



MDP-Based Resource Allocation Scheme Towards a Vehicular Fog Computing with Energy Constraints

Habtamu Birhanie, Messous Mohamed-Ayoub, Sidi Mohammed Senouci,
El-Hassane Aglzim, Ahmedin Mohammed

► To cite this version:

Habtamu Birhanie, Messous Mohamed-Ayoub, Sidi Mohammed Senouci, El-Hassane Aglzim, Ahmedin Mohammed. MDP-Based Resource Allocation Scheme Towards a Vehicular Fog Computing with Energy Constraints. Futurmob 2018, Jun 2018, Nevers, France. 10.1109/GLOCOM.2018.8648081 . hal-03327528

HAL Id: hal-03327528

<https://hal.science/hal-03327528>

Submitted on 13 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

MDP-based Resource Allocation Scheme towards a Vehicular Fog Computing with Energy Constraints

Habtmu Mohammed Birhanie^{1,2}, Mohammed Ayoub Messous¹, Sidi-Mohammed Senouci¹, El-Hassane Aglzim¹, Ahmedin Mohammed Ahmed²

¹DRIVE EA 1859, Univ. Bourgogne Franche Comté, F58000, Nevers, France.

habtmu_birhanie@etu.u-bourgogne.fr; {ayoub.messous, Sidi-Mohammed.Senouci, el-hassane.aglzim}@u-bourgogne.fr

²Kombolcha Institute of Technology, Wollo University, Kombolcha, Ethiopia

a.m.1985@ieee.org

Abstract— As mobile applications deliver increasingly complex functionalities, the demands for even more intensive computation would quickly transcend energy capability of mobile devices. On one hand and in an attempt to address such issues, fog computing paradigm is introduced to mitigate the limited energy and computation resources available within constrained mobile devices, by moving computation resources closer to their users at the edge of the access network. On another hand, most of electric vehicles (EVs), with increasing computation, storage and energy capabilities, spend more than 90% of time on parking lots. In this paper, we conceive the basic idea of using the underutilized computation resources of parked EVs as fog nodes in order to provide on-demand computation at the vicinity of the access network. The proposed Vehicular Fog Computing (VFC) architecture aggregates the abundant unused resources of parked vehicles, and uses it to serve mobile users' demands. The resource allocation problem is formulated as a Markov Decision Process (MDP) and dynamic programming is used to solve the underlying decision problem. Extensive simulation results show the effectiveness of the proposed approach by improving the global reward value by 51% and scoring an energy gain of 66% compared to two other models.

Keywords— Vehicular Fog Computing, Markov Decision Process, Electric Vehicle, Energy.

I. INTRODUCTION

Nowadays, the widespread of mobile devices and wireless communications has led to higher demands for data exchange and computing requirements [1]. However, mobile devices have usually limited capabilities in terms of computation power, battery lifetime, storage space and even available bandwidth. In order to address these limitations and to continue supporting the ever-increasing applications demands, service providers are usually willing to make use of powerful servers offered through the cloud. Therefore, cloud computing is considered as a promising solution to deliver services to end-users and provides locally-deployed applications with elastic resources at a lower cost [2]. However, cloud computing does not solve all the problems due to its own drawbacks, which are for instance related primarily to latency and quality of service (QoS) issues for real-time applications. Thus, the recent development is to push the storage and processing capabilities to the edge of the access network closer to their end users, which introduces the new concept of fog computing. Fog computing is a

decentralized computation framework which essentially extends cloud computing resources and services to the edge of the network, by bringing the advantages and power of the cloud servers closer to where requests are created [2].

Many recently body of research focuses on introducing vehicles to the fog computing paradigm. They mostly proposed integrating the resources available locally within vehicles as part of the fog computing infrastructure. The resulting framework would constitute the new paradigm called Vehicular Fog Computing. Instead of sending users' computation requests to remote cloud servers, VFC handles the nearby users' demands using locally available vehicular resources [3]. Furthermore, VFC distinguishes itself from other existing techniques with its proximity to end-users, dense geographical distribution and mobility support [3]. Besides, it focuses on the usage of vehicles as a service delivery to improve communication [4].

