

Green Edge Intelligence for Smart Management of a FANET in Disaster-Recovery Scenarios

Giuseppe Faraci, Santi Agatino Rizzo , and Giovanni Schembra 

Abstract—Disruption of the ground communication infrastructure in emergency scenarios makes post-disaster rescue operations very complicate. This paper proposes to use edge intelligence to support rescue operators in managing these emergency scenarios. A set of Unmanned Aerial Vehicles (UAV), organized as Flying Ad-Hoc network (FANET), autonomously takeoff and land to provide emergency operators with edge computing services. A charging station for batteries is supplied by a renewable-energy generator. The FANET Controller applies model-based Reinforcement Learning to decide how many UAVs have to take off, according to the current edge-computing service requests and the power availability, and a forecast of them. The optimal management policy has to provide the necessary level of edge-computing avoiding wide use of satellite channels in a short-time horizon during low green-energy generation and high service request periods. Results highlight that the optimal policy is an efficient modification of the greedy one, i.e., the policy enabling the takeoff of all the necessary UAVs without being care of challenging events in the future. A deep analysis has revealed that the level of modification depends on the combination of the edge-computing service request and the green power availability.

Index Terms—5G, emergency networks, energy-aware networks, green edge intelligence, reinforcement learning, UAV.

I. INTRODUCTION

NATURAL disasters, such as earthquakes, hurricanes and floods, occur frequently in many areas of the world, causing immense damage to people's lives and properties [1], [2], [3]. During these events, multiple parties have to make quick decisions for rescues, but the disruption of ground communication infrastructures and other interconnection infrastructures such as power facilities makes the communication between the disaster area and the outside world challenging to facilitate rescue operations [4], [5], [6]. In these scenarios, reestablishing the communication system is critically important to support the timely exchange of information regarding locations, events, and situational severity for victims and rescuers. Until recently, emergency communications have mainly been based on narrow-band telecommunications technologies, enabling voice-centric

Manuscript received 9 February 2022; revised 16 July 2022; accepted 30 September 2022. Date of publication 6 February 2023; date of current version 14 March 2023. The work was supported in part by MIUR, Italy under the PRIN “Liquid Edge” and in part by the PIACERI project “OMNIA”, and the POR S6 project. The review of this article was coordinated by Dr. Ming Li. (Corresponding author: Giovanni Schembra.)

Giuseppe Faraci and Santi Agatino Rizzo are with the DIEEI - University of Catania, 95124 Catania, Italy (e-mail: giuseppe.faraci@unict.it; santi.rizzo@unict.it).

Giovanni Schembra is with the DIEEI - University of Catania, 95124 Catania, Italy, and also with the CNIT – Research Unit of Catania, 95124 Catania, Italy (e-mail: giovanni.schembra@unict.it).

Digital Object Identifier 10.1109/TVT.2022.3217331

services only, but with very limited data transmission capability and interoperability [7]. In the last decade, thanks to the great advancement in personal communication technologies, from 3G to 4G [8], [9], broadband wireless technologies have begun to be used in emergency communications, allowing users to send and receive crucial voice, video, and other kinds of data [10], [11]. However, conventional emergency communication systems are mostly infrastructure-based, so presenting difficulties in maintaining communication services when the communication infrastructure is severely damaged [12].

Unmanned Aerial Vehicles (UAV) have been demonstrated to address this problem by providing a quick turn-around option. During the critical first 72 hours, UAVs can be used in emergency scenarios for tasks such as situational awareness, deploying communication systems, or search and rescue missions [13], [14], [15]. Moreover, small base stations can be mounted onboard them to rapidly deploy an easy-to-operate and responsive emergency ad-hoc network [16], providing broadband communication services anytime and anywhere [8], [17], [18], [19], [20], [21], [22], [23]. This way, UAVs have gained importance as a low-cost tool for post-disaster monitoring and reconstruction management for first responders, planners, and elected officials. Now, the 5th generation broadband cellular networks (5G) have a strong potential to meet the high demands of up-to-date emergency communications for reliability and resilience. To this purpose, in the last few years, some papers have proposed to integrate edge computing facilities onboard of UAVs, making them Multi-Access Edge Computing (MEC) UAVs [24], [25], [26], in order to provide users with edge computing in areas not covered by the structured 5G Internet. However, as observed in some of these works, the problem of limited UAV flight lifetime due to battery charge duration is further exacerbated due to the power consumption of the computing elements, which is added to the engine power consumption. For this reason, a widely adopted solution is the use of a storm of UAVs [27] connected to create a Flying Ad-Hoc network (FANET) [28]. Further insights regarding FANETs can be found in [29], [30], while a reader can refer to [31], [32], [33] for issues related to the management of FANET topology.

The use of a FANET allows that, when the battery charge level of a UAV is low, it can land for charging or substituting the battery, while the service is maintained active by the remaining ones. Of course, the higher the number of UAVs that are on the ground for charging, the worse the service provided to the users. This problem is of crucial importance in the considered emergency scenarios, since the scarceness of edge computing

and connectivity can strongly deteriorate rescue missions and emergency management activities. A possible solution to this problem is the use of a satellite channel to offload jobs to a remote data centre. However, this can be considered as a backup non-optimal solution, aimed at not losing jobs that cannot be managed by the FANET, but with evident disadvantages regarding costs, high delay and privacy issues. Another important element that must be considered in emergency scenarios is the lack of connection to the power grid. A solution is the use of a mobile renewable generator [34] to charge the UAV batteries, but available energy is limited and time-variant.

Therefore, the target of this paper is to apply artificial intelligence in the management of FANETs to maximize edge-computing services they provide in post-disaster environments. This is compliant with the new paradigm of *Edge Intelligence* (EI), which aims at using Artificial Intelligence (AI) at the network edge for management edge-computing facilities, in this case, provided by the UAVs of a FANET [35]. To this purpose, Reinforcement Learning (RL) is applied to decide the takeoff of the UAVs with a full battery, taking into account the time-variant behavior of both the edge-computing service requirements and the power availability from a wind generator (WG). More specifically, the contribution of this paper is three-fold.

- 1) Proposing an unmanned UAV-based platform to provide emergency operators with edge-computing service capabilities in emergency scenarios. The platform is constituted by a WG-powered charging station (CS), a set of UAVs and backup batteries for immediate swapping and takeoff of landing UAVs [36]. A FANET Controller (FC) decides how many UAVs have to autonomously takeoff among the necessary and ready ones.
- 2) Proposing a decision-making strategy maximizing a reward function that depends on the ability to avoid the use of backup satellite transmission to remote data centres for job offload, especially in a short-time horizon.
- 3) Defining a discrete-time analytical model of the system to provide the RL-based FC with a Markov Decision Process (MDP), to support its actions. The RL uses the model to forecast power availability and service requests in order to find the optimal management policy that the FC must apply to maximize the reward function.

To the best of our knowledge, this is the first paper that proposes a framework to provide edge intelligence in a disaster area where neither computing nor networking facilities nor the power grid are available.

An extensive performance analysis is presented to evaluate the performance of the proposed system. A comparison among the proposed optimal policy, the greedy one (i.e., the policy enabling the take-off of all the necessary UAVs without taking care of more challenging events in the future) and other policies is carried out in some case studies. The comparison has highlighted the ability of the optimal policy to avoid a wide use of satellite channels in a short time horizon.

The paper is organized as follows. [Section II](#) describes the reference system, constituted by an emergency area covered by a FANET that provides it with connectivity and edge-computing services. [Section III](#) defines the Markov model of the considered

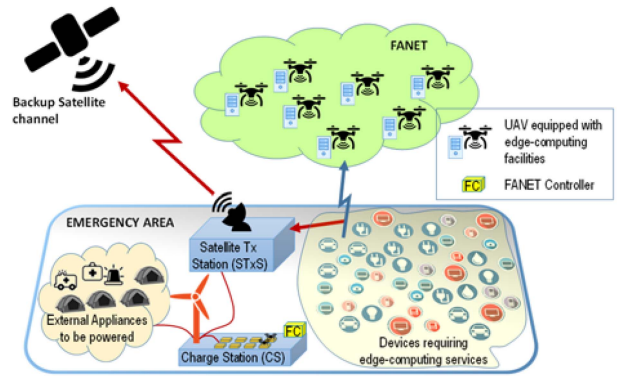


Fig. 1. Reference System.

system and derives the transition probability and the reward matrices of the MDP used to support the decision-making process of the FC. [Section IV](#) analytically derives the main performance parameters characterizing the system behaviour. [Section V](#) presents a use case with some numerical analysis. Finally, [Section VI](#) concludes the paper.

II. REFERENCE SYSTEM DESCRIPTION AND ASSUMPTIONS

The system considered in this paper is a post-disaster emergency scenario. As sketched in [Fig. 1](#), it is made up of a huge number of user devices (sensors, actuators and user equipment) deployed on the ground in a wide emergency area. They require connectivity and edge-computing services to support the monitoring and reconstruction activities of first responders, planners, elected officials and other people involved in emergency management. It is assumed that the considered area is completely isolated, that is, not connected to the structured Internet and not supplied by the power grid. To this purpose, a Flying Aerial Network (FANET) is used to provide the ground devices with the required edge-computing services. The FANET is realized by a fleet of N UAVs, each equipped with an edge-computing facility, and a TX/RX element to allow UAVs to communicate to each other and with the devices on the ground. Communications between ground devices and UAVs and among UAVs are exclusively in charge of the UAVs, with no need for terrestrial facilities.

Since the focus of the paper regards takeoff management of charged UAVs of the same FANET, it can be not considered any problem of communications among UAVs and between UAVs and ground devices. The literature presents various solutions to solve this problem, for example [\[37\]](#) reports some details.

