Reliability Engineering and System Safety 202 (2020) 107056

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Mobility-aware load Balancing for Reliable Self-Organization Networks: Multi-agent Deep Reinforcement Learning

Amin Mohajer[b], Maryam Bavaghar[a], Hamid Farrokhi[a,*]

[a] Department of Network Security and Information Technology, ICT Research Institute (ITRC), Tehran, Iran
[b] Department of Communications Technology, ICT Research Institute (ITRC), Tehran, Iran

ARTICLE INFO

Keywords:
Distributed Learning Automata, Self-Optimization Networking, Mobility Management
Cognitive Cellular Networks, Load Balancing

ABSTRACT

Self-Organizing Networks (SON) is a collection of functions for automatic configuration, optimization, and healing of networks and mobility optimization is one of the main functions of self-organized cellular networks. State of the art Mobility Robustness Optimization (MRO) schemes have relied on rule-based recommended systems to search the parameter space; yet it is unwieldy to design rules for all possible mobility patterns in any network. In this regard, we presented a Deep Learning-based MRO solution (DRL-MRO), which learns the required parameter's appropriate values for each mobility pattern in individual cells. Optimal mobility setting for Handover parameters also depends on the user distribution and their velocities in the network. In this framework, an effective mobility-aware load balancing approach applied for autonomous methods of configuring the parameters in accordance with the mobility patterns in which approximately the same quality level is provided for each subscriber. The simulation results show that the function of mobility robustness optimization not only learns to optimize HO performance, but also it learns how to distribute excess load throughout the network. The experimental results prove that this solution minimizes the number of unsatisfied subscribers (N_{us}) and it can also guarantee a more balanced network using cell load sharing in addition to increase cell throughput outperform the current schemes.

1. An overview of son functions

Self-organization Networking is a wide ranging research and standardization trend the scope of modern wireless networking. The optimal cell radius in next-generation cellular networks is continuously decreasing and this has increased operational expenses (OPEX) and caused the exponential enhancement of network operation complexity [1]. Self-organization networking (SON) as an effective approach in network resource management is introduced as a viable solution for minimizing such challenges by automating network operations. SON functions are defined based on NGMN standards [2] and represent the functions that should be automated. Some examples of the most effective SON models can be listed as:

- Robust Mobility Optimization,
- Load Balancing,
- Inter/Intra-Cell Interference Coordination (ICIC)
- Capacity and Coverage Optimization (CCO).

These models are traditionally introduced as recommender controllers using rule-based fuzzy engines in which knowing behavioral information of functions is essential for designers. The presented approach applies self-organized functions to cognitive wireless networks and implements them in a reinforcement learning platform using learning-based agents in which the feedback of the agents' actions is used to learn.

Each SON function is specified through a set of thresholds that begins the execution of an associated self-organization networking algorithm.

The primary SON function is a control agent with three main targets:

1) Network performance monitoring, considering threshold and trigger conditions.
2) Execution of self-organization algorithms and optimization of affected indices.
3) Checking the feedback relevant to effects of the taken actions.

However, traditional SON functions are developed mostly in the form of recommended systems as rule-based controllers that apply

* Corresponding author.
E-mail addresses: a.mohajer@ieee.org (A. Mohajer), maryam.bavaghar@gmail.com (M. Bavaghar), hamid.farrokhi@itrc.ac.ir (H. Farrokhi).

https://doi.org/10.1016/j.ress.2020.107056
Received 22 October 2019; Received in revised form 23 May 2020; Accepted 28 May 2020
Available online 06 June 2020

specific rules on target parameters.

Some efficient rule-based controllers for handover relation optimization are presented in [3–6]. Two SON functions relevant to physical characteristics, like as tilt adjustments to achieve load balancing, are introduced in [9] and [10]. In these functions, the rule designer should completely understand the impacts of each action on each scenario. In practice, these approaches aren't applicable, even for an expert system. The most critical issue is that their control procedures make complex structures which are so complicated to plan and to implement as SON rules [7]. Hence, they are usable exclusively for general actions and are not applicable for particular conditions of the network [8].

To resolve the functional issues of previous approaches, the self-organization networking technique is applied to cellular cognitive networks by empowering cognitive SON functions. The cognitive cellular networks apply self-organization networking beyond traditional rule-based controllers, in which cognitive SON functions autonomously learn how to move toward the optimal configuration. Particularly, it is suggested that cognitive rules be planned as learning agents which act based on the feedback of agents' actions. Learning has traditionally been used in some self-organization networking functions and achieved appropriate results, for example, in [11–13], although herein, a learning approach is suggested, which is effective for all self-organization functions. Reinforcement learning is applied in some self-organized networks, for example, *COO* [11,20], Load Balancing [21,22,34,35,38], Handover Control [23,30,31], User Quality of Experience and QoS assurance [16,19,32,33] and Resource Allocation [12,13,17,18,26-28]. This research generalizes this concept by considering SON functions as learning automata frameworks. Some advantages of using such reinforcement learning methods with efficient results are compared and the results discussed in [14] and [15].

In actual cellular networks, subscribers are seldom uniformly distributed, and this is considerable when a serving sector s is congested, while the network has available resources in adjacent sites. The only viable solution for this issue is to automatically redistribute the load among all sites. Mobility load balancing minimizes the number of subscribers which are not satisfied (N_{us}) by moving some of the edge users served by overloaded cells towards one or more adjacent sites, or so called target sites. In this paper we propose a cognitive cellular network (CCN) empowered by an efficient self-organization networking approach which enables the SON functions to separately learn and find the best configuration setting. An effective learning approach is proposed for the functions of the cognitive cellular network, which exhibits how the framework is mapped to SON functions. One of the main functions applied in this SON framework is mobility load balancing. In this paper, a novel Stochastic Learning Automata has been suggested as the load balancing function in which approximately the same quality level is provided for each subscriber. This framework can also be effectively extended to cloud-based systems, where adaptive approaches are needed due to unpredictability of total accessible resources, considering cooperative nature of cloud environments. The results demonstrate that the function of mobility robustness optimization not only learns to optimize HO performance, but also it learns how to distribute excess load throughout the network.

The structure of the paper is as follows. Section I briefly introduce an overview of SON functions and studies the current schemes in learning framework, Complexity Analysis, Convergence, Policy Gradient Prediction and Multi-Agent Learning Approach. Section II presents a novel approach to deep learning–based mobility robustness optimization in which Handover Performance Metrics, Handover Control Indexes & Sensitivity Degree and Search Method in Self-organization Parameters are studied. The mobility load balancing solution using stochastic learning automata and its application as a function of cognitive cellular networks are introduced in section III. Also, Section III characterizes the execution of SON functions in the mobility robust optimization framework. We discuss about the evaluation scenarios and the achieved simulation results in section IV. The conclusions drawn and recommendations made for future researches are discussed in section V.