Nowadays, the number of vehicles roaming the streets is ever increasing. Besides networking and communication interfaces, most recent vehicles may be equipped with fairly high computation and storage capabilities. Moreover, the technological advances in automotive sector and the battery storage capabilities have led to the proliferation and the wide spread of EVs, especially in city centers. The vast majority of EVs are equipped with pretty powerful processing units, large data storage and decent communication capabilities. However, according to a survey [5], more than 90% of vehicles travel is less than 100 miles daily, which on average represents a 34 KWh of energy consumed for a Tesla Model S whose battery capacity is 85 KWh with ~40% energy consumption. Based on another statistic [5], vehicles spend only between 4-5% of the day traveling while they spend the rest of the time in parking lots. Therefore, with the right incentive mechanisms for the owners of the parked EVs, the latter may be used to deliver different services. One of the most promising potential ideas is to use EVs as computing devices for nearby mobile users. This will make a better usage of the underused available resources in electric vehicles for computation services.

In the light of such insight, the current work suggests using electric vehicles energy for locally processing mobile application computation requests. Particularly, we consider the parked electric vehicles as fog nodes that would offer computing services to serve nearby users. Here, we address

the problem of resource allocation for mobile applications with highly intensive computation tasks. The abundant unused resources of parked EVs are aggregated in order to serve the computation and storage demands of these mobile devices particularly constrained in terms of energy and processing capabilities. A Markov Decision Process is adopted to model this resource allocation problem and solve the underlying optimization problem using a dynamic programming.

The reminder of this paper is organized as follows: in Section II, we present related work. The system architecture is presented in Section III and both the problem and the corresponding solution are described in Section IV. The most prominent simulation results are summarized and discussed in Section V. Finally, concluding remarks and future research directions are highlighted in Section VI.

II. RELATED WORKS

Several previous works have been achieved with as main goal to serve mobile applications demands in terms of intensive computation. Most of the works have been conducted about the decision making for offloading computation to the cloud, cloudlet, VCC (Vehicular Cloud Computing), or VFC. Zheng et al. [6] proposed a resource allocation scheme to maximize the reward of the vehicular cloud. Their approach is based on infinite horizon Semi Markov Decision Process (SMDP). The proposed scheme uses four stages to formulate the resource allocation problem: (i) state space - represents the current resources and request states in a vehicular cloud; (ii) action space - is a set of actions that can be used by vehicular cloud based on the current state; (iii) reward - consists of the income and cost that allow to use a discount model that computes and analyzes the sum of the rewards; (iv) transition probability - calculates the probability of a state of the system to change to another state under a specific action. The aim was to maximize the long-term expected total reward of the vehicular cloud.

The authors in [9] proposed to use parked vehicles to improve communication in a vehicular ad hoc network (VANETs) due to the sparsity of the network and the unbalanced nature of its traffic. The idea is to use parked vehicles as a static backbone infrastructure to serve and improve the VANET communication. Likewise, authors in [8] investigated network connectivity between parked vehicles through theoretic analysis and realistic survey and simulation. They also envisage new services and application based on resources sharing among moving and static vehicles.

In [3], the authors investigated the potential of using vehicular networks as an infrastructure for communication and computation. Furthermore, they considered both moving and parked vehicles to deliver communication and computation resources in order to serve requests from nearby end users. The proposed VFC works by aggregating abandoned resources of individual vehicles. Their results showed an improvement in the computational performance, compared to conventional systems, due to making the best



Fig. 1: Vehicular Fog Computing

use of under-utilized computational resources of individual vehicles.

With the transition to hybrid and fully electric transportation, in terms of battery power, vehicles are becoming uniquely positioned among mobile devices. As energy represents the main critical resource in a constrained mobile environment, we consider that higher intensive computation tasks can be offloaded from mobile devices to parked electric vehicles. Therefore, in this paper we propose a novel system of VFC that employs electric vehicles as a fog node for computation to serve local mobile device application demands. To our best knowledge, we are the first to consider EVs energy to serve local mobile users application demands.