Moreover, UAVs are assumed equal to each other in flying, connectivity and computing characteristics.

In the periods when the FANET is not able to satisfy the whole computation load required by the ground devices, jobs to be processed are sent to a mobile Satellite Transmission Station (STxS) to offload jobs through a backup satellite connection to a remote data centre. However, this solution is necessary to avoid job loss during overload periods. On the other hand, it has to be considered a second-rate solution because it adds costs (due to the use of both a satellite channel, which is a precious resource

and the computation resources in the remote data centre) and it is against the principles of edge computing, which is applied to achieve low latency reaction and privacy preservation. For this reason, the main objective of the proposed framework is maximizing matching between FANET resource availability and time-variant edge-computing service requests, while offloading through the STxS is to be avoided as much as possible.

In the sequel, for the sake of simplicity, the amount of edge-computing service provided by each UAV is defined as the unit of service. Therefore, the time-variant service requested by ground devices and the service provided by the FANET will be expressed in terms of number of UAVs. Likewise, also the residual service, i.e., the one required to the remote data centre through the backup satellite channel because not supported by the FANET, will be expressed in UAV service units.

The main issue of a UAV is the short flying time due to the limited battery charge duration, although the technology trend is fast moving towards a longer duration thanks to high energy density batteries. For this reason, a CS equipped with C_p charging points (CPs) is installed on the ground inside or very close to the disaster area that has to be serviced by the FANET.¹

Each CP can charge one battery at a time. The CS is supplied by a mobile WG. The power available from this WG varies along the time due to the variability of the wind in the area. In order to reduce the number of UAVs waiting for charging on the ground, some high energy density and extreme lightweight batteries, B , are also made available [38]. To avoid any human intervention, an automatic battery swap mechanism has been considered [36]. When there is no CP with a full battery, the landed UAV must wait for battery charging to take off again. The average waiting time depends on the power available from the WG and the number of UAVs waiting for a charged battery. When the power available from the WG is sufficient to charge the CPs with a waiting UAV, the remaining power is delivered to other CPs in order to charge batteries that will be used by future landing UAVs. Finally, when the power available from the WG is greater than the current number of CPs to be supplied, the green power surplus can be used to supply external appliances installed in the same area.

Since the devices on the ground need a time-variant edge-computing service, the number of UAVs necessary to guarantee this service changes over time. Moreover, the fact that the WG produces a time-variant power has to be taken into account since this determines the number of available charged batteries

¹How close placing the CS to the disaster area strongly depends on the specific scenario of the disaster, its location, and the availability of space for installation. However, consider that all that is needed to realize the proposed system, including a small wind power generator (i.e. a micro wind turbine) can be transported onsite by a truck or, if access roads to the disaster area are interrupted, by a helicopter, which for sure will be available because used for the transportation of other materials needed to the same area for other purposes. An area of a few hundred square meters is enough for installing the entire framework. For this reason, it is expected that the CS will be installed inside or very close to the disaster area. This will enable them to provide edge computing services also in a subinterval while they are leaving or moving toward the CS. Regarding the battery duration problem, there are currently very limited solutions that enable 2-3 hours flight time but they are very expensive. On the other hand, it is expected that the problem will be mitigated in a medium-term period by the development of more efficient UAVs that will allow lower power consumption during hovering [39], [40] and the improvement of battery technologies that are also pushed by the strong interest in the electrification trend of the automotive market.

for UAVs that land. In the periods when the number of UAVs on air is less than the required edge-computing service, the Transmission Station has to use the expensive backup satellite channel to offload jobs to a remote cloud. Thus, managing UAV takeoff among the UAVs that are ready on the ground with a charged battery is challenging to face future potential critical situations that would cause high costs for using the backup satellite channel. This decision must be taken by adopting a *policy* maximizing the cumulative reward, that is a long-term reward, $Z(n)$, defined as the total discounted reward from time slot n :

$$Z(n) = \sum_{k=0}^{\infty} \gamma^k \cdot R(n+k+1) \quad (1)$$

where $R(n)$ is the immediate reward achieved in the time slot n , while γ is the *discount factor*, with $\gamma \in [0, 1]$. This is an input parameter that informs the decision-maker about how much it should care about the immediate reward as compared to future rewards. Values of γ close to zero mean that it has to take care of the immediate reward. On the contrary, a value of γ close to 1 means that the decision-maker is far-sighted, i.e., it has to take care of all future rewards. For this reason, it uses this reward instead of the immediate one, in order not to be greedy by taking actions associated with the maximum reward at the current time only, but it has to plan the future. It is better to sacrifice immediate reward to gain a high cumulative long-term reward.

In the considered scenario, the immediate reward $R(n)$ in (1) is a measure of the FANET ability to satisfy the edge-computing request avoiding satellite usage. Therefore, it is defined as a function of the difference, d , between the amount of edge-computing service required by the ground devices and the edge-computing service that the flying UAVs provide them. More specifically, the goal is to avoid that the edge-computing service provided by the FANET becomes, in some instants, too much lower than the required service, thus requiring offloading through the backup satellite channel, which is seen as a penalty caused by both additional costs, delay increasing and privacy detriment. Moreover, it is assumed that the satellite owner applies an incremental price for the amount of traffic to be transmitted. Therefore, for a given number of jobs to be offloaded over a long period, they shouldn't be concentrated in a small interval. Finally, from a privacy point of view, it is also better avoiding to offload many jobs produced in a small interval since they are correlated to each other, while the correlation of information related to different periods is lower. Consequently, the immediate reward function is defined as follows to avoid high satellite usage in the short-term as follows:

$$R(n) = \begin{cases} -d^2 & \text{if } d > 0 \\ 0 & \text{otherwise} \end{cases} = -[\max\{d, 0\}]^2 \quad (2)$$

This way, when d is positive, that is when the FANET is not able to satisfy the request, the reward is negative. On the contrary, a null reward is considered when the FANET satisfies the demand ($d \leq 0$).

In the system, the decision-maker role is in charge of the FC, an entity running in the CS to coordinate takeoff of the ready

UAVs. To this purpose, the FC takes its decisions by using the best management policy found thanks to RL [41].

It is worth noting that, a centralized solution, that is, a single-agent RL centralized on the FC, for two main reasons. On one side, using multi-agent RL implies running one agent on each UAV, which may present problems of coordination. Moreover, it could need dedicated hardware (e.g., with a GPU board) on each UAV to run computationally hungry applications for training. Besides the costs of this hardware, this could be too heavy to be put on-board and can influence flight autonomy for its power consumption. On the other side, the FC can access easily all the information needed to train the model and centrally run the RL algorithm since, in this specific scenario where the number of UAVs is not so much huge, it does not incur the curse of dimensionality problems that often motivate multi-agent approach.

More specifically, the FC uses a model-based RL formulated as a Markov Decision Process (MDP) in Section III. Thanks to this model, the long-term cumulative reward can be computed. Therefore, at the planning stage (off-line mode) of the FC, the best policy is found by selecting the one whose decisions enable cumulative reward maximization. After that, at the operation stage, the FC works online. It observes the system conditions (WG power, edge-computing service request, state of the various UAVs and batteries), then identifies the most similar state of the discrete model and, finally, chooses its action, consisting in deciding the number of UAVs to left on the ground among the ones that are ready for takeoff, according to the best policy found at the design stage.

III. SYSTEM MODEL

In this section, a discrete-time Markov model representing the MDP necessary to solve the RL problem described in the previous section has been defined. Table I summarizes the main notation.

The environment is described by the state of WG, edge-computing request and FANET (number of UAVs on air, on ground with empty and full battery), the number of full and empty batteries. Since the FC always knows the state of the considered environment, the environment is said fully observable. A MDP Σ that describes the environment also depends on the actions associated with the states of the system. A policy is a specific set of actions. More in deep, for a given policy ρ specifying an action a for each state s_Σ , a MDP is completely defined by the tuple $(\mathfrak{S}^{(\Sigma)}, \mathfrak{S}^{(A)}, P^{(\Sigma|\rho)}, \Psi^{(\Sigma|\rho)}, \gamma)$, where $\mathfrak{S}^{(\Sigma)}$ is the system state space, $\mathfrak{S}^{(A)}$ is a finite set of actions, $P^{(\Sigma|\rho)}$ is the state transition probability matrix, $\Psi^{(\Sigma|\rho)}$ is the immediate reward matrix, and γ is the discount factor. Moreover, the value of a state s'_Σ is defined as the expected return when starting from s'_Σ and following ρ thereafter, that is:

$$v_\rho(s'_\Sigma) = E_\rho \left\{ \sum_{k=0}^{\infty} \gamma^k R(n+k) \mid S^{(\Sigma)}(n-1) = s'_\Sigma \right\} \quad (3)$$

The matrix $P^{(\Sigma|\rho)}$ depends on the policy ρ that specifies the action for each starting state. Therefore, its generic element, representing the transition probability from the generic state s'_Σ to the generic state s''_Σ , provided that, according to the policy