2. Deep learning based mobility robustness optimization

One of the most critical issues in cellular network operation is determination of optimal handover settings considering hysteresis and time-to-trigger as the most effective control parameters. These two parameters should be configured based on the common subscriber velocity throughout the cell coverage, which can be relevant to both large state-space and large parameter-space. Thus, large spaces cannot be accurately assessed manually, which *mobility robust optimization* aims to resolve this problem . In this section, an effective novel Deep Reinforcement Learning Mobility Robustness Optimization called *DRL-MRO* is proposed.

For all handover events, relevant to hysteresis and time-to-tigger (called the "Trigger Point" from now on) in occurrence of a handover success, a ping pong or a radio link failure, the *mobility robustness optimization* algorithm tries to optimize robustness of the radio connectivity among the mobile subscriber devices and the serving network; for example, minimization of radio link failures decrease the number of ping-pongs and the useless handovers concurrently [2]. Many studies have investigated mobility robust optimization and some of the main results have been described in [28–32]. Most of these studies applied a recommender system as an automatic controller to find the best solution in accordance with the parameter space. The mentioned studies make two basic hypotheses, which are not adaptive to practical environments:

1) The mobility pattern is assumed not to be dynamic so that a basic scan is enough to obtain the optimal profile, while networks with a static velocity profile are impractical.
2) The underlying dependence between the handover indices and the control parameters is assumed. In addition to being uncertain to error in cases of inaccurate assumptions about this dependence, the necessary rules are very complicated, even with the correct model.

In order to resolve these challenges, the proposed DRL-based *mobility robust optimization* doesn't rely specifically on expert knowledge or command sequences; its agents learn the optimum trigger points (*OTPs*). It categorizes user equipments' speed to a list of mobility profiles in order to learn the optimum trigger points for each profile.

2.1. Handover performance metrics

The increment of hysteresis and/or time-to-trigger delays handover triggering by reducing handover attempts and ping-pongs. Nevertheless, when the handover is over-delayed, the signal-to-noise + interference decreases so much that is equivalent to occurrence of a radio link failure, specifically the *radio link failure because of late handovers* (*RLFLs*). Conversely, the degradation of *hysteresis* and/or *time-to-trigger* provokes earlier handovers. So, the handover includes a candidate cell which its signal strength is not permanently appropriate, and the user equipment re-initiates a handover back to the serving cell, resulting in a ping-pong. In a more severe state, the *ratio of signal-strength-to-interference + noise* in the candidate cell is so weak that the user equipment faces link failure during the inverse handover, which means a radio link failure because of early handover (*RLFE*).

Herein, some crititical metrics have been introduced that must be considered in the evaluation of the handover performance, such as radio link failure and ping-pong event.

1) Radio Link Failure: A failure in the radio access link happens if the level of the signal-to-noise/interference of the user equipment is below a certain amount for a period of time [27]. The rate of radio link failure, because of either too early handovers (*FE*) or too late

handovers (*FL*), is counted as the number of failures in a second in each cell.

2) Ping-pong rate: A ping-pong occurs if a handover success from cell X to cell Y happens in a time less than the "$PP-time$" just after a former successful handover from Y to X. The ping-pong rate is also calculable as the number of ping-pong occurrences per second in a cell area. $PP-time$ is not standardized, so in this research, it is considered approximately equal to the longest time-to-trigger.
3) Number of handover candidates: During execution of the scenario, all rates in each cell will be normalized to $\#\text{handover}-\text{candidates}$ to guarantee that all sectors have comparable conditions for assessing their events and statistics. However, the real number of handover candidates is related to two handover parameters, hysteresis and time-to-trigger, which specify if user equipment is a good candidate for handover or not. It is obvious that there will be a cyclic dependency between handover parameters, such that *hysteresis* and *triger time* are dependent to the *number of handover candidates,* although the *number of handover candidates* is needed in order to measure the hysteresis and time-to-trigger indices. To resolve this issue, the definition of handover candidate is modified in this research, so that the candidate has either initiated a handover or experienced a radio link failure during the assessment time, ensuring that all subscribers are counted exactly once, even if they experience several events during a time interval.
4) Performance of handover aggregate: To make a significant comparison among trigger points so as to choose the best one, the individual assessment of all three expressed metrics is not reliable. For this reason, one weighted aggregated metric is defined to include all three introduced metrics (radio link failure, ping-pong rate, number of handover candidates), which is formulated as Equation (1).

$$HOAP = w_1 P + w_2 F_E + w_3 F_L; \quad \sum w_i = 1 \tag{1}$$

The following conditions should be considered regarding the handover aggregate performance:

1) Handovers and handover success rates are not directly included in the *handover aggregate performance,* because minimizing ping-pongs also minimizes unessential handover events and successful handovers.
2) Handover failures due to radio link failures, which happen in a handover interval, are supposed to be considered due to radio link failures.
3) As well as improper handover settings, radio link failure can be caused by other reasons, such as poor coverage problems. In this paper, the factors that do not exist or their impacts are insignificant have been ignored. This seems a reasonable assumption according to the obtained outcomes which demonstrated that if handover is started early enough (hysteresis = 0 dB and time-to-trigger = 0 s), radio link failures are removed (with significantly more ping-pongs).
4) The determination of weight coefficients *wi* is subjective. Actually, these coefficients are selected so as to identically balance impacts of early handovers (ping-pong, link failure) against impacts of late handovers (link failure), that is w_3 equal to w_1 and w_2 combined. Because radio link failures are less desirable than ping-pongs, w_2 must be bigger than w_1. The mutually selected weight coefficients are as a vector w = (0: 2; 0: 3; 0: 5), which were applied in all scenarios in which handover performance was assessed.

2.2. Handover control indexes: sensitivity degree

The main goal of the *mobility robustness optimization* algorithm is to obtain the optimum trigger points as an ideal setting for networks with dynamic mobility pattern. In this scheme, to achieve an adaptive learning framework, the sensitivity of the main parameters to the speed of user equipment should be investigated. In the current study, all effective handover parameters have been assessed in four scenarios with a different velocity pattern. In which, user equipment is moving with a constant velocity pattern. The optimum trigger points change with velocity and the relation of *handover aggregate performance* with hysteresis and time-to-trigger is linear.