III. SYSTEM ARCHITECTURE

We consider for this study the synoptic architecture for a Vehicular Fog Computing as shown in Fig 1. Beside the central cloud, the proposed fog computing scheme comprises three different components: the fog controller, a set of parked EVs considered as fog nodes and finally mobile devices. The fog controller, which is deployed by service providers, offers wireless access (Cellular or Wi-Fi) for mobile devices. It also affects computation demands to the nearby parked EVs (within its coverage range).

Upon the arrival of energy intensive mobile application computation requests, the fog controller decides whether to run it at the VFC or redirect the requests to cloud according to the availability of energy within parked EVs. Cloud computing is used whenever there is no sufficient resources available to serve users applications demands using the parked EVs.

Mobile devices have become very popular and run different kind of energy intensive applications. However, due to the device size, its computation, storage and energy capabilities are limited [7]. To get good application execution and to save energy for mobile devices, computation offloading is a promising option. In the proposed model, we proposed to use part of the unused resources of parked EVs by nearby mobile devices for computation while making sure to not exhaust all the EVs energy.

We consider that electric vehicles parked at parking lot are considered as fog servers to use, for computation, vehicles as an infrastructure represented by the aggregation of the abundant resources of individual electric vehicles. VFC offers this resource pool containing a certain level of energy/computation in order to compute highly intensive computation tasks. In the proposed system, each mobile application demand and electric vehicle arrival follows a Poisson process. In addition, electric vehicles' departure rate follows an exponential distribution. In this paper, we assume that there is no charging mechanism in the parking lot. Moreover, available energy in the parking lot depends mainly on the parking lot occupancy level and state of charge of the parked EVs. If higher energy level becomes available, additional computation requests from mobile device would be served. Therefore, the main objective in this paper is to optimize the state of charge for parked EVs in order to better serve the demands of mobile applications. The underlying optimization problem is formulated through a MDP, where the details are provided in the following section.

IV. MDP BASED RESOURCE ALLOCATION

The idea of the resource allocation problem is to find an optimal allocation of a fixed amount of resources using optimization techniques. In this work, the aim is to optimally use the available computation resources of parked EVs to serve local mobile applications demands. To resolve this issue, we formulate the problem as a Markov Decision Process. In the following, we recall the MDP concept before presenting the problem formulation and solution.

A. Markov Decision Processes

Markov decision process (MDP) constitutes a mathematical framework for dynamic controlling systems that evolves stochastically [11]. MDPs are used for modeling decision making. The systems resolved by MDP are observed at times $t=1, 2, \dots, n$ where n is called the time horizon. The decision maker takes a decision regarding the action to be done at each time. The target of the decision maker is to choose the actions that optimize the performance of the system.

A MDP consists of a set of 5 elements $\langle S, A, P, R, \gamma \rangle$ as described below:

- S : is a finite set of the system states,
- A : is a finite set of actions to be chosen by the decision maker,
- P : is the transition probability; thus, $P(S; S'; a)$ is the probability of moving from state S to state S' if action a is chosen,
- R : is the immediate reward (or expected) received after moving from state S to state S' via action a ,
- γ : is the discount factor which satisfies $\gamma \in [0, 1]$. It represents the difference between R_{t+1} and R_t .

MDPs aim is to specify the mechanism by which the decision maker chooses actions at different times. This mechanism is called Policy $\pi(S)$. The policy is a mapping

from states S to actions A . It defines which action should be taken in each state. Thus, MDP aims to choose a policy that will maximize a cumulative function of the random rewards:

$$\sum_{t=0}^{\infty} \gamma^t R(S_t; S_{t+1}, (S_t))$$

Several techniques are available in the literature to solve MDPs problems: Linear Programming (LP) and Dynamic Programming (DP) [11]. In this work, we choose to use dynamic programming where the model and simulation assumptions are presented in the following sections.