TABLE I
MAIN NOTATION

N	Total number of available UAVs
B	Total number of available backup batteries
$Z(n)$	Total discounted reward from time slot n
$R(n)$	Immediate reward achieved in the time slot n
d	Difference between the required edge-computing service and the provided one
$\mathfrak{S}^{(\Sigma)}$	System state space
$\mathfrak{S}^{(A)}$	Space of actions
$P^{(\Sigma \rho)}$	State transition probability matrix
$\Psi^{(\Sigma \rho)}$	Immediate reward matrix
$S^{(\Sigma)}(n)$	State of the overall system in the time slot n
$S^{(\Sigma)}(n)$	State of the overall system in the time slot n
$S^{(G)}(n)$	State of the the renewable power availability process $G(n)$ in the time slot n
$S^{(M)}(n)$	State of the process $M(n)$ representing the edge-computing service request coming from the ground devices system in the time slot n
$S^{(B)}(n)$	State of the process $B(n)$ representing the number of backup batteries that are fully-charged in the time slot n
$\underline{S}^{(D)}(n)$	Three-dimensional array counting the number of UAVs whose battery is empty, $S_E^{(D)}(n)$, the ones with a charged battery, $S_R^{(D)}(n)$, and the flying ones, $S_A^{(D)}(n)$
$A(n)$	Action performed in the time slot n
$\underline{\pi}^{(\Sigma)}$	Steady-state system probability array
ρ^*	Optimal policy
$\bar{\mathfrak{R}}$	Mean immediate reward
\wp_k	k -level penalty for remote offloading

ρ , the action a is performed at the beginning of the time slot n according to the starting state s'_Σ , is:

$$P_{[s'_\Sigma, s''_\Sigma]}^{(\Sigma|a)} = \Pr \left\{ S^{(\Sigma)}(n) = s''_\Sigma \mid S^{(\Sigma)}(n-1) = s'_\Sigma, A(n) = a \right\} \quad (4)$$

Likewise, the generic element of the reward matrix, representing the immediate reward received performing the action a at the slot n when the system transits from s'_Σ to s''_Σ , is:

$$\Psi_{[s'_\Sigma, s''_\Sigma]}^{(\Sigma|a)} = E \left\{ R(n) \mid S^{(\Sigma)}(n-1) = s'_\Sigma, S^{(\Sigma)}(n) = s''_\Sigma, A(n) = a \right\} \quad (5)$$

Section III-A will define some mathematical notation to model each component of the whole system. Then, Section III-B will describe the MDP of the whole system.

The optimal policy can be derived by means of a dynamic programming algorithm called *value iteration* (see Algorithm 1), where the term $v(s_\Sigma)$ indicates the value of the generic state

Algorithm 1: Value iteration to derive the optimal policy.

Initialize the value-function array v arbitrarily
 Repeat
 For each $s'_\Sigma \in \mathfrak{S}(\Sigma)$
 $temp \leftarrow v(s'_\Sigma)$
 $v(s'_\Sigma) \leftarrow \max_a \sum_{s''_\Sigma} P_{[s'_\Sigma, s''_\Sigma]}^{(\Sigma|a)} \cdot [\Psi_{[s'_\Sigma, s''_\Sigma]}^{(\Sigma|a)} + \gamma v(s''_\Sigma)]$
 $\Delta \leftarrow \max\{\Delta, |temp - v(s'_\Sigma)|\}$
 until $\Delta < \varepsilon$ (ε is a small positive number)
 Output a deterministic policy, ρ^* , such that:
 $v_*(s'_\Sigma) =$
 $\arg \max_a \sum_{s''_\Sigma} P_{[s'_\Sigma, s''_\Sigma]}^{(\Sigma|a)} \cdot [\Psi_{[s'_\Sigma, s''_\Sigma]}^{(\Sigma|a)} + \gamma v(s''_\Sigma)]$,
 for all $s'_\Sigma \in \mathfrak{S}(\Sigma)$

$s_\Sigma \in \mathfrak{S}(\Sigma)$ under a generic policy ρ , while $v_*(s_\Sigma)$ indicates the value of the same state under the optimal policy ρ^* .

A. Markov Model of the System

In this section, the environment of the MDP is modelled with a discrete-time multi-dimensional Markov model. As already said, the time slot is defined as the UAV battery *charging time*, Δ . For the sake of simplicity, it is assumed that the mean flight time, defined as the time ranging between the UAV take-off instant and the landing instant, so including climb, cruise, and descendent phases, is, on average, a multiple, H , of the charging time, being H an integer. Therefore, the flight time has a time duration of $H \cdot \Delta$.

The behaviour of the overall system in the generic time slot n is represented as:

$$\underline{S}^{(\Sigma)}(n) = \left(S^{(G)}(n), S^{(M)}(n), S^{(B)}(n), \underline{S}^{(D)}(n) \right) \quad (6)$$

whose component processes are defined below.

The first two elements of $\underline{S}^{(\Sigma)}(n)$ are the states of the two independent processes that influence the behaviour of the system: $G(n)$, characterizing the renewable power availability, is defined as the number of UAVs that the WG is able to charge at the time slot n ; $M(n)$, characterizing the time-variant edge-computing service request coming from the ground devices, is defined as the number of UAVs that are required to satisfy the request.

The process $M(n)$ is modelled as a Markov chain characterized by its state, $S^{(M)}(n)$, and its transition probability matrix, $P^{(M)}$, whose generic element is defined as follows:

$$P_{[s'_M, s''_M]}^{(M)} = \Pr \left\{ S^{(M)}(n) = s''_M \mid S^{(M)}(n-1) = s'_M \right\} \quad (7)$$

for each s'_M and s''_M belonging to the set of states $\mathfrak{S}^{(M)}$.

In order to model the renewable-energy generation process $G(n)$, as in [42], the Switched Batch Bernoulli Process (SBBP) model, which is the most general Markov-modulated process in the discrete-time domain, being able to capture both first- and second-order statistics of a process, is used. According to [43], it is defined by: $S^{(G)}(n)$, representing the state of the underlying Markov chain at the generic time slot n ; $\mathfrak{S}^{(G)}$, representing the state space of the underlying Markov chain, that is, the set of all the possible states that the process $S^{(G)}(n)$ can assume;

$\Psi^{(G)}$, representing the space of values of the process $G(n)$, that is, the set of all the possible values that the process $G(n)$ can assume; $P^{(G)}$, representing the state transition probability matrix of the underlying Markov chain. Its generic element is defined as done in (7) for the process $M(n)$; $B^{(G)}$, representing the value occurrence probability matrix, whose generic element is defined as follows:

$$B_{[s_G, g]}^{(G)} = \Pr \left\{ G(n) = g \mid S^{(G)}(n) = s_G \right\} \quad (8)$$

Before modeling the FANET behavior, the behavior of each UAV is modelled as a 3-state Markov model, as depicted in Fig. 2. When a flying UAV on-air (state A) reaches a low state of charge (SoC), it lands on a CP of the Charging Station and waits for charging. This is represented by the permanence on the state ‘‘Empty Battery’’ (E), where it remains for some time slots until the CP is not supplied. When the CP where the UAV is placed is supplied, battery charging occurs in a single time slot, given the choice of set the time slot equal to the battery charging time. Therefore, in the successive time slot, the UAV enters the state ‘‘Ready to takeoff’’ (R). On the other hand, it immediately enters this state when there is a full battery in the CS. The UAV will stay in this state until the FC does not authorize its takeoff, thus the cycle restarts. Therefore, the transitions $E \Rightarrow E$ and $E \Rightarrow R$ depend on the available power from the WG, while the transitions $R \Rightarrow R$ and $R \Rightarrow A$ depend on the FC decision.

Therefore, the model of the whole FANET behavior can be represented by the state process $\underline{S}^{(D)}(n)$, a three-dimensional array counting the number of UAVs whose battery is empty, $S_E^{(D)}(n)$, the ones with a charged battery and so ready for takeoff, $S_R^{(D)}(n)$, and the ones flying on air, $S_A^{(D)}(n)$, that is:

$$\underline{S}^{(D)}(n) = \left(S_E^{(D)}(n), S_R^{(D)}(n), S_A^{(D)}(n) \right) \quad (9)$$

Of course, the sum of the numbers of UAVs in each state gives the total number of UAVs in the FANET:

$$S_E^{(D)}(n) + S_R^{(D)}(n) + S_A^{(D)}(n) = N \quad (10)$$

Let us highlight that the value of $S_R^{(D)}(n)$ represents the upper bound of the action space in the slot n . Therefore, the whole action space can be defined as follows:

$$\mathfrak{S}^{(A)} = \left\{ \begin{array}{l} a \text{ such that } a \leq S_R^{(D)}(n), \forall \underline{S}^{(\Sigma)}(n), \text{ with} \\ \underline{S}^{(\Sigma)}(n) = \left(S^{(G)}(n), S^{(M)}(n), S^{(B)}(n), \underline{S}^{(D)}(n) \right) \\ \text{and } \underline{S}^{(D)}(n) = \left(S_E^{(D)}(n), S_R^{(D)}(n), S_A^{(D)}(n) \right) \end{array} \right\} \quad (11)$$

Finally, the last process to be modelled is $B(n)$, representing the number of backup batteries that are fully charged and, consequently, ready to be mounted on a UAV for takeoff. Its state dynamics, indicated as $S^{(B)}(n)$, depend on the state of the other processes, as described later. Recalling that, the total number of available backup batteries is B , then the state space of the process $S^{(B)}(n)$ is $\mathfrak{S}^{(B)} = [0, B]$.