Table 1
DRL Mobility optimization: Velocity regimes

Target Area	Initial speed (Kmph) Mean	Range
Office environment	4	2-6
Dense (Center area)	12	8-16
Cluster Edge	34	30-38
Cluster Suburb	56	50-62
Street	115	110-120

As is obvious, a very high value of hysteresis is not acceptable for all velocity patterns; however, a low value of time-to-trigger with moderately high hysteresis could provide satisfactory results. In the same vein, setting a high value for time-to-trigger could be acceptable only at light velocities with medium to low hysteresis. Also, the best result in low velocities is achievable with moderate hysteresis and low to moderate trigger time, in which handover processes can efficiently be delayed without causing a critical issue, because the risk of radio link failure and the possibility of ping-pong are significantly low in lower velocities. It is obvious with a velocity equal to 10 km/h without considering the values of trigger time, the handover performance is good considering hysteresis equal to 2 dB.

Table 2 and 3.

With increasing the velocity, the handover delay should be decreased, especially when using trigger time. The handover aggregate performance is more sensitive to trigger time in comparison with the optimum trigger points. in other words, the handover functionality will significantly change with time-to-trigger, but it is almost constant with changes in hysteresis. It should be noted that at very high velocities, trigger time and hysteresis both have a great effect on handover control, and both of these parameters should be low. With velocities equal to 60 and 90 km/h, optimum trigger points are limited to the lower left corners of the grid, while the range of trigger time is 0-0.64 seconds. Also, with such a small value of trigger time, although there is great variation in the handover aggregate performance, this variation is dark in areas near the optimum point.

The clearest conclusion obtained from the result is that as demonstrated in [29,30], the optimum trigger points do not lie along any one diagonal in various speeds. Consequently, to specify the target trigger point, any *mobility robustness optimization* model should scan at least

Table 2
Simulation Parameters

Parameter	Value
etwork Bandwidth	20 MHz
Site-to-site distance	1000 m
interval	65 ms
# Subscribers	380 mobile, 80 static
User speed	Variable, mean speed = 2, 8, 25, 50 or 130 kmph
Mobility pattern	Random walking model
Pathloss formulation	$A + B.\log_{10}[\max(d[km], 0.035)]$; A = 128.1 and B = 37.6
Shadowing effect	Extent of deviation = 5.5 dB; Correlation factor = 45 m
transmit power BS	45 dBm
Antennas type	3 direction eNB, antenna gain 14.5 dBi, & antenna height = 32 m
User antennas	1 Omni directional, antenna gain 1.9 dBi & antenna height = 1.3m
Default data rate	1024 kbps

Table 3
Defined Actions based on Mobility states

Mean speed (kmph)	State (x)	Default Hysteresis (dB)	Default Time-to-Trigger (s)
0-5	0	4.0	0.0-6.3
5-9	1	3.0	0.0-3.67
9-13	2	2.1	0.0-2.35
13-18	3	2.1	0.0-0.98
18-23	4	1.3-2.1	0.0-0.75
23-30	5	1.3-2.1	0.0-0.35
30-35	6	1.3-2.1	0.0-0.325
35-42	7	1.3	0.0-0.48
42-50	8	0.9	0.0-0.48
50-58	9	0.6	0.0-0.37
58-70	10	0.6	0.0-0.325
70-80	11	0.0-0.4	0.0-0.12
80 +	12	0.0-0.4	0.0-0.12

more than half of the state-space.

2.3. Deep learning based mobility robustness optimization (DRL-MRO)

The main target of *Qlearning-based mobility optimization* is to determine the optimum hysteresis and trigger time and to maximize the handover aggregate performance in any mobility profile in the coverage area.

It should be noted that with this process, only cell performance will be affected, and these actions don't change the mobility states of the user equipments. Accordingly, that is sufficient for learning an *action a* relevant to *state x*, which improves the predicted immediate *reward r* at *time t*.

$$Q_{t+1}(x_t, a_t) = (1 - \alpha)Q_t(x_t, a_t) + \alpha[(x_t, a_t)] \quad (2)$$

In this equation, α represents the learning rate defined in the previous section.

The components of the *Qlearning-based mobility optimization* algorithm are described in the following.

1) State-Space of *mobility robustness optimization*: The essential handover setting in the radio access network is performed according to the mobility pattern of the user equipment in the coverage area as exhibited in the analysis of section V-B. State x is introduced as a degree of mobility which is a continuous variable indicating the average velocity during the self organization networking execution. We have described x as special bands for each of which the proper handover settings should be learned.

The mapping of mobility states and velocity profiles is considered, in which velocities are categorized into mobility states. Estimates of the default setting are used in the manual optimization process. It is supposed that the velocity is either calculable or estimated by the base stations. A simple model of speed estimation in cells can be based on the ratio of approximate cell size to average user equipment's stay time within the cell range. This parameter can be calculated accurately considering the Doppler peaks (Sacharov peaks) in the angular power spectrum effect [33]. In any case, with an appropriate estimation of velocity, the algorithm (Deep learning-based mobility optimization) is able to learn and set the optimal setting for all cells. The main simulation parameters and their values are listed.

2) Action Space of *Deep-learning mobility robustness optimization*: The action of each cell in this algorithm is to submit the optimal values of hysteresis and time-to-trigger to its associated subscribers. Without a self-organization networking solution, an operator manually sets default parameters which are achieved through trial and error; Based on a local search in the self-organization networking approach. However, a specific set of settings will be implemented, just like as items denoted in the following. Nevertheless, considering the dependence between optimum trigger points and velocity, the settings must be modified according to the immediate velocity changes in the cell. It is obvious from the results that when the hysteresis is higher than 5, the performance level for all real velocities is almost always sub-optimal. Thus, this study focused only on the hysteresis level up to 5 dB. However, differences in handover aggregate performance for some trigger-time settings are not satisfactory, particularly at low time-to-trigger levels. But there are no significant changes in the performance at any speed in most levels of hysteresis at a time-to-trigger equal to:08s; 0:1s; 0:16s. As such, not all time-to-trigger values are considered. For example, the possible time-to-trigger is a set of 12 values (0.04, 0.10, 0.128, 0.256, 0.32, 0.48, 0.512, 0.64, 1.02, 1.28, 2.56, 5.12) in second. The overall # *actions* called the *resulting action space* for each state is equal to 143 feasible combinations of the determined hysteresis and time-to-trigger.
3) Reward function of *Deep-learning mobility optimization*: The aim of this study is to minimize radio link failures without extremely increasing ping-pongs and handovers. Because the learner is a reward-maximizing agent, the reward $r_{x,\,t}$ must not be the positive HO-aggregate-performance index assessed during the execution of the self organization networking. As mentioned before, all individual rates should be normalized to the number of handover candidates, which is formulated as:

$$r_{x,t} = -(w_1 P + w_2 F_E + w_3 F_L)/NH; \quad (3)$$

Note that the weight vector used in this learning procedure should be adjustable, because the weight coefficients may need to be modified, particularly with regard to the small assessment period relevant to the SON operation interval. In other words, when no radio link failure occurs due to early handover, the results could be tilted in favor of too many ping-pongs. Then, the vector of weight coefficients is maintained as $w = (0{:}\ 2;\ 0{:}\ 3;\ 0{:}\ 5)$ for the usual condition and will be modified to $w = (0{:}\ 4;\ 0{:}\ 0;\ 0{:}\ 6)$ when no radio link failure due to early handover is observed.

