018.

[19] N. Masoud and R. Jayakrishnan, "A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system," *Transp. Res. B, Methodol.*, vol. 106, pp. 218–236, Dec. 2017.

[20] S. Ben Cheikh-Graiet, M. Dotoli, and S. Hammadi, "A Tabu search based metaheuristic for dynamic carpooling optimization," *Comput. Ind. Eng.*, vol. 140, Feb. 2020, Art. no. 106217.

[21] M. A. Masmoudi, M. Hosny, E. Demir, and E. Pesch, "Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem," *J. Heuristics*, vol. 26, no. 1, pp. 83–118, Feb. 2020.

[22] X. Tang, F. Zhang, Z. Qin, Y. Wang, D. Shi, B. Song, Y. Tong, H. Zhu, and J. Ye, "Value function is all you need: A unified learning framework for ride hailing platforms," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 3605–3615.

[23] M. Li, Z. Qin, Y. Jiao, Y. Yang, J. Wang, C. Wang, G. Wu, and J. Ye, "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proc. World Wide Web Conf.*, May 2019, pp. 983–994.

[24] I. Jindal, Z. T. Qin, X. Chen, M. Nokleby, and J. Ye, "Optimizing taxi carpool policies via reinforcement learning and spatio-temporal mining," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1417–1426.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

HIBA ABDELMOUMÈNE received the bachelor's, master's, and Ph.D. degrees in computer science from Badji Mokhtar University, Annaba, Algeria, in 2008, 2010, and 2016, respectively. Since October 2016, she has been a Senior Lecturer with the Department of Computer Science, Université 8 Mai 1945 Guelma, Guelma, Algeria. She is also affiliated as a Research Member with the Electronic Document Management Laboratory (LabGED), Badji Mokhtar University. Her research interests include artificial intelligence, optimization, decision-theoretic planning, and uncertainty.

CHEMESSE ENNEHAR BENCHERIET received the Ph.D. degree in signal processing from Badji Mokhtar University, Annaba, Algeria, in 2007. She is currently a Senior Lecturer (Class A) with the Computer Science Department, Université 8 Mai 1945 Guelma. She is the author of over 40 peer-reviewed scientific publications and conference papers and has contributed in eight national and international research projects in different fields. She is the author of books entitled *Détection et Reconnaissance de visages: De l'importance des caractéristiques géométriques faciales et de la couleur de peau* edited by EUE, in 2012, and *Codage et Représentation de l'Information* edited by OPUG, in 2015. Her research interests include machine learning and deep learning, face detection and recognition, fake faces detection, skin detection and classification, intelligent transportation systems, and analysis and classification of remote sensing images. She has participated as a member of program and organizing comity in more than 20 national and international conferences. She has contributed as a Reviewer in renowned journals, such as *IET Computer Vision*, *IET Biometrics*, *IET Image Processing*, *Electronic Letters* (IET), IEEE Access, and *The Visual Computer* (Springer).

HABIBA BELLEILI received the Ph.D. degree in computer science from Badji Mokhtar University, Annaba, Algeria. She is currently a Professor with Badji Mokhtar University, where she is also a Research Member with the Electronic Document Management Laboratory (LabGED). Her research interests include pattern classification and optimization in the medical field, multi-agent systems, and decision making.

ISLEM TOUATI received the bachelor's and master's degrees from Université 8 Mai 1945 Guelma, Guelma, Algeria, in 2020 and 2022, respectively.

CHAYMA ZEMOULI received the bachelor's and master's degrees from Université 8 Mai 1945 Guelma, Guelma, Algeria, in 2021 and 2023, respectively.