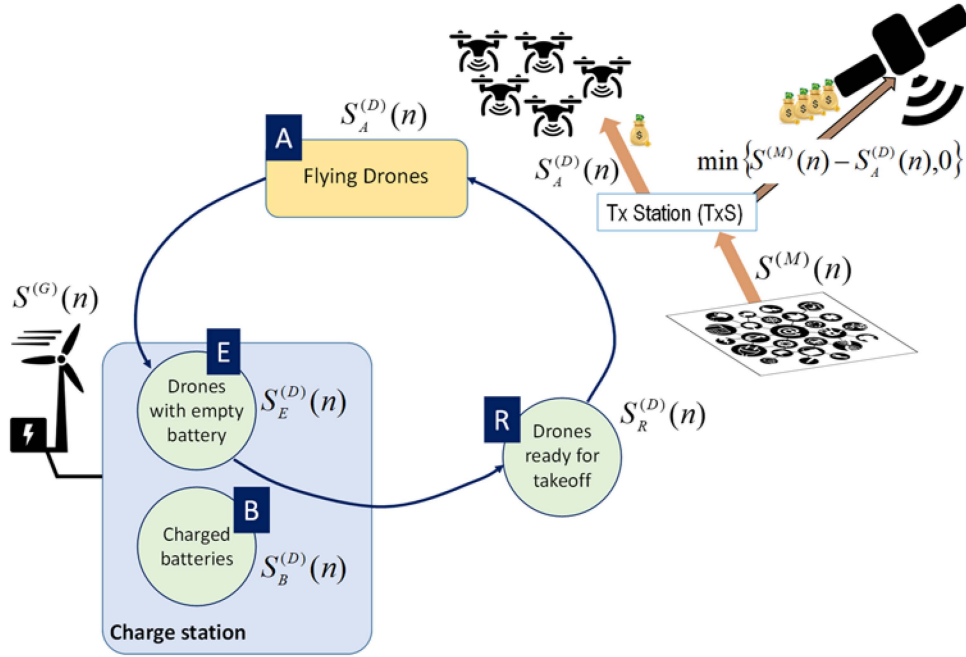


Fig. 2. Model of UAV and battery charge behavior according to computing request.

It is worth noticing that the complexity is given by the size of the state space, which is calculated as the Cartesian product of the state space of the component processes. However, since the model solution is carried out offline, it does not create any convergence or time resolution problems.

B. Transition Probability and Reward Matrices of the MDP

To define the *transition probability matrix* and the *short-term reward matrix* for the Markov chain $\underline{S}^{(\Sigma)}(n)$, it is defined the state at the time slot $n-1$ as $\underline{s}'_{\Sigma} = \underline{S}^{(\Sigma)}(n-1) = (s'_G, s'_M, s'_B, \underline{s}'_D)$, and the state at the time slot n as $\underline{s}''_{\Sigma} = \underline{S}^{(\Sigma)}(n) = (s''_G, s''_M, s''_B, \underline{s}''_D)$. The elements \underline{s}'_D and \underline{s}''_D are defined:

- $\underline{s}'_D = S^{(D)}(n-1) = [s'_E, s'_R, s'_A]$
- $\underline{s}''_D = S^{(D)}(n) = [s''_E, s''_R, s''_A]$.

The system state evolution depends on the edge-computing service amount requested by the ground devices and the power availability from the WG, assumed independent of each other, as well as on the decisions taken by the FC, which applies RL to choose how many UAVs have to remain on the ground among the ones with a full battery. In order to define the transition probability matrix of the system as a whole, the event sequence for each time slot is defined. In time slot n , it is described as shown in Fig. 3.

At the very beginning of the time slot

- 1) Decision of the action $A(n) = a$, representing the number of UAVs that the FC will leave on ground (they will not take off) among the ones that are ready for takeoff. This decision is taken by the FC based on the initial state $\underline{S}^{(\Sigma)}(n-1) = \underline{s}'_{\Sigma}$; of course, it cannot be greater than

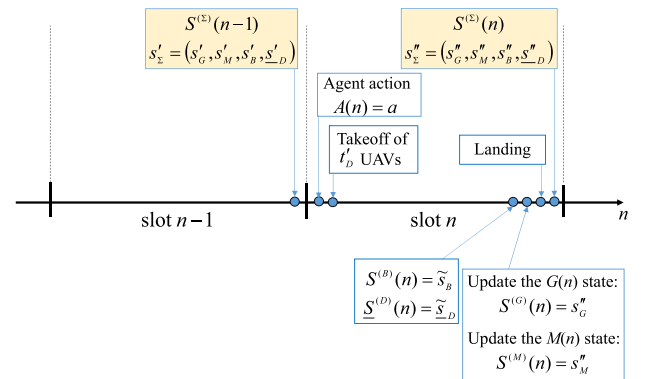


Fig. 3. Time diagram of the sequence of events in a state transition.

the number of UAVs that are ready for takeoff, that is: $a \leq s'_R$.

- 2) Takeoff of a number $S^{(D)}(n)$ of UAVs, whose value is given by:

$$S^{(D)}(n) = t'_D = \min \{s'_R - a, s'_M - s'_A\} \quad (12)$$

At the end of the time slot

- 3) Update of the number of charged batteries and the UAV state according to the number of CPs that have been supplied by the power generator for the whole duration of the time slot.
- 4) Update of the state of the WG process and the edge-computing service request process.
- 5) Landing of low-battery UAVs.
- 6) Update of the whole state $\underline{S}^{(\Sigma)}(n) = \underline{s}''_{\Sigma}$.

The generic element of the transition probability matrix for a given action a can be defined as follows:

$$P_{[s'_{\Sigma}, s''_{\Sigma}]}^{(\Sigma)}(a) = P_{[s'_G, s''_G]}^{(G)} \cdot P_{[s'_M, s''_M]}^{(M)} \cdot P_{[(s'_B, s''_B), (s''_D, s''_D)]}^{(B,D)}(s'_G, a) \quad (13)$$

The terms $P_{[s'_G, s''_G]}^{(G)}$ and $P_{[s'_M, s''_M]}^{(M)}$ in the above equation are the generic elements of the transition probability matrices of the processes $S^{(G)}(n)$ and $S^{(M)}(n)$, while $P_{[(s'_B, s''_B), (s''_D, s''_D)]}^{(B,D)}(s'_G, a)$ is the generic element of the transition probability matrix of the joint process $(S^{(B)}(n), \underline{S}^{(D)}(n))$. The first two matrices are known as the input of the problem, while the last one depends on the state of the WG and the number of UAVs that the FC has decided to leave on the ground. The derivation of the last matrix, to simplify reading, is reported in Appendix.

The generic element of the immediate reward matrix can be calculated in accordance with the immediate reward function defined in (2). It is the immediate reward for a given transition from the time slot $n - 1$, when the system is in the generic state s'_{Σ} , to the time slot n , when the system is in the generic state s''_{Σ} , and for a given action a taken according to the state s'_{Σ} . According to (2), it is defined as follows:

$$\Psi_{[s'_{\Sigma}, s''_{\Sigma}]}^{(\Sigma)}(a) = \begin{cases} -(s'_M - \tilde{s}_A)^2 & \text{if } s'_M - \tilde{s}_A > 0 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

IV. SYSTEM PERFORMANCE

In order to calculate the main system performance indices, it is necessary the steady-state probability array achieved when the optimal policy ρ^* is applied. Its generic element is defined as follows:

$$\underline{\pi}_{[s_{\Sigma}]}^{(\Sigma)} = \lim_{n \rightarrow \infty} \Pr \left\{ \underline{S}^{(\Sigma)}(n) = s_{\Sigma} \right\} \quad (15)$$

where is omitted the dependence on ρ^* for the sake of conciseness. The same will be done in the sequel. The state probability array $\underline{\pi}^{(\Sigma)}$ can be derived by solving the following steady-state equation system:

$$\begin{cases} \underline{\pi}^{(\Sigma)} \cdot P^{(\Sigma)} = \underline{\pi}^{(\Sigma)} \\ \underline{\pi}^{(\Sigma)} \cdot \mathbf{1}^T = 1 \end{cases} \quad (16)$$

where $P^{(\Sigma)}$ is the transition probability matrix calculated as in Section III with the optimal policy ρ^* derived by solving the Bellman optimality equation system [43]. Three quantities have been considered as performance parameters.

The first performance parameter is the *mean immediate reward*, $\bar{\mathfrak{R}}$. It represents the FANET edge-computing satisfaction level and is the target to be optimized. In order to evaluate how much the proposed framework is able to match the edge-computing service requests coming from the ground devices, the random variable δ is defined as the difference between the amount of edge-computing service request and the edge-computing service actually provided by the flying UAVs during a generic time slot, i.e., $\delta = s_M - \tilde{s}_A$.

When the edge-computing request suddenly decreases below the number of UAVs that are currently flying, the number of

flying UAVs is greater than the actual edge-computing service request, thus the variable δ could also assume negative values. Of course, values of δ greater than zero give penalties due to the need for remote offloading through the backup satellite link because the amount of edge-computing service provided by the FANET is less than the required one. The probability density function (pdf) of the random variable δ can be calculated as follows:

$$f_{\delta}(d) = \Pr \{ \delta = d \} = \sum_{\forall s_{\Sigma} \in \mathfrak{S}^{(\Sigma)}} (s_M - \tilde{s}_A) \cdot \pi_{[s_{\Sigma}]}^{(\Sigma)} \quad (17)$$

From (2), the mean immediate reward can be calculated as follows:

$$\bar{\mathfrak{R}} = E \{ R(n) \} = - \sum_{\forall d} [\max \{ d, 0 \}]^2 \cdot f_{\delta}(d) \quad (18)$$

The second performance parameter is the *k-level penalty for remote offloading*, φ_k . It is expressed with a polynomial law (being k the polynomial function degree) of the distance between the amount of edge-computing service request and the edge-computing service actually provided by the flying UAVs. In other terms, the higher the importance to avoid satellite usage in the short term, the higher the value to be used for k . The performance parameter is defined as follows:

$$\varphi_k = E_k \{ \delta \} = \sum_{\forall d} [\max \{ d, 0 \}]^k \cdot f_{\delta}(d) \quad (19)$$