B. Problem Formulation and Solution

As indicated in the previous subsections, we formulate the problem as an MDP with the following five elements $\langle S, A, P, R, \gamma \rangle$:

- States $S = S_0; S_1; S_2, \dots, S_n$; refers to the different energy status of the EVs:
 - S_0 : Full Energy,
 - S_n : Empty Energy,
 - S_i : Energy level with load level = $i\%$, $i \in \{1; 2; 3; \dots; n-1\}$.
- Actions $A = a_0; a_1; a_2, \dots, a_n$; where a_i refers to the action to be selected according to the transition probabilities:
 - a_0 : represents the action taken when no resources are available for computation,
 - a_i : use the energy for computation ($i\%$ of energy level for computation), $i \in \{1; 2; 3; \dots; n-1\}$.
- Transition probability P : $P(S; S'; a)$ is the probability of moving from the energy level S at $t = n$ to the energy level S' at $t = n + 1$ by doing the action a . $P(S, S', a)$ is determined by the arrival and departure of EVs with a certain SoC (State of Charge). Hence, $P(S, S', a)$ is related to the energy consumption of users computation demand and the arrival of EVs in the next times $t = n+1, n+2, \dots$, etc. The availability of energy for computation for next time $t = n+1, t = n+2, \dots$ is related to the arrival of new EVs in the parking area. Fig. 2 show synoptic MDP graph with five states. It also shows the different transition probabilities between different states with action a_0 . Each state represented aggregated energy level of parked EVs.
- Rewards, R : is the reward collected by choosing different actions. In our study, we represent R as a positive integer $R \in \{0, 1, 2, 3, \dots, n\}$, where $R=0$ means the fog controller do not assign users computation demand to EVs.
- γ : the discount factor, $\gamma \in [0; 1]$.

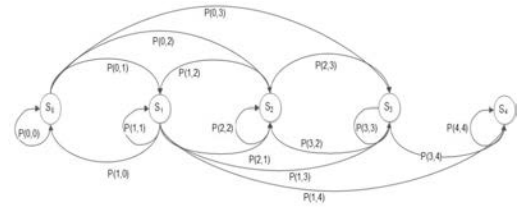


Fig 2. MDP state diagram (example for action a_0)

- The controller: represents the fog controller, presented in the previous sections, which is responsible of aggregating and managing the resources of nearby EVs. It also takes decisions regarding the allocation the available energy. The fog controller is installed by service provider at the parking lots.

The MDP model aims to find the policy that specifies the best policy $\pi(s)$ that the controller has to choose regarding the use of the surplus power in the parking lot. The goal is to maximize the rewards by using the energy of EVs for computation without affecting the energy used by EVs for motion. The objective function to be maximized is defined in equation (1):

$$\text{Maximize} \left(\sum_{t=0}^{\infty} \gamma^t R(S_t, S_{t+1}, \gamma(S_t)) \right) \quad (1)$$

Subject to:

$$\begin{aligned} \sum_{t=0}^{\infty} R((S = S_n)_t, S'_{t+1}, a_t) &= 0 \\ \forall S' \in \{S_1, S_2, S_3, \dots, S_n\}, \forall a_i \in \{a_1, a_2, a_3, \dots, a_n\} \end{aligned}$$

The algorithm uses the following two equations, (2) and (3), to calculate the optimal policy for all the states S until no further changes take place (according to the discount factor):

$$\pi(S) = \arg \max \{ \sum_{S'} P_a(S, S') (R_a(S, S') + \gamma V(S')) \} \quad (2)$$

$$V(S) = \sum_{S'} P_{\pi(S)}(S, S') (R_{\pi(S)}(S, S') + \gamma V(S')) \quad (3)$$

Finally, the algorithm generates a stationary policy that takes the form $\pi : S \rightarrow A$ with $\pi(S)$ denoting the action to be executed in state S . The detailed description of the algorithm is presented hereafter.