4) Cooperative learning in Qlearning mobility optimization: Channel condition, which is considered by the level of reference signal received power, can affect handover triggering. Although applying algorithm $A3$ decreases this dependence on the signal strength values, likewise, decisions are made according to signal strength differences among all neighbor cells. In this case, handover KPIs depend on subscriber mobility pattern as well as degree of the control parameters. Considering mobility-handover states, a state occurring in one cell may be repeated in other cells at some other time. Also, there is no requirement that the cells learn independent policies, but they should learn a singular policy function according to the mobility states. The result is a cooperative deep learning problem in which each cell performs actions independently but update an exclusive Q-table, which indicates the shared learning policy.

Based on the cooperative property of the learning approach, this feature is established in cases where the values of the main parameters are comparable among adjacent sites. In other words, cell size and the applied transmission power are considered comparable in all network cells. In addition, the profile of the *reference signal received power* at the cell edges may cause different outcomes in different behaviors for different cells. If cells have simultaneously connected subscribers with various behavioral patterns, the solution may fail. In other words, one state for a site that covers a highway and an office park cannot be considered for the entire cell. Such a cell has simultaneously covered two user groups (low-speed office subscribers and fast-speed highway

subscribers), each of which requires a different setting. Consequently, because most cellular networks do not experience such particular states, a cooperative learning approach as presented in this research will not be completely practical.

2.4. Search method: Self-organization parameters

As mentioned, network cells have up to 160 feasible actions relevant to mobility states. Even applying cooperative approaches, assessing actions for consecutive times will be time consuming.

1) Methodology of parameter search: To speed up the algorithm's convergence, the 160 actions are categorized into sub-groups. For any state, 3 learning structures, $R1 - R3$ (exhibited in Figure 1), are applied that begin network-wide and move to local step-by-step. In $R1$, actions are selected from all sections of the grid to specify the area of desirable action. According to the sensitivity evaluation, combinations of high trigger time and small hysteresis is not optimal at all. In the same way, the target district will be excluded among the possible candidates demonstrated by "R1 actions" in Figure 1. The resulting trigger point of $R_1(RTP_1)$ determines the area of the optimal spot. The result has also been applied to describe the $search - space$ relevant to the subsequent regime R_2. In R_2, actions in the diagonal line which moves from grigger point 1, are obtained in order to achieve the acceptable latency for detecting the movement pattern. Subsequent actions differ in hysteresis by 1 dB to explore an adequate large action space. For example, if trigger point 1 is achieved as TP_1=(2: 0 dB; 0: 256s), at R_2 the agent explores the area indicated as R_2 actions. The achieved trigger point (TP_2) is also applied to describe the $search - space$ relevant to the subsequent regime R_2. Also, R_3 improved the learned trigger point 2 by searching its neighbor points. Trigger point 2 will be compared with its 4 neighbor points towards the left, right, top, or below. Also, "R_3 *Actions*" denotes the exploration district for R_3, considering that $Trigger\ Point\ 2 = (5{:}\ 0\ dB;\ 0{:}\ 128s)$.
2) The interval of the self-organization networking: All settings which are used in a cell are monitored during a self-organization networking interval. Assuming that various #*subscribers* are located in sites, various sites can experience variable number of events over the identical time interval. In the same way, instead of setting the self-organization networking interval based on a fixed time period, it is better to set it in accordance with the minimum possible handover statistics, which should happen considering the kind of each action. This value can be defined as sum of the dropped handover events (such as *succeed handovers, radio link failures* due to *earlier* and *late handovers*). This value has been set to 100 events, however, any value which guarantees that comparable counts of all the essential statistics are detected will be acceptable.
3) The *Qlearning-based mobility optimization algorithm*: According to the defined Q-learning elements, the optimization process is executed based on Algorithm 1, so that in each feasible state, the action sets are begun with R_1 considering 0 for the initial entries of the table. Afterward, the learning process is started during each self-organization networking duration t. Cell c monitors its surroundings during duration t, in which c specifies if optimization is required or not, for example, during the modification of $speed - state$. In the learning stage, c chooses an action as defined in former subsections; otherwise, it chooses the optimal action which has been learned. c also sends that action to all of the connected user equipments and begins obtaining the performance profiles required in the next time duration (t + 1). At the finishing point of time $t + 1$, c assesses its handover aggregate performance and receives the reward r_t for the action at time t. Finally, it renews the learning-table before the repetition of the cycle.

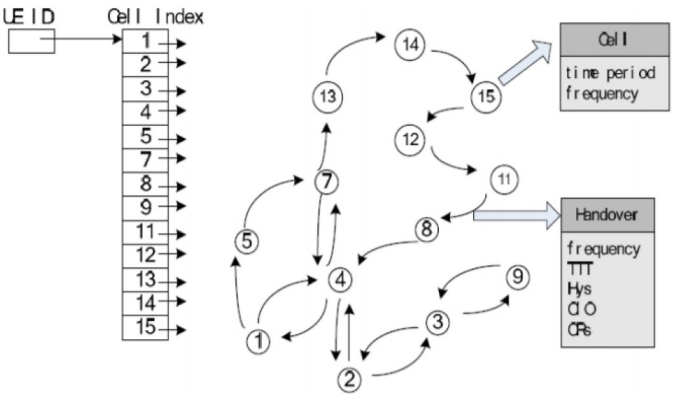

Fig. 1. The action space considering different learning scenarios R1, R2, R3.