Finally, the third performance parameter is the *Mean residual power*. It represents the mean value of the power generated by the WG which is not used by the CS. This amount of power is available to supply other loads. For example, the residual power can supply other appliances needed to manage the emergency in the same area. To this purpose, the *saved power process* $P(n)$, is defined as the difference between the power generated by the WG, $G(n)$, and the one absorbed by the CS, $C(n)$, both expressed in terms of the number of equivalent UAVs that are used as units of measure of the generated and the absorbed power. Specifically, $C(n)$ is the number of CPs that are supplied during the time slot n . Therefore, $C(n)$ is given by the sum of charging UAV batteries and other charging batteries to be ready for UAVs that will land in the future. Its first-order statistics can be represented by the pdf of its values:

$$f_P(p) = \Pr \{ P(n) = G(n) - C(n) = p \} \quad (20)$$

It is worth noting that, due to (27) and (28), it is $C(n) \leq G(n)$, and so $P(n) \geq 0$. From the definition of the SBBP process $G(n)$ and the joint process $(S^{(B)}(n), \underline{S}^{(D)}(n))$ defined so far, it follows that:

$$f_P(p) = \sum_{s'_{\Sigma} \in \mathfrak{S}^{(\Sigma)}} \pi_{[s'_{\Sigma}]}^{(\Sigma)} \sum_{g \in \mathfrak{S}^{(G)}} B_{[s'_{\Sigma}, g]}^{(G)} \cdot I_{s'_{\Sigma}, g}(p) \quad (21)$$

The term $I_{s'_{\Sigma}, g}(p)$ is a Boolean function indicating whether an amount of power equal to p is not used when the system state is s'_{Σ} and the WG has generated an amount of power equal to g . It can be derived considering that, during the time slot n , given the system state s'_{Σ} , the CS charges $\tilde{s}_B - s'_B$ non-mounted batteries

$$P^M = \begin{bmatrix} 6.00E-01 & 1.71E-01 & 1.14E-01 & 5.71E-02 & 9.52E-03 & 1.59E-02 & 1.59E-02 & 1.59E-02 \\ 8.57E-02 & 6.10E-01 & 1.05E-01 & 9.52E-02 & 5.71E-02 & 1.59E-02 & 1.59E-02 & 1.59E-02 \\ 2.86E-02 & 6.67E-02 & 6.76E-01 & 1.14E-01 & 6.67E-02 & 1.59E-02 & 1.59E-02 & 1.59E-02 \\ 1.90E-02 & 4.76E-02 & 9.52E-02 & 6.86E-01 & 1.05E-01 & 1.59E-02 & 1.59E-02 & 1.59E-02 \\ 9.52E-03 & 3.81E-02 & 7.62E-02 & 1.14E-01 & 7.14E-01 & 1.59E-02 & 1.59E-02 & 1.59E-02 \\ 2.86E-02 & 2.86E-02 & 2.86E-02 & 2.86E-02 & 2.86E-02 & 7.20E-01 & 8.57E-02 & 5.14E-02 \\ 2.86E-02 & 2.86E-02 & 2.86E-02 & 2.86E-02 & 2.86E-02 & 7.71E-02 & 6.86E-01 & 9.43E-02 \\ 2.86E-02 & 2.86E-02 & 2.86E-02 & 2.86E-02 & 2.86E-02 & 3.43E-02 & 8.57E-02 & 7.37E-01 \end{bmatrix}$$

Fig. 4. Transition probability matrix of the edge-computing service request process.

and $s'_E - \tilde{s}_E$ mounted batteries:

$$I_{s'_E, g}(p) = \begin{cases} 1 & \text{if } g - (\tilde{s}_B - s'_B) - (s'_E - \tilde{s}_E) = p \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

The mean value of the residual power, $E\{P\}$, will be applied in the numerical analysis to evaluate the mean power that can be used to supply other electrical loads.

V. SYSTEM SIMULATION

In this section, the proposed optimal FANET management has been applied in some different scenarios. The main aim is to show the performance achieved by RL in terms of the ability to satisfy the edge-computing service request, as well as how to exploit the variability of the green power availability.

These analyses are firstly performed in a specific scenario in Section V-A. Then, in Section V-B, other comparisons are presented by varying some parameters of the edge-computing service request process, but keeping its steady-state probability distribution.

A. Reference Scenario for Numerical Results

The reference scenario considered in this section is a post-disaster area whose edge-computing service requests are due to the need for computation to support rescue operations and the maintenance of a temporary field hospital built on site. These generate jobs each requiring a number of CPU operations with an average value $\gamma = 1.84 \cdot 10^6$. In order to process these jobs, it has been considered a FANET where each UAV is equipped with an edge-computing facility constituted by a Computer Processor Units (CPUs) Intel Core™ i7 Processor, 8 MB Cache, 2.7 GHz, and 32 GB DDR4 SO-DIMM ram. Therefore, the above CPU is able to process jobs with a rate $\bar{\mu}_P = 1.47$ kjob/s.

The edge-computing service request process is characterized by two main macro-states: a low-activity macro-state, with a job generation process ranging in the interval [13.2, 19.1] kjob/s, and a high-activity macro-state, with a job generation process ranging in the interval [26.5, 29.4] kjob/s. Quantizing the overall job-generation process with levels of $\bar{\mu}_P = 1.47$ kjob/s to express it in units of required UAVs, and applying the inverse-eigenvalue technique [42] to derive the SBBP modelling it, it is obtained an eight-state SBBP that assumes values in the set $\mathfrak{S}^{(M)} = \{9, 10, 11, 12, 13, 18, 19, 20\}$. Its transition probability matrix is shown in Fig. 4. Therefore, it has been considered a FANET constituted of 20 UAVs to be able to cope with the highest edge-computing service request.

A UAV provides edge-computing service during its flight. Therefore, the average time a UAV provides edge-computing services is equivalent to its average flight autonomy, whilst during its permanence in the CS, it does not provide any edge-computing service. To this purpose, in order to improve the service provided by the FANET, $B = 20$ additional batteries have been also considered for battery swapping, so that the overall number of batteries is 40. If a landed UAV finds a charged battery available in the CS, it is assumed that it immediately swaps the battery and takes off. Therefore, when there is an available full battery on the ground, the UAV provides the edge-computing service with no interruption, i.e., both in the time slot when it landed and in the next one since it immediately takes off, of course, if it is enabled by the FC to take off.

Considering that the UAV engines have a power consumption of about 510 W while the computing facility mounted onboard each UAV has an average power consumption of 35 W, and assuming that it uses a 48 V Lithium battery with a capacity of 34000 mAh, the flight autonomy of each UAV is about 3 hours, while the charging process is of 1 hour. Considering the use of highly efficient power converters, it is assumed an efficiency over 90%, thus the WG power to charge a battery is about 1.8 kW.

Starting from measurements of a horizontal axis direct-drive wind turbine, the available power from the WG has been modelled with 4-state SBBP whose intervals have been quantized in terms of UAV batteries that can be concurrently charged: [0–7], [5–12], [10–18], [16–25]. The number of CPs has been chosen equal to the highest number of batteries the WG can charge in a time slot, that is 25. Therefore, accounting for the power needed to charge a battery, the WG rated power is 45 kW.

According to the setting of the evaluation scenario described so far, the MDP is constituted by a number of states equal to 155232, and a total number of 1.190.112 actions, considering that in each state the set of possible actions ranges between 0 and the number of UAVs that are on ground ready for takeoff. In the considered case, the model solution, calculated using a computer with an Intel i7 4.7 GHz CPU and 16 GB Ram, was of about 2 hours. However, this is not a problem for a real-time application since, as said so far, the RL approach used in this paper is model based, and therefore solutions can be derived offline.

In order to evaluate the performance achieved with the optimal policy obtained by RL according to (2), it has been compared with other policies. In particular, the *greedy policy* has been considered, which is equivalent to leave always $a = 0$ UAVs on the ground. Similarly, the policies always considering a equal to, respectively, 1, 2 and 3 UAVs have been also analyzed. Finally, two random policies with, respectively, exponential and linear probability distributions associated with the value of a have been also considered: the lower the value of a , the greater the probability that a ready-for-takeoff UAVs are left on the ground. In the following figures, these strategies will be labeled as “Exp” and “Lin”.

Fig. 5 shows the k -level penalty for remote offloading, \wp_k , defined in (19), for $k \in \{1, 2, 3, 4\}$. It has been realized with some circles whose radius is inversely proportional to the penalty and normalized with respect to the best one. Obviously, the greater

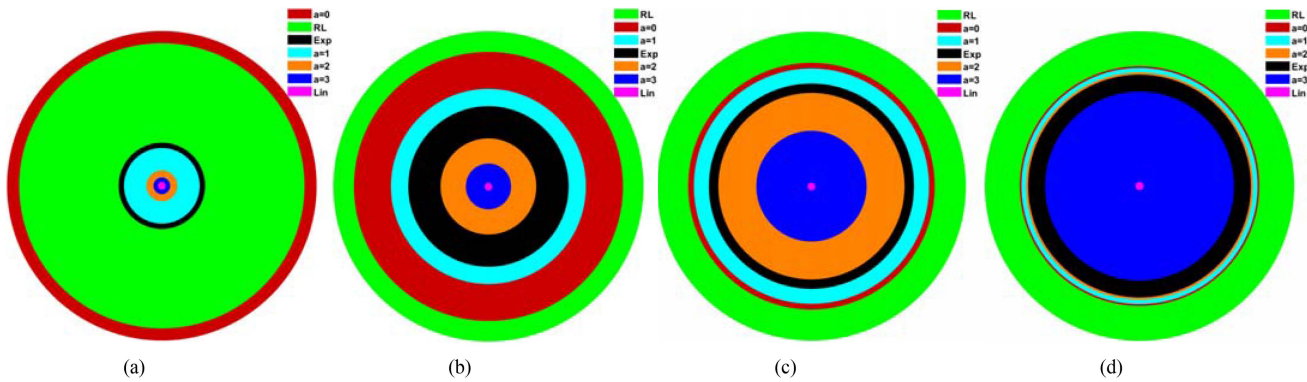


Fig. 5. Penalty for remote offloading. The radius is the inverse of the penalty computed using (18), normalized w.r.t. the greatest one (the greater the radius the better the performance of a policy). (a) $k = 1$. (b) $k = 2$. (c) $k = 3$. (d) $k = 4$.

the radius the better the performance of a policy. In Fig. 5(a) ($k = 1$), the greedy policy is the best one. This means that, in this case, there is no advantage in leaving $a = 0$ ready-for-takeoff UAVs in the CS to be used for future high-request events. This is because the penalty is linearly proportional to the difference, d , between the amount of edge-computing service required by the ground devices and the edge-computing service that the flying UAVs provide them. In this case, it is neglected to avoid high satellite usage in a small period, since no additional full batteries are left to face future hard situations. However, the best policy discovered by the RL provides also good results, very close to the greedy one.