C. MDP-based decision making algorithm

The decision-making approach we proposed is described through sequence diagram in Fig. 3. In order to make a decision to accept or reject computation request of mobile users, the controller adopts first come first serve scheduling discipline. Up on arrival of EVs into the parking lot, they sent their energy status information to fog controller. In addition, fog controller receives users' requests from mobile devices. Once the energy status of EVs and mobile applications energy demands received, an action that scores maximum reward is selected. Then, the tasks will be admitted into fog nodes. However, if there is no energy to serve the users demand in the parking, the controller rejects the users request and their computation request will be redirected to cloud. To evaluate the performance of the proposed algorithm, we did simulation with different cases. The performance evaluation is presented in the following section.

V. EXPERIMENTAL EVALUATION AND ANALYSIS

As we presented in section IV, the state space is modeled with the number of each parked EVs energy level. However, considering each parked EVs in the parking as a state will create larger and infinite state space. In order to simplify simulation process, we used an aggregated state space. The idea behind is that the resulting reduced state space forms small stationary finite state MDP, which can then be efficiently solved. Hence, we identified five possible different states to represent the occupancy rates, namely $S = \{Full,$

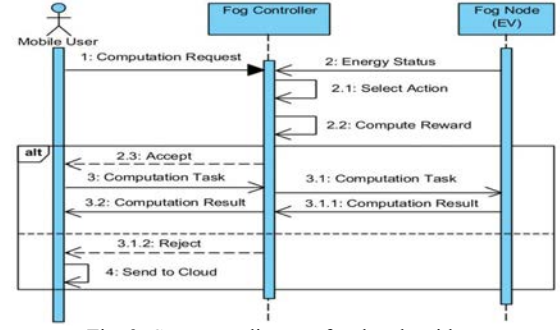


Fig. 3: Sequence diagram for the algorithm

High, Medium, Low, Empty which represent the overall available energy in the parking: up to 100%, up to 75%, up to 50%, up to 25 and $\approx 0\%$ respectively. In addition, we also define four possible actions, namely $A = \{a_0, a_1, a_2, a_3\}$, which represent the action taken at each energy level. Thus, action a_0 is taken when there is 0% up to 25% energy is available in the parking. An action a_1 is selected when the available energy level is from 25% up to 50%, a_2 selected when energy status is from 50% up to 75% and action a_3 is selected when the available energy is greater than 75%.

Furthermore, for each action a_i , we defined its corresponding transition probability matrix $P(S, S', a_i)$. The matrix is an n-by-n matrix that comprises all the probabilities for moving from energy status S at time t to energy status S' at time $t+1$ using action a_i . Finally, we use reward and global reward value for measuring the performance of the proposed model. Reward is collected at each decision slot whereas global reward is the sum for all rewards collected over simulation time. In order to study the performance of our approach, we calculate the reward of our MDP strategy (MDP_str) at each decision period throughout simulation time. Then, we compute the global reward for each case. Thus, higher global reward value shows the availability of higher energy gain in parked EVs. Whereas, lower global reward shows lower energy level in parked EVs.

A. Simulation scenarios

In order to evaluate the performances of the proposed MDP model, we used Matlab and its MDP Toolbox [10] to implement our MDP_str scheme for resource allocation. Furthermore, in order to prove the effectiveness of our strategy, we compared the simulation results with Fixed_str1 and Fixed_str2 strategies. The later are defined as two different resource allocation algorithms that define fixed thresholds for the available energy in order to decide whether to serve users' demand or not. Specifically, Fixed_str1 and Fixed_str2 use respectively 30% and 60% of the available energy to process highly intensive computation tasks. However, no computation demands are served if the available energy level is less than the previously fixed thresholds.

We chose six different scenarios for the change in occupancy level over a 24 hours period as shown in Fig. 4. Specifically, the first 3 scenarios shown in Fig. 4.(a) consider 3 different Poisson distribution for occupancy level of EVs in

TABLE I. SUMMARY OF SIMULATION SCENARIOS

Case 1	(C1: Poisson): $\lambda = 30\%$	Case 4	(C4: fixed): $= 30\%$
Case 2	(C2: Poisson): $\lambda = 60\%$	Case 5	(C5: fixed): $= 60\%$
Case 3	(C3: Poisson): $\lambda = 80\%$	Case 6	(C6: fixed): $= 80\%$