Algorithm 1
The DRL Mobility Optimization Algorithm

Require: Subscriber velocity status during Self-organized networking execution
1: considering $R_i = R_1$; **initialize** action set $A_{x,,\,R1}$ for scenario 1 belong to all status x
Repeat in all self-organized networking duration t
2: **If** handover action is done at self optimization duration $t - 1$ **do**
3: calculate HO Aggregated performance and determine reward $r_{t-1}(x_{t-1}, a_{t-1})$
4: renew Q-table based on formulation (2)
end if
5: specify the mobility status x_t (based on table I)
6: **if** learning process finished in status x, **do**
7: find $a_{x,t} = a_x^{opt}$, as the best possible action relevant to the status x
8: **else if** the process of R_i exploration supposed imperfect **do**
9: consider $a_{x,\,t}$ (subsequently after $a_{x,t-1}$) belongs to $A_{x,,\,Ri}$
10: **else do**
11: consider $a_{x,t} = a_{x,\,Ri}^{opt}$, the most optimal value for status x at R_i
12: if all learning processes relevant to status x finished **do**
13: consider all learning stages completed for status x **do**
14: consider $a_{x,\,t}$ as the most optimal action relevant to status x
15: learning finished, unlimitedly apply $a_{x,\,t}$ in status x
15: **else do**
16: $R_i \leftarrow R_i + 1$
17: apply $a_{x,\,t}$ to determine $A_{x,,\,Ri}$. in other words reform action set A for R_i
18: **end if**
19: **end if**
20: broadcast target action $a_{x,\,t}$ to all subscribers located in the cell
21: $t \leftarrow t + 1$, observe, roll up and calculate statistics, continue from step 2
22: **end loop**

3. Mobility load balancing: stochastic learning automata

In actual cellular networks, subscribers are seldom uniformly distributed, and this is considerable when a serving sector s is congested, while the network has available resources in adjacent sites. The only viable solution for this issue is to automatically redistribute the load among all sites. Mobility load balancing minimizes the number of subscribers which are not satisfied (N_{us}) by moving some of the edge users served by overloaded cell s towards one or more adjacent sites, or so called *target sites*. In the formulation of the problem, T represents the set of adjacent target cells and ρ_s denotes the serving cell's load. One cell in the list is indicated with t and other ones are called T-cells. Considering $A3$ handover conditions given in the formulations, mobility load balancing modifies the load distribution by virtually shrinking serving overloaded s via simultaneously expanding the set of target cells. As mentioned in the self-organization networking standards of new generation mobile networks, this action can be performed by modifying the handover relation parameters of the serving cell s and target cell t and effective handover boundary thresholds such as cell individual offsets ($O_t^{s,t}$ *and* $O_s^{s,t}$; $\forall\ t \in T$).

In continue, the proposed learning automata based load balancing approach is presented as one of the main features enabled in the suggested SON model.

In this scenario, an learning automata is responsible for patching/dispatching a request. So, the learning automata submits each request to server i with probability $p_i(t)$. r denotes the number of servers each of which modelled based on $M/M/1$ queue in which the Input is a poisson process and its intensify is indicated by λ_i. Also, μ_i demonstrates the service rate. In server i, based on the introduced framework, the mean-response time (RT) is obtained as

$$MRT_i(t) = \frac{1}{\mu_i - \lambda_i(t)}, \tag{4}$$

In this formulation, $\lambda_i(t)$ represents the average input rate of server i. in accordance with $M/M/1$ model [12], we can consider $\lambda_i(t) = p_i(t)\lambda$ if $\{p_i\}$ is constant or with very slowly changes. For server i, consider $s_i(t)$ as the instantaneous RT and $\hat{\bar{s}}_i(t)$ can be an estimation for the response time. In which to estimate the average RT, we apply exponential mobility model with learning rate α.

$$\hat{\bar{s}}_i(t+1) = \hat{\bar{s}}_i(t) + \alpha(s_i(t) - \hat{\bar{s}}_i(t)). \tag{5}$$

The average RT for the other actions will be constant.

$$\hat{\bar{s}}_i(t+1) = \hat{\bar{s}}_i(t) \text{ for } j \neq i\ ,\ j \in [1, n]. \tag{6}$$

In this scenario, the reward and the penalty functions are defined as the following:

- Reward if $\hat{\bar{s}}_i \leq \frac{1}{r}\sum_{k=1}^{r} \hat{\bar{s}}_k$
- Penalty if $\hat{\bar{s}}_i > \frac{1}{r}\sum_{k=1}^{r} \hat{\bar{s}}_k$

We first should convert the markov process to being ergodic considering predefined lower bound p_{min} and the upper bound p_{max}. For obtaining this goal, consider a minimal value as the lower bound of the total power as $0 < p_{min} < 1$

Each action selection probability is denoted as x_i, in which $1 \leq i \leq r$ and r represents the quantity of selected actions. Therefore, the most major degree of each action-selection probability p_i, is equal to $p_{max} = 1 - (r-1)p_{min}$, where $1 \leq i \leq r$. The minimal degree p_{min} is dedicated to other $r-1$ actions, although p_{max} will be taken with the highest approximation. $\alpha(t)$ denotes the action index. The degree of $p_i(t)$ is also updated at the specific time t based on the below rules.

$$p_i(t+1) \leftarrow p_i(t) + \theta(p_{max} - p_i(t))$$

$$when\ \alpha(t) = i\ and\ v_i = 1$$

$$p_i(t+1) \leftarrow p_i(t) + \theta(p_{min} - p_i(t))$$

$$when\ \alpha(t) = j, j \neq i\ and\ v_i = 1,$$

In this formulation, θ and v_i are user-defined indexes with a value near to 0 and a reward function index. As $0 < \theta < 1$

3• $v_i = 1$, reward, if $s_i(t) \leq \frac{1}{r}\sum_{k=1}^{r} \hat{\bar{s}}_k(t)$.

3• $v_i = 0$, penalty, if, $\hat{\bar{s}}_i(t) > \frac{1}{r}\sum_{k=1}^{r} \hat{\bar{s}}_k(t)$.Algorithm 2 represents the formalized framework as psudo-code step by step. The mean of all RTs for all network nodes is described by $\hat{\bar{s}}(t)$.

$$\hat{\bar{s}}(t) \leq \frac{1}{r}\sum_{k=1}^{r} \hat{\bar{s}}_k(t) \tag{7}$$

3.1. Learning automata analysis

For the functional analysis of the scheme, we describe the approximate behavior of the learning automata based load balancing solution. So, the proposed approach will be analysed from the convergence and stability perspectives. In this framework, for small values of α considering $\theta << \alpha$, $\hat{\bar{s}}_i(t)$ will be estimated for all $1 \leq i \leq r$. by $MRT_i(p_i(t)) = \frac{1}{\mu_i - \lambda_i p_i(t)}$

Algorithm 2
Learning Automata Load Balancing

Loop
1: *At time instant* t, *Consider the probability vector* $[p_1, p_2, ..., p_r]$ *and select an action*
2: *Update the RT estimations*
• *For the selected action update the RT*
$\hat{\bar{s}}_i(t+1) = \hat{\bar{s}}_i(t) + \alpha(s_i(t) - \hat{\bar{s}}_i(t))$.
• *For the other actions in server* i, *the response estimation should be constant, so*
$\hat{\bar{s}}_j(t+1) = \hat{\bar{s}}_j(t)$ *for* $j \neq i,\ j \in [1, r]$.