Instead, starting from $k = 2$, the best policy discovered by the RL is the best one, as shown in Fig. 5(b)–(d). In fact, when $k = 2$ the RL policy is the best one since it has been found by using (2), which is equivalent to (19) for this k value. When $k > 2$, the RL policy also performs better than the other ones thanks to its ability to stoke ready-for-takeoff UAVs for periods in which there will be higher requests of edge-computing service and/or lower power availability. This fact is confirmed by the increasing ability of the RL policy to outperform the greedy one when k increases, as demonstrated by the radius reduction of the red circle ($a = 0$) when passing from Fig. 5(b) to Fig. 5(c) and, finally, to Fig. 5(d).

B. Scenarios With Different Edge-Computing Service Requests

This section analyzes the behaviour of the proposed strategy for different edge-computing scenarios. Moreover, it investigates the features of the RL policy enabling them to outperform the other policy. The study highlights that the use of different strategies the RL policy adopts under different available green power is the key of its success.

Three scenarios with lower and six scenarios with higher edge-computing service requests are here analyzed. More specifically, each new generic scenario is derived by adding a value x to the elements of $\mathfrak{S}^{(M)}$, representing the set of all the possible values of required UAVs to satisfy the time-variant edge-computing service requests coming from the devices on the ground in the reference scenario used in the above analysis. In other words, the generic scenario will be characterized by the

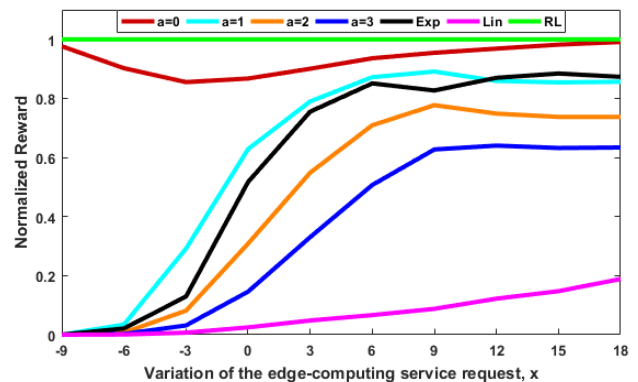


Fig. 6. Reward normalized with respect to the best one for different scenarios.

set of service requests here indicated as $\mathfrak{S}^{(M,x)}$ and defined as follows:

$$\mathfrak{S}^{(M,x)} = \left\{ s_M + x, \forall s_M \in \mathfrak{S}^{(M)} \right\} \quad (23)$$

More specifically, the first three scenarios are derived with $x \in \{-9, -6, -3\}$, while the other six scenarios are derived with $x \in \{3, 6, 9, 12, 15, 18\}$. In this way, for example, the lowest-load scenario, obtained with $x = -9$, is characterized by a job generation process ranging in the interval $[0, 5.9]$ kjob/s during the low-activity macro-state, and in the range $[13.2, 16.2]$ kjob/s in the high-activity macro-state. As for the reference scenario, the number of UAVs is always chosen equal to the highest edge-computing service request, i.e., $N = 20 + x$, to have a chance of cope also with this request. Consequently, the number of backup batteries has been set as $B = 20 - x$ in order to keep constant the overall number of batteries (as aforementioned equal to 40). All the other quantities of the reference scenario have been maintained. In each new scenario, the optimal policy has been found by using the RL.

Fig. 6 reports the rewards obtained as in (18), with the previously mentioned policies, normalized with respect to the one obtained by the RL. Obviously, the values at $x = 0$ are equal to the ones obtained in the previous section for $k = 2$ and reported in Fig. 5(b). The main outcome of Fig. 6 is that the performance of the greedy policy ($a = 0$) tends towards the

best one (the RL one) in the extreme scenarios (i.e., very low or high edge-computing requests). Instead, for intermediate values of x , the greedy policy is far from the optimal one; this means that, in the most of cases, it is more convenient to left one or more ready-for-takeoff UAVs in the CS although this implies not satisfying the immediate edge-computing request, to avoid a too poor quality of service in the future. The performance of the other policies is very poor when the edge-computing request is low (on the left side of the figure), and performance improves as the average request increases, although there is a great distance from the best policy, achieved by RL.

In order to better understand the reasons behind these results, the parameter named *Additionally stocked UAVs* (AS-UAVs) has been defined. The AS-UAVs are the UAVs that the FC leaves on the ground although they are ready for takeoff (i.e., their battery is charged), although they should be useful on air because the flying ones are not sufficient to provide the required service at that moment. However, the FC decides to maintain them on the ground for future takeoff when it forecasts a higher service request than the current one, which could deteriorate the immediate reward due to its ability to account for too poor quality service occurrences. Of course, when there are more ready-for-takeoff UAVs than the ones necessary to take off, the not necessary ones remain in the CS even when the greedy policy is applied, but these UAVs are not AS-UAVs. Therefore, it is useful to underline that the AS-UAVs are additional UAVs with respect to the number of UAVs that remain in the CS when the greedy policy is applied. Of course, when the UAVs ready-for-takeoff are less than the necessary ones, each ready-for-takeoff UAV forced to stay at the CS is an AS-UAV, since it is expressly stocked for future use.

It can be derived as follows:

$$\Gamma = \min \left\{ S^{(M)}(n) - S^{(A)}(n), S^{(R)}(n) \right\} - S^{(D)}(n) \quad (24)$$

Its expected value is computed as:

$$\begin{aligned} \bar{\Gamma} &= E \{ \Gamma(n) \} \\ &= \sum_{s'_{\Sigma} \in \mathfrak{S}(\Sigma)} \pi_{[s'_{\Sigma}]}^{(\Sigma)} \cdot [\min \{ s'_M - s'_A, s'_R \} - t'_D] \end{aligned} \quad (25)$$

where $S^{(M)}(n) - S^{(A)}(n) = s'_M - s'_A$ is the number of UAVs that are necessary on air to satisfy the service request, while $S^{(R)}(n) = s'_R$ is the total number of available UAVs on the ground, ready for takeoff. Instead, $S^{(D)}(n) = t'_D$ is the actual number of UAVs that will takeoff according to the FC decision, as defined in (12).

Fig. 7 shows the values of $\bar{\Gamma}$ computed as in (25). When the edge-computing request is low, the best policy operates similarly to the greedy one, as apparent from the magnification inside Fig. 7. The best policy behaves differently in very few cases for the scenario $x = -9$, but such a little difference is sufficient for obtaining an improvement. As x increases, the number of cases where the RL operates differently from the greedy one slightly increases (see the magnification of Fig. 7), but this little difference enables a good improvement in terms of reward (see Fig. 6). This is mainly because \wp_2 is close to 0 for the best policy, so that a little difference with the value reached by the

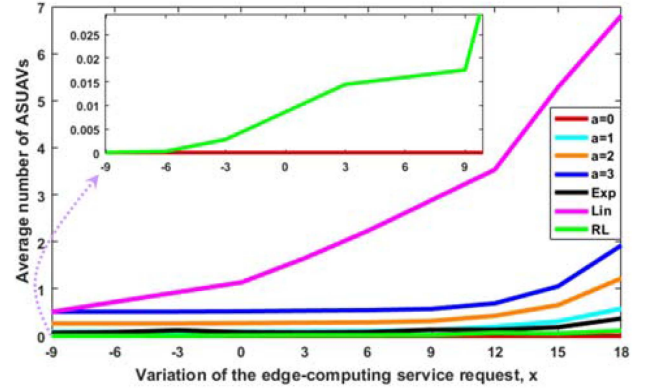


Fig. 7. Average number of ready UAVs that a policy left in the CS, although they should take off, for different scenarios. Magnif.: x -axis: $[-9, 10]$ and y -axis: $[0, 0.03]$.

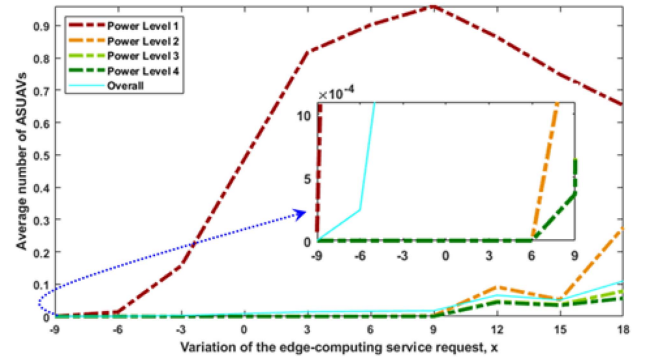


Fig. 8. Average number of ready UAVs left by the RL in the CS, although ready for take-off, for different scenarios and different levels of power available from the WG. Magnif.: x -axis: $[-9, 9]$ and y -axis: $[0, 10^{-3}]$.

greedy policy is sufficient to involve this good improvement. Fig. 7 shows that, in these scenarios, the other policies behave differently from the greedy and the optimal ones. Therefore, their performance is very poor with respect to the best one.