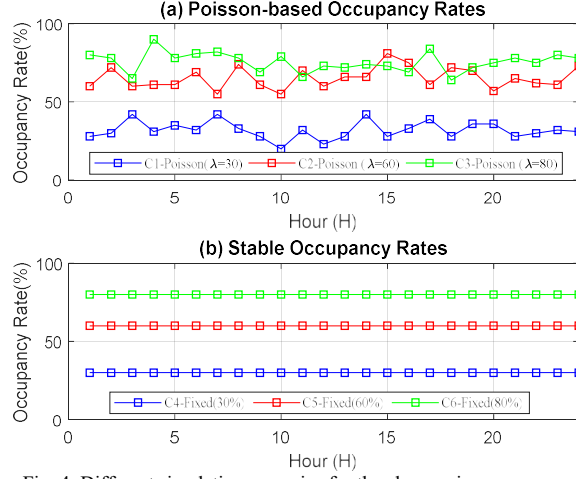


Fig. 4: Different simulation scenarios for the changes in occupancy rates over a 24 hours period

a given parking lot. The corresponding values of parameter λ are: 30%; 60% and 80%. Whereas, in Fig. 4.(b) shows the three other scenarios with fixed occupancy level over time, namely; 30%; 60% and 80%. Table 1 summarizes the used scenarios. Furthermore, the level of energy available in the parking lot is determined by the amount of energy offered by each EV. It is also influenced by the deference between arrival rate and departure rate of EVs over time. The following subsections present the different simulation scenarios followed by the summary and discussion of the main results.

B. Performance evaluation and discussion

As mentioned before, the proposed system is evaluated with six different cases (shown in Fig. 4). Fig. 5 shows the simulation results achieved by each resource management

strategy for the different scenarios. We can notice from Fig. 5(a) that our *MDP_str* always achieves higher scores in terms of global reward. Hence, *MDP_str* makes better use of the overall energy available in EVs in order to optimally serve a higher number of users' demands compared to the two other strategies. Specifically, the global reward is improved on average by 37.78% compared to *Fixed_str1* and 57.33% compared to *Fixed_str2*. Moreover, Fig. 5.(b) shows the global energy gain obtained by the three resource allocation policies for the six different scenarios. Indeed, *MDP_str* scores higher energy gain compared to *Fixed_str1* and *Fixed_str2*. The *MDP_str* improves the energy gain by 65.40% and 54.41% on average compared to *Fixed_str1* and *Fixed_str2* respectively.

To provide yet more extensive insights for the performance of the proposed scheme, we evaluated the different performance metrics over simulation time. Specifically, the noticeable changes in: (i) reward, (ii) global reward and (iii) energy level are evaluated throughout simulation time. Fig. 6 shows the result evaluation for a selection of three different scenarios, namely: case 2, case 4 and case 6.

First, the occupancy rates in case 2 follows a Poisson distribution with a mean value $\lambda = 60\%$. As it is shown in Fig.6.case2.(a), the *MDP_str* achieves higher values for rewards over time compared to *Fixed_str1* and *Fixed_str2*, which achieved comparable rewards. Consequently, in Fig.6.case2.(b) the final overall reward value achieved by *MDP_str* was twice the value achieved by the other two policies. Moreover, *MDP_str* obtained respectively 62.5% and 31.2% better energy gains at time $t=24$ compared to *Fixed_str1* and *Fixed_str2*.

Second, case 4 characterizes a stable energy level over time (30%). As depicted in the different sections of Fig. 6.case4, *Fixed_str1* scored zero reward (section (a) & (b)) and no energy gain (section (c)) throughout simulation time. This is because the available energy in EVs was below the fixed