3: *The function of Penalty/Reward is as*
$v_i = 1$: *(Reward) if:* $\hat{\bar{s}}_i \leq \frac{1}{r}\sum_{k=1}^{r} \hat{\bar{s}}_k$,
Otherwise, $v_i = 0$ *(Penalty)*,

4: *Consider* $\alpha(t)$ *as the selected action's index,* $p_i(t)$ *will be updated at time* t *according to the below ruls:*
$p_i(t+1) \leftarrow p_i(t) + \theta(p_{max} - p_i(t))$
when $\alpha(t) = i$ *and* $v_i = 1$,
$p_i(t+1) \leftarrow p_i(t) + \theta(p_{min} - p_i(t))$
when $\alpha(t) = j, j \neq i$ *and* $v_i = 1$,

It should be noted that for $1 \leq i \leq r$, $\hat{\bar{s}}_i(t)$ converges to $\bar{s}_i(p_i(t))$ in which $\bar{s}_i$ demonstrates the MRT_i. This is obvious upon the stochastic probability theory [4]. We have $\theta \ll \alpha$, and we know in comparison with $\hat{\bar{s}}_i's$, $p_i's$ grow slower that warranties multi-time scale differentiation. Applying the notation $\alpha(t) = i$ describes that action i is selected at time t, therefore $\hat{\bar{s}}_i(t+M)$ is calculated as

$$\hat{\bar{s}}_i(t+M) = \hat{\bar{s}}_i(t) + \alpha \sum_{k=0}^{M-1} I_{\{\alpha(t+k+1)=i\}}(s_i(t+k) - \hat{\bar{s}}_i(t+k))$$

According to the ruls of tiny-step processes theory, it can be supposed that when α is sufficiently small, the probability vector $[\hat{\bar{s}}_1(t), \hat{\bar{s}}_2(t), ..., \hat{\bar{s}}_r(t)]$ will be kept fixed approximately during a discrete time interval $\{t, t+1, . . ., t+M\}$. Hence, below probability equations are obtained for $1 \leq i \leq r$:

$$\hat{\bar{s}}_i(t+M) \approx \hat{\bar{s}}_i(t) + M\alpha(R_i(t, M) - Q_i(t, M)\hat{\bar{s}}_i(t)) \tag{8}$$

For $i \in [1, r]$, in the condition that the value of the approximations $\{\hat{\bar{s}}_1(.), \hat{\bar{s}}_2(.), ..., \hat{\bar{s}}_r(.)\}$ are constant at $\{\hat{\bar{s}}_1(t), \hat{\bar{s}}_2(t), ..., \hat{\bar{s}}_r(t)\}$, and M is sufficiently large value, it's possible to estimate the quantities:

$$R_i(t, M) = \frac{\sum_{k=0}^{M-1} I_{\{\alpha(t+k+1)=i\}} s_i(t+k)}{M},$$

And also,

$$Q_i(t, M) = \frac{\sum_{k=0}^{M-1} I_{\{\alpha(t+k+1)=i\}}}{M},$$

The approximation vector $p_1(.), p_2(.), ..., p_r(.)$ can be considered fixed in the timing interval $\{t, t+1, . . ., t+M\}$, based on our study that p_i grows at slower time scale in comparison with $\hat{\bar{s}}_i$. It should be noted that the equation $\theta \ll \alpha$ allows the separation during this time interval. Considering M is sufficiently large, $Q_i(t, M)$ is achievable as

$$Q_i(t, M) = \frac{\sum_{k=0}^{M-1} I_{\{\alpha(t+k+1)=i\}}}{M},$$

Which indicates the part of time action i selected in time duration $[t, t+M]$, and will be converged to $p_i(t)$. If we suppose the action probabilities constant, we have convergence of the RT processes $s_i(.)$, to a fixed distribution with the average $\bar{s}_i(p_i(t))$.

$$R_i(t, M) = \frac{\sum_{k=0}^{M-1} I_{\{\alpha(t+k+1)=i\}} s_i(t+k)}{M}$$

Which may be etimated by $p_i(t)\ \bar{s}_i(p_i(t))$. Applying the mentioned estimations, it's concluded from formulation (8), that the modification of the vector $[\hat{\bar{s}}_1(.), \hat{\bar{s}}_2(.), ..., \hat{\bar{s}}_r(.)]$ decreases to the below ODE system

when α is significantly low:

$$\frac{\hat{\bar{s}}_i(t)}{dt} = p_i(t).\,(\bar{s}_i(p_i(t)) - \hat{\bar{s}}_i(t)). \tag{9}$$

The degree of formulation (9), decreases to have the running RT approximation, given by $[\hat{\bar{s}}_1(.), \hat{\bar{s}}_2(.),\ldots., \hat{\bar{s}}_r(.)]$, modifying to a SS vector $[\bar{s}_1(p_1(t)),\ \bar{s}_2(p_2(t)), \bar{s}_r(p_r(t))]$, with considering α close to zero. In accordance with the features of M/M/1 queue model and the above assumptions, $\bar{s}_i(p_i(t))$ is equal to

$$\bar{s}_i(p_i(t)) = MRT_i(p_i(t)) = \frac{1}{\mu_i - \lambda_i p_i(t)}. \tag{10}$$

Actually, in this scenario, the reward is defined based on the event that immediate RT detected whenever a server less than $\hat{\bar{s}}(t)$ is selected. which $\hat{\bar{s}}(t)$ is the mathematical average of $\hat{\bar{s}}(t)$ for $1 \le i \le n$. In continue, it has been exhibited that the reward probability reduces with increasing p_i.

We can consider that $D_i(t)$ is slways severely decreasing as a function of p_i. Reward probability $D_i(t) = Prob(s_i(t) \le \hat{\bar{s}}(t))$ that $\hat{\bar{s}}(t)$ is calculable via formulation (7). Based on the achieved results from the previous assumption, if in comparison with $\hat{\bar{s}}_i's$, $p_i's$ grow slower that warranties multi-time scale differentiation $\hat{\bar{s}}(t)$ can be approximated by the sum of the average RT relevant to each server, for example, summation of $MRT_i(t)$, $1 \le i \le r$. Also, $\hat{\bar{s}}(t) \approx \frac{\sum_{k=i}^{r} \bar{s}_i(p_i(t))}{r} = \frac{\sum_{k=i}^{r} MRT_i(p_i(t))}{r}$. The approximation of $s_i(t) > \hat{\bar{s}}(t)$ is computable as [29]:

$$D_i(t) = Prob(s_i(t) \le \hat{\bar{s}}(t)) = 1 - exp(-\hat{\bar{s}}(t)(\mu_i - \lambda_i(t))). \tag{11}$$