In the scenarios with very high edge computing requests, i.e., when $x > 9$, the difference between the behaviour of the best and greedy policies increases, although they remain quite similar on average. On the other hand, the number of UAVs is also increased, and it is more than three times greater than in the lowest scenario. Therefore, the difference between the number of UAVs that take off according to the best policy and the others is proportionally lower than in the previous scenarios. In other terms, although the difference between the best policy and the others increases in terms of the average number of AS-UAVs, the difference in terms of the overall behaviour decreases. Moreover, as the edge-computing service request increases, the performance of any policy worsens, thus the reward is more and more distant from 0 also for the best policy. The consequent conclusion is that, for high values of x , the normalized reward of the other policies increases, as shown in Fig. 6, although there is a great distance from the best policy.

Fig. 8 reports the same quantity as Fig. 7 only for the RL (that is, the curve labelled as “Overall” in Fig. 8 is the same

labelled “RL” in the previous figure). Moreover, this quantity is also reported by grouping the results according to the states characterized by the same level of power generated by the WG. For example, the red curve, labelled as “Power Level 1”, represents the average number of AS-UAVs when the WG is in the lowest power generation state. These curves reveal some interesting aspects concerning the choices performed by the best policy in relation to the power available from the green generator. When the power available is low, the best policy keeps some AS-UAVs, to cope with more challenging future events. As apparent from the magnification in Fig. 8, the small difference revealed so far between the best and the greedy policies is mainly due to their behaviour during the periods of very low green power availability. In fact, for all the other scenarios with higher power levels, the average number of AS-UAVs is zero likewise for the greedy policy when the average edge-computing request is not high. Such behavior is mainly due to the combination of two main reasons. In such scenarios, being the average edge-computing request is not too much (i.e., x small or negative) there is a higher number of available batteries in the CS since $B = 20 - x$, and the high power availability enables to keep a large number of full batteries on the ground. Therefore, it is very likely that a landing UAV finds a full battery, so it is immediately ready to take off. According to this reasoning, it is not necessary to leave UAVs in the CS for future challenging events since these events will be easily faced thanks to the large amount of full batteries.

This strategy cannot be used during the lowest power generation level since it is less likely that there is a power surplus to charge the batteries on the ground since almost the whole power is used to charge the batteries of the landed UAVs. Moreover, in the case of the lowest power generation level, the average number of AS-UAVs increases as the average request increases in the various scenarios. Indeed, another interesting aspect that arises from the figure analysis is that, when the average edge-computing service is high ($x > 9$), such an increment does not continue further. Actually, the average number of AS-UAVs decreases as the average request increases in the various scenarios. In these cases, where there is a large edge-computing request but with the lowest power generation availability (Power Level 1), it can be considered that all the events are challenging, and then it becomes more difficult to stock AS-UAVs. On the contrary, in these scenarios with a high service request, during the periods with the other power levels, an increment in the average request involves that the average number of AS-UAVs increases. To discuss the hypothesis about the possible causes behind this behaviour, it is necessary to keep in mind the previous reasoning about the advantages deriving from the full batteries on the ground useful to avoid keeping the UAVs in the CS. In the scenarios with a high service request, the number of batteries in the CS is low regardless they are empty or not. Moreover, there are more mounted batteries to charge due to the greater number of UAVs ($N = 20 + x$) and due to the need for more UAVs on air. These aspects have greater importance in the case of a medium/low level of available power (Power Level 2). More specifically, in this case, it is very limited the number of batteries that can be charged in addition to the UAV ones,

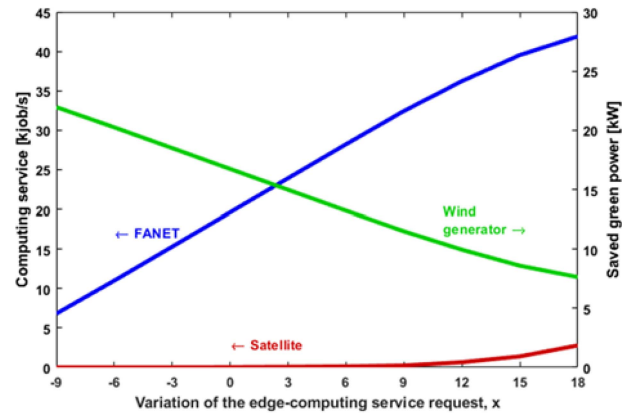


Fig. 9. Average edge-computing service provided by FANET and Satellite when the RL policy is adopted for different scenarios. Average WG power that can be delivered to the local load.

thus it is necessary to stock some additional UAVs for facing future more challenging events. Since the possibility to charge some batteries in the CS increases as the level of available green power increases, the number of AS-UAVs reduces as the level of power increases (Power Level 3 under Power Level 2, Power Level 4 under Power Level 3).

Finally, Fig. 9 shows both the provided average edge-computing service and the remote computing request when the FANET is managed by using RL policy in the various scenarios and the related average saved WG power. The quantities are almost linearly dependent on the average edge-computing service request until $x = 12$, after they slightly initiate to tend towards constant values. An important aspect is that the slope of the curve is about equal to 1 until the scenario $x = 12$. This means that the increment of the average service request is satisfied by an increment of the same magnitude in terms of provided service. This important result is obtained by only increasing the number of UAVs (although the number of batteries is kept constant) and by finding the best policy in each scenario.

VI. CONCLUSION AND FUTURE WORKS

This paper proposes to use artificial intelligence for automatic management of a FANET providing edge-computing in post-disaster scenarios. A CS for battery charging with a WG has been also considered. The FC applies a model-based RL to decide how many UAVs have to take off taking into account the current power generation availability and the edge-computing service requests, and a forecast of them. An additional novelty is the discrete-time analytical model of the system defined to provide the FC with a Markov Decision Process, in order to support its decisions.

It is worth stressing that the proposed RL approach does not suffer from convergence problems because no online training is needed. Indeed, the fact that an exhaustive model of the environment has been defined and that the number of UAVs composing a FANET is usually not so huge, makes it possible to find the optimal policy in closed form offline by means of the Bellman equation system.

The optimal management policy dynamically adapts its behaviour to avoid wide use of backup satellite channels in a short-time horizon during low green-energy generation and high service request periods. The optimal policy is an efficient modification of the greedy one with a level of variation that depends on the combination of the service request and the green power availability. More specifically, their behaviour mainly differs during the periods of low green power availability, since, in this case, the optimal policy frequently stocks UAVs for facing future edge-computing requests although these UAVs would be necessary to satisfy the current request. The comparisons with other policies have demonstrated the achieved gain, proving that the proposed framework based on a FANET supplied by green generation and the proposed management strategy are suitable to face emergence scenarios that need computing resources but lack connection with the power grid and the core network infrastructure.

In future works, the FANET management framework proposed in this paper to support the paradigm of Network Function Virtualization, could include the placement of service chains in the FANET. This is done by taking into account the state of charge of the battery of each flying UAV, in order to maximize the amount of traffic flows that the FANET is able to manage.

APPENDIX

TRANSITION PROBABILITY MATRIX OF THE PROCESS ($S^{(B)}(n), \underline{S}^{(D)}(n)$)

In order to derive this last matrix, it is necessary to describe the behaviour of UAVs and batteries in detail during the time slot. As represented in Fig. 3, the state evolution during the time slot has been divided into two phases. First, the intermediate (point 3 of the previous list) values of the UAV and battery charge states, indicated with \tilde{s}_D and \tilde{s}_B , respectively, have been evaluated; then, the final state s''_{Σ} . Has been calculated

With the aim of calculating the *intermediate state*, it is considered that the intermediate number of flying UAVs is increased by the taken-off UAVs, t'_D , as in (12), that is:

$$\tilde{s}_A = s'_A + t'_D \quad (26)$$

Actually, this is also the number of flying UAVs at the *beginning state* (point 2), differently from the other intermediate quantities computed in the following.

The number of UAVs that can be potentially charged by WG in the time slot n is said g .