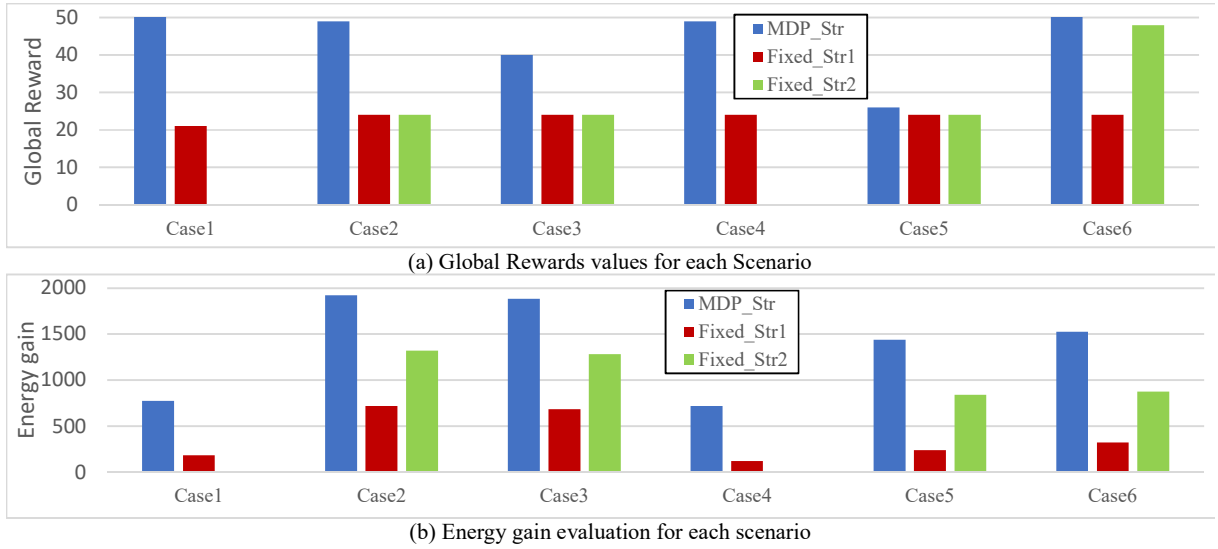


Fig. 5: Overall performance evaluation for the different simulation scenarios

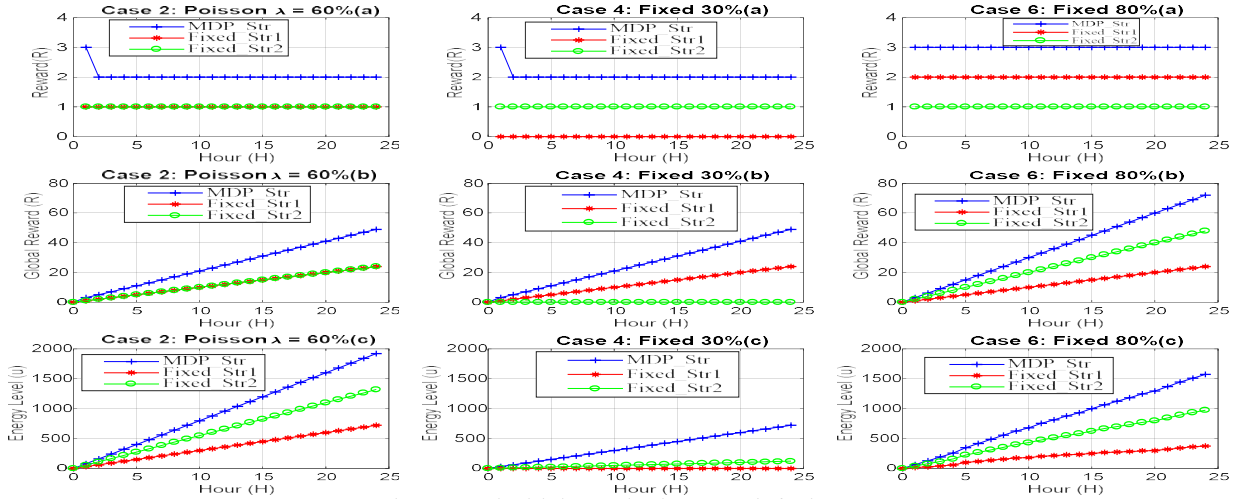


Fig. 6: Reward, Global Reward and Energy Gain for the 3 cases

threshold. However, *MDP_str* achieved better global reward and better energy gain compared to *Fixed_str2*.