It's demonstrated that this value decreases with increasing $p_i(t)$. To obtain this goal, $\frac{dD_i(t)}{dp_i}$ exhibited by:

$$\frac{dD_i(t)}{dp_i} = \frac{\delta D_i(t)}{\delta p_i} + \sum_{\substack{j=1 \\ j\neq i}}^{r} \frac{\delta D_i(t)}{\delta p_j}\,\frac{\delta p_j}{\delta p_j}.$$

For simplicity we defined an dummy constants $b_j \ge 0$ for $j \ne i$, like as [13], with considering these dummy parameters, it's given

$$p_1 = b_1 p_i,\ p_2 = b_2 p_i,\ \ldots p_r = b_r p_i,\ with\ b_j \ge 0\ for\ j \ne i.$$

Hence,

$$\begin{aligned} p_1 &= \frac{b_1(1-p_i)}{\sum_m b_m} \\ \ldots &= \ldots \\ p_j &= \frac{b_j(1-p_i)}{\sum_m b_m} \\ p_r &= \frac{b_r(1-p_i)}{\sum_m b_m} \end{aligned} \tag{12}$$

Because $\sum_m p_m = 1$, therefore: $\frac{dp_j}{dp_i} = \frac{-b_j}{\sum_{m\neq j} b_m} < 0$ for all j $\ne$ iBased on definition of $D_i(t)$, we have:

$$D_i(t) = 1 - \exp(-\hat{\bar{s}}(t)(\mu_i - \lambda_i(t))$$

$$= 1 - exp\left(-\frac{\mu_i - \lambda_i(t)}{r}\sum_k \frac{1}{\mu_k - \lambda_k(t)}\right)$$

$$= 1 - exp\left(-\frac{1}{r} - \sum_{\substack{k=1 \\ k\neq i}}^{r} \frac{1}{r(\mu_k - \lambda p_k(t))}\right).$$

This definition is completely independent of p_i, which shows $\frac{\delta D_i(t)}{\delta p_i} = 0$. Hence, $\frac{\delta D_i(t)}{\delta p_i}$ modified to $\frac{dD_i(t)}{dp_i} = \sum_{\substack{j=1 \\ j\neq i}}^{r} \frac{\delta D_i}{\delta p_j}\,\frac{\delta p_j}{\delta p_i}$. With some algebraic manipulations the equation is simplified to

$$\frac{\delta D_i}{\delta p_j} = \frac{\lambda}{r(\mu_i - \lambda p_i(t))^2} \exp(-\hat{\bar{s}}(t)(\mu_i - \lambda_i(t))\Big).$$

So, $\frac{dp_j}{dp_i} < 0$, $\frac{\delta D_i(t)}{\delta p_i} = \sum_{\substack{j=1 \\ j\neq i}}^{r} \frac{\delta D_i}{\delta p_j}\,\frac{\delta p_j}{\delta p_i} < 0$.

When the lower bound of power p_{min} is very small, the learning automata system has a specific equilibrium point as a solution.

$$E[p_i(t+1) - p_i(t)|p(t)] = p_i D_i(p_i)[\theta(1 - p_i)]$$

$$+ \sum_{\substack{j=1 \\ j\neq i}}^{r} p_j D_j(p_j).\,[\theta(p_{min} - p_i)]$$

Therefore,

$$\begin{aligned} &E[p_i(t+1) - p_i(t)|p(t) = p] = \\ &p_i D_i(p_i).\,[\theta(1 - p_{max} + 1 - p_i)]\ + \sum_{\substack{j=1 \\ j\neq i}}^{r} p_j D_j(p_j)[\theta(p_{min} - p_i)] \end{aligned} \tag{13}$$

$$= p_i D_i(p_i).\left[\theta\left(1 - p_{max} + \sum_{\substack{j=1 \\ j\neq i}}^{r} p_j\right)\right] + \sum_{\substack{j=1 \\ j\neq i}}^{r} p_j D_j(p_j)[\theta(p_{min} - p_i)]. \tag{14}$$

Considering the fact that 1-p_{max} = (r-1) p_{min}, formulation (14) can be modified as:

$$E[p_i(t+1) - p_i(t)|p(t) = p] = \theta \sum_{\substack{j=1 \\ j\neq i}}^{r} p_i p_j (D_i(p_i) - D_j(p_j))$$

$$+ \theta p_{min}\left(\sum_{\substack{j=1 \\ j\neq i}}^{r} p_j D_j(p_j)\right)$$

$$- \theta(r-1)p_{min} p_i D_i(p_i)$$

$$= \theta \sum_{\substack{j=1 \\ j\neq i}}^{r} p_i p_j (D_i(p_i) - D_j(p_j))$$

$$+ \theta p_{min} \sum_{\substack{j=1 \\ j\neq i}}^{r} (p_j D_j(p_j) - p_i D_i(p_i)) \approx \theta\omega_i(p)$$

Where $\omega_i(p)$ is described as $\sum_{\substack{j=1 \\ j\neq i}}^{r} p_i p_j (D_i(p_i) - D_j(p_j))$

Whenever p_{min} is close to zero, i.e., as $p_{min} \to 0$, the formulation E $[p_i(t+1) - p_i(t)|p_j = p]$ is calculable via:

$$E[p_i(t+1) - p_i(t)|p(t) = p] = \theta\omega_i(p). \tag{15}$$

Hence,

$$\frac{dp_i(t+1)}{dt} = \theta\omega_i(p) \tag{16}$$

We can now continue with the achieved results.

a) There is an unique zero for $\omega(p) = (\omega_1(p),\ \omega_2(p), \ldots.,\omega_r(p))$ as a specific solution in the adjacency of $p^* = (p_1^*,\ldots,p_r^*)$.

The mentioned claims describe a system with r equalities:

$$\begin{cases} \sum_{\substack{j=1 \\ j\neq 1}}^{r} p_1 p_j (D_1(p_1) - D_j(p_j)) = 0 \\ \sum_{\substack{j=1 \\ j\neq 2}}^{r} p_2 p_j (D_2(p_2) - D_j(p_j)) = 0 \\ \vdots \\ \sum_{\substack{j=1 \\ j\neq r}}^{r} p_n p_j (D_r(p_r) - D_j(p_j)) = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} p_1 \sum_{\substack{j=1 \\ j\neq 2}}^{r} p_j (D_1(p_1) - D_j(p_j)) = 0 \\ p_2 \sum_{\substack{j=1 \\ j\neq 2}}^{r} p_j (D_2(p_2) - D_j(p_j)) = 0 \\ \vdots \\ p_n \sum_{\substack{j=1 \\ j\neq r}}^{r} p_j (D_r(p_r) - D_j(p_j)) = 0. \end{cases}$$