The intermediate number of UAVs with an empty battery, \tilde{s}_E , is the initial number, s'_E , decreased by the number of UAVs that have been charged by means of the available supplied CPs, $\min\{g, M\}$, and which so move to the state of UAVs ready for takeoff:

$$\tilde{s}_E = \max\{s'_E - \min\{g, C_P\}, 0\} \quad (27)$$

The remaining supplied CPs, i.e., the ones that have not been used for UAVs, are used to charge some empty batteries. In this case, the number of charged batteries could increase by a quantity complementary to (27) but limited by the number of batteries, L . Otherwise, the number of charged batteries remains

the same. Therefore, the updated number of charged batteries is:

$$\tilde{s}_B = \begin{cases} \min\{s'_B + [\min\{g, C_P\} - s'_E], L\} & \text{if } \tilde{s}_E = 0 \\ s'_B & \text{otherwise} \end{cases} \quad (28)$$

Instead, the intermediate number of UAVs that are ready to takeoff is equal to the initial number, decreased by the taken-off UAVs, t'_D , and increased by the number of UAVs that are charged, $s'_E - \tilde{s}_E$:

$$\tilde{s}_R = s'_R - t'_D + s'_E - \tilde{s}_E \quad (29)$$

Now, the *final state* (point 6 of the previous list), s''_{Σ} , is derived by updating the intermediate states. It is done by considering the transition of $S^{(M)}(n)$ and $S^{(G)}(n)$, and the number of landing UAVs. Being this last number defined ℓ , the final number of UAVs on air is:

$$s''_A = \tilde{s}_A - \ell \quad (30)$$

Since some landed UAVs could swap their battery with a charged one thus becoming immediately ready to take off, the number of UAVs that are ready for takeoff and the number of remaining charged batteries is:

$$s''_R = \tilde{s}_R + \min\{\ell, \tilde{s}_B\} \text{ and } s''_B = \max\{\tilde{s}_B - \ell, 0\} \quad (31)$$

At the same time, the number of UAVs waiting for charging is increased by the landed UAVs that do not find a charged battery, that is:

$$s''_E = \tilde{s}_E + \max\{\ell - \tilde{s}_B, 0\} \quad (32)$$

So, with all this in mind, the transition probability matrix of the joint process ($S^{(B)}(n), \underline{S}^{(D)}(n)$) can be computed. To this purpose, the total probability theorem is applied to g , representing the number of CPs that can be supplied simultaneously by the renewable generator among the M CPs that are available in the charge station, and on the number ℓ , representing the number of UAVs that land at the end of the time slot n among the \tilde{s}_A that are flying:

$$\begin{aligned} P_{[(s'_B, \underline{s}'_D), (s''_B, \underline{s}''_D)]}^{(B,D)}(s'_G, a) \\ = \sum_{g \in \mathbb{S}^{(G)}} B_{[s'_G, g]}^{(G)} \cdot \sum_{\ell=0}^{s'_A} p_{[\ell]}^{(Land)}(\tilde{s}_A) \cdot \\ \begin{cases} 1 & \text{if } s''_A = \tilde{s}_A - \ell \text{ and } s''_B = \max\{\tilde{s}_B - \ell, 0\} \\ & \text{and } s''_R = \tilde{s}_R + \min\{\ell, \tilde{s}_B\} \\ & \text{and } s''_E = \tilde{s}_E + \max\{\ell - \tilde{s}_B, 0\} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (33)$$

The term $B_{[s'_G, g]}^{(G)}$ represents the probability that g CPs can be supplied when $S^{(G)}(n) = s'_G$. The term $p_{[\ell]}^{(Land)}(\tilde{s}_A)$ is the probability that ℓ UAVs, among \tilde{s}_A UAVs that are flying, land in a time slot because needing to be charged. In order to calculate it, since the UAV battery SoCs are independent of each other, the landing probability for a UAV in a time slot is modelled as a Bernoulli process with probability ρ_{Down} . This parameter depends on the average time on air. Assuming that in the time slot when a UAV takes off does not land, the per-slot landing probability starting from the second time slot on air is $\rho_{Down} = 1/(H - 1)$. Therefore, considering that \tilde{s}_A represents

the number of UAVs that are already on air at the beginning of the time slot (*beginning state*), they potentially may land with probability $p_{[\ell]}^{(Land)}(\tilde{s}_A)$ following a Binomial distribution:

$$p_{[\ell]}^{(Land)}(\tilde{s}_A) = \binom{\tilde{s}_A}{\ell} \cdot (\rho_{Down})^\ell \cdot (1 - \rho_{Down})^{\tilde{s}_A - \ell} \quad (34)$$

REFERENCES

- [1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.
- [2] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.
- [3] N. Zhao et al., "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.
- [4] J. Zhao, F. Gao, Q. Wu, S. Jin, Y. Wu, and W. Jia, "Beam tracking for UAV mounted SatCom on-the-move with massive antenna array," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 363–375, Feb. 2018.
- [5] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.
- [6] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.
- [7] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-Enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, Mar. 2017.
- [8] J. Xu, Y. Zeng, and R. Zhang, "UAV-Enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.
- [9] F. Cheng et al., "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.
- [10] Y. Chen, N. Zhao, Z. Ding, and M.-S. Alouini, "Multiple UAVs as relays: Multi-hop single link versus multiple dual-hop links," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6348–6359, Sep. 2018.
- [11] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-Enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, Mar. 2017.
- [12] E. Christy, R. P. Astuti, B. Syihabuddin, B. Narottama, O. Rhesa, and F. Rachmawati, "Optimal UAV flying path for device-to-device communications in disaster area," in *Proc. Int. Conf. Signals Syst.*, 2017, pp. 318–322.
- [13] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, "A UAV-cloud system for disaster sensing applications," in *Proc. 81st Veh. Technol. Conf.*, 2015, pp. 1–5.
- [14] G. Tuna, B. Nefzi, and G. Conte, "Unmanned aerial vehicle-aided communications system for disaster recovery," *J. Netw. Comput. Appl.*, vol. 41, no. 1, pp. 27–36, 2014.
- [15] R. Siegwart et al., "Legged and flying robots for disaster response," in *Proc. World Eng. Conf. Conv.*, 2015, pp. 1–8.
- [16] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," *IEEE Wireless Commun. Netw. Conf. Workshops*, 2015, pp. 329–334.
- [17] N. Ahmed, S. S. Kanhere, and S. Jha, "On the importance of link characterization for aerial wireless sensor networks," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 52–57, May 2016.
- [18] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Can. J. Forest Res.*, vol. 45, no. 7, pp. 783–792, 2015.
- [19] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, Apr.–Jun. 2016.
- [20] M. Erdelj, M. Krøl, and E. Natalizio, "Wireless sensor networks and multi-UAV systems for natural disaster management," *Comput. Netw.*, vol. 124, pp. 72–86, 2017.
- [21] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan.–Mar. 2017.
- [22] X. Li, H. Yao, J. Wang, S. Wu, C. Jiang, and Y. Qian, "Rechargeable Multi-UAV aided seamless coverage for QoS-guaranteed IoT networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10902–10914, Dec. 2019.
- [23] X. Li, H. Yao, J. Wang, X. Xu, C. Jiang, and L. Hanzo, "A near-optimal UAV-aided radio coverage strategy for dense urban areas," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9098–9109, Sep. 2019.
- [24] G. Faraci, C. Grasso, and G. Schembra, "Design of a 5G network slice extension with MEC UAVs managed with reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2356–2371, Oct. 2020.
- [25] G. Faraci, C. Grasso, and G. Schembra, "Fog in the clouds: UAVs to provide edge computing to IoT devices," *ACM Trans. Internet Technol.*, vol. 20, no. 3, Aug. 2020, Art. no. 26.
- [26] C. Grasso, R. Raftopoulos, and G. Schembra, "Deep Q-Learning for job offloading orchestration in a fleet of MEC UAVs in 5G environments," in *Proc. IEEE 7th Int. Conf. Netw. Softwarization*, 2021, pp. 186–190.
- [27] F. D'Ursol, C. Grasso, C. Santoro, F. F. Santoro, and G. Schembra, "The tactile internet for the flight control of UAV flocks," in *Proc. IEEE 4th Conf. Netw. Softwarization Workshops*, 2018, pp. 470–475.
- [28] C. Rametta and G. Schembra, "Designing a softwarized network deployed on a fleet of drones for rural zone monitoring," *Future Internet*, vol. 9, no. 1, Mar. 2017, Art. no. 8.
- [29] E. P. F. da Cruz, "A comprehensive survey in towards to future FANETs," *IEEE Latin Amer. Trans.*, vol. 16, no. 3, pp. 876–884, Mar. 2018.
- [30] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.
- [31] T. Dapper e Silva, C. F. Emygdio de Melo, P. Cumino, D. Rosário, E. Cerqueira, and E. Pignaton de Freitas, "STFANET: SDN-Based topology management for flying ad hoc network," *IEEE Access*, vol. 7, pp. 173499–173514, 2019.
- [32] D.-Y. Kim and J.-W. Lee, "Integrated topology management in flying ad hoc networks: Topology construction and adjustment," *IEEE Access*, vol. 6, pp. 61196–61211, 2018.
- [33] Z. Zhao et al., "Software-defined unmanned aerial vehicles networking for video dissemination services," *Wireless Hoc Netw.*, vol. 83, pp. 68–77, Feb. 2019.
- [34] "Mobile power station - The world's first 10kW portable wind turbine," Accessed: Nov. 3, 2022. [Online]. Available: <https://upriseenergy.com/>
- [35] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.
- [36] S. Davis, "Military operations command more from mobile power sources," *Power Elect. Technol.*, vol. 36, no. 3, 2010, pp. 1–9.
- [37] T. Long, M. Ozger, O. Cetinkaya, and O. B. Akan, "Energy neutral internet of drones," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 22–28, Jan. 2018.
- [38] D. Lee, J. Zhou, and W. T. Lin, "Autonomous battery swapping system for quadcopter," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2015, pp. 118–124.
- [39] S. Conti, G. Faraci, R. Nicolosi, S. A. Rizzo, and G. Schembra, "Battery management in a green fog-computing node: A reinforcement-learning approach," *IEEE Access*, vol. 5, pp. 21126–21138, 2017.
- [40] S. T. Muntaha, S. A. Hassan, H. Jung, and M. S. Hossain, "Energy efficiency and hover time optimization in UAV-Based HetNets," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5103–5111, Aug. 2021.
- [41] X. I. E. Chenxiao and X.-L. Huang, "Energy-efficiency maximization for fixed-wing UAV-enabled relay network with circular trajectory," *Chin. J. Aeronaut.*, vol. 35, no. 9, pp. 71–80, 2022.
- [42] A. Lombardo, G. Morabito, and G. Schembra, "Modeling intramedia and intermedia relationships in multimedia network analysis through multiple Time-scale statistics," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 142–157, Feb. 2004.
- [43] R. Bellman, "On the theory of dynamic programming," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 38, no. 8, pp. 716–719, Aug. 1952.