Finally, in case 6, when the parking occupancy level is as high as 80 %, the results depicted in Fig. 6.case 6 (a) show a rather stable reward values for the three policies. However, *MDP_Str* always obtained higher rewards values than those of *Fixed_str1* and *Fixed_str2*. This would also have an influence on the global reward values depicted in Fig. 6.case 6 (b), where *MDP_Str* outperformed once again the other models. Furthermore, the high occupancy levels in the parking lot would result to a better availability of energy resources and consequently serves higher number of users demands. Fig. 6.case 6 (c) shows that, at time $t=24$, *MDP_str* has provided 78.7% and 42.4% better energy gain compared to *Fixed_str1* and *Fixed_str2* respectively. Specifically, *Fixed_str1* and *Fixed_str2* offered respectively 324 and 877 units of energy, while *MDP_str* achieved 1524 units.

As a summary to the performance evaluation section, the analysis of the obtained simulation results has shown the effectiveness of our MDP-based solution for resource allocation. Moreover, we proved that such scheme is suitable to optimize the usage of constrained resources, such as energy in parked EVs. *MDP_str* improves the overall energy used to compute highly intensive computation tasks by 75% and 58% compared with *Fixed_str1* and *Fixed_str2* respectively.

VI. CONCLUSION

In this paper, we proposed a new approach to optimize the usage of the surplus of energy available within parked electric vehicles for the processing of highly intensive computation tasks. The later would be offloaded from nearby mobile devices with constrained energy through a VFC architecture. An optimal decision-making scheme is formulated using a Markov Decision Process in order to maximize the long-term reward of the system. In the proposed model, the decision to accept or reject users' demands to fog nodes (EVs) is done by fog controller based on the available resources in a parking lot. Simulation results show that the proposed strategy outperforms other fixed strategies in terms of rewards and better energy gain.

Since using the same batteries all the time might significantly degrades EVs battery lifetime, as future work, we are already focusing on creating a history profile for battery usage in computation tasks. The aim would be to grant a fairness of EVs batteries use in order to improve their life time.

REFERENCES

- [1] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H. P. Tan, and S. Lin. Markov decision processes with applications in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 17(3):1239–1267, 2015.
- [2] D. T. Hoang, P. Wang, D. Niyato, and E. Hossain. Charging and discharging of plug-in electric vehicles (PEVs) in vehicle-to-grid (v2g) systems: A cyber insurance-based model. *IEEE Access*, 5:732–754, 2017.
- [3] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin and S. Chen, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860-3873, June 2016.
- [4] R. Cogill et al., "Parked cars as a service delivery platform," 2014 International Conference on Connected Vehicles and Expo (ICCVE), pp. 138-143, Vienna, 2014.
- [5] D. T. Hoang, P. Wang, D. Niyato, and E. Hossain. Charging and discharging of plug-in electric vehicles (pevs) in vehicle-to-grid (v2g) systems: A cyber insurance-based model. *IEEE Access*, 5:732–754, 2017.
- [6] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen, "An SMDP based resource allocation in vehicular cloud computing systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7920–7928, 2015.
- [7] H. Zhang, Q. Zhang and X. Du, "Toward Vehicle-Assisted Cloud Computing for Smartphones," in *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5610-5618, Dec. 2015.
- [8] R. Cogill et al., "Parked cars as a service delivery platform," 2014 International Conference on Connected Vehicles and Expo (ICCVE), pp. 138-143, Vienna, 2014.
- [9] N. Liu, M. Liu, W. Lou, G. Chen and J. Cao, "PVA in VANETs: Stopped cars are not silent," 2011 Proceedings IEEE INFOCOM, Shanghai, 2011, pp. 431-435.
- [10] M. J. Cros, Markov decision processes toolbox, Sept. 2016, available: <https://www.mathworks.com/matlabcentral/fileexchange/25786-markov-decision-processes-mdp-toolbox> [online].
- [11] E. Altman. "Constrained Markovien Decision Processes", chapter 1–4, pages 1–43. Chapman-Hall/CRC, 1998.