It should be noted that in this problem we applied the lower bound of power p_{min} for the framework. Therefore, it guarantees that $p_1 \neq 0$, $p_2 \neq 0$, …, $p_r \neq 0$. So, it's confidently divide the *i'th* formulation by p_i, yielding:

$$\Leftrightarrow \begin{cases} \sum_{\substack{j=1 \\ j\neq 1}}^{r} p_j (D_1(p_1) - D_j(p_j)) = 0 \\ \vdots \\ \sum_{\substack{j=1 \\ j\neq 2}}^{r} p_j (D_2(p_2) - D_j(p_j)) = 0 \\ \vdots \\ \sum_{\substack{j=1 \\ j\neq 1}}^{r} p_j (D_r(p_r) - D_j(p_j)) = 0. \end{cases}$$

with some algebraic manipulations, we will have:

$$\Leftrightarrow \begin{cases} D_1(p_1) = \sum_{j=1}^{r} p_j D_j(p_j) \\ \vdots \\ D_2(p_2) = \sum_{j=1}^{r} p_j D_j(p_j) \\ \vdots \\ D_r(p_r) = \sum_{j=1}^{r} p_j D_j(p_j) \end{cases}$$

Which it guarantees $D_1(p_1) = D_2(p_2) = ...=D_r(p_r)$. In continue, it exhibited that the obtained solution is unique.

a) The uniqueness of the equilibrium point to which the algorithm converges p^* must be checked. In this regard, we assume there is existing $q^* = (q_1^*,...,q_n^*)$, that is another solution of $\omega(q)$ so that $q^* \neq p^*$.

Because p^* and q^* are two inequal probability vectors $q^* \neq p^*$, it is certainly obvious that each of them has at least two equilibrium points i and j so that $p_j^* > q_j^*$ or $p_j^* < q_j^*$. Obviousely, it is proven that with increase any one entity of a probability vector, another entity must be reduced because the sum of the entities must be constant. Now suppose that $p_j^* > q_j^*$. So, with considering the uniformity of $D_i(.)$, it's achieved that $D_i(p_i^*) < D_i(q_i^*)$. Vice versa, if $p_j^* < q_j^*$ it expresses that $D_j(p_j^*) > D_j(q_j^*)$, which this conclusion is obvious considering the monotonicity of $D_j(.)$. On the other hand, due to p^* and q^* are both the zero solution, we must have $D_i(p_i^*) = D_j(p_j^*)$, and also, $D_i(q_i^*) = D_j(q_j^*)$. And it is an contradiction! because it's not possible to simultaneously obtain: $D_i(p_i^*) < D_i(q_i^*)$ that is equivalent to $D_i(p_i^*) = D_i(q_i^*)$ and $D_i(p_i^*) = D_i(q_i^*)$. Consequently, uniqueness of p^* is proved.

The algorithm will be converged in an optimal zero point, which is alternatively Lyapunov stable. For proving this theorem we should follow the Lyapunov function:

$$V(p(t)) = \sum_{k=i}^{r} \int_{0}^{pt} D_k(z)dz.$$

The derivation is as:

$$\frac{dV(p(t))}{dt} = \sum_{i=1}^{r} \frac{dV(p(t))}{dp_i} \frac{dp_i}{dt}. \tag{17}$$

Considering the integral derivation, $\frac{dV(p(t))}{dp_i} = D_i(t)$. So, based on Eq. (16), $\frac{dp(t)}{dt} = \theta\omega_i(p)$. Thus

$$\frac{dV(p(t))}{dt} = \theta \sum_{i=1}^{r} D_k(t)\omega_k(p), \tag{18}$$

where $\omega_i(p)$ is described as $\omega_i(p) = \sum_{\substack{j=1 \\ j\neq i}}^{r} p_i p_j (D_i(p_i) - D_j(p_j))$. So,

$$\frac{dV(p(t))}{dt} = \theta \sum_{i=1}^{r} D_i \sum_{j=1}^{r} p_i p_j (D_i - D_j)$$

$$= \theta \sum_{i=1}^{r} \sum_{j=1}^{r} p_i p_j (D_i^2 - D_i D_j)$$

$$= -\frac{\theta}{2} \sum_{i=1}^{r} \sum_{j=1}^{r} p_i p_j (D_i - D_j)^2$$

Therefore, $\frac{dV(p(t))}{dt} \leq 0$As it's obvious, the value of the Lyapunov function should be equal to 0 at its equilibrium point. So, $\frac{dV(p(t))}{dt} = 0$. And for every i, j, $p_i^* p_j^* (D_i^* - D_j^*)^2 = 0$. Since $p_i^* > p_{min}$ and $p_j^* > p_{min}$, the formulation $D_i^*(p_i^*)D_j^*(p_j^*) = 0$ is certainly true for all i, j

$$D_i^*(p_i^*) = D_j^*\left(p_j^*\right) = 0.$$

Based on the Lyapunov theorem, it's obvious that p^* is the optimal point of the problem. The results denoted that, although all SON functions apply a similar procedure, particular arrangements are needed for each SON function based on its specific conditions. For example, each SON function needs a special strategy to recognize its action space. In practice, the higher performance achieved by the proposed method demonstrates the effectiveness of the self-organized cognitive approach. Moreover, learning automata provides an appropriate solution for developing self-optimization functions. It is noteworthy that learning mobility optimization is capable to set the handover parameters such as hysteresis and trigger time for each special mobility pattern. Using the cooperative learning method, all base stations work based on learning a single-policy function (unique Q-table). Such an applicable approach is effective in various environments as exhibited by the proper KPI results achieved through the practical scenarios with subscribers with different and adaptively variable speeds.

In the reinforcement learning load balancing, the agents are able to set the best value of the cell individual offset required to decrease overload in various load statuses. Learning-based load balancing also learns the various cell individual offset adjustments which are needed for different load conditions. One other impact of load redistribution is increasing number of the satisfied subscribers from the perspective of data rate. In terms of convergence and complexity, it is obvious that with similar costs, the algorithm has a linear order of complexity to the cell density. Also, there is always a trade-off between adaptation velocity and complexity.

4. Simulation results

In this section, the presented approach was evaluated using Network Simulator and LTE Mobile BroadBand simulator with the ability of simulation up-link/down-link 3GPP LTE Advanced radio access network described in [24,25]. This software plane is empowered by the required SON features. For example, some independent classes have been added for Mobility Optimization and Load Balancing functions. The aims of *mobility robustness optimization* are to determine the optimum network setting and to maximize the handover performance in any mobility profile in the coverage area.

4.1. KPI assessment and User QoS

The achieved outcomes exhibited in the following plots, discuss about load changes in cells to assess the dynamic performance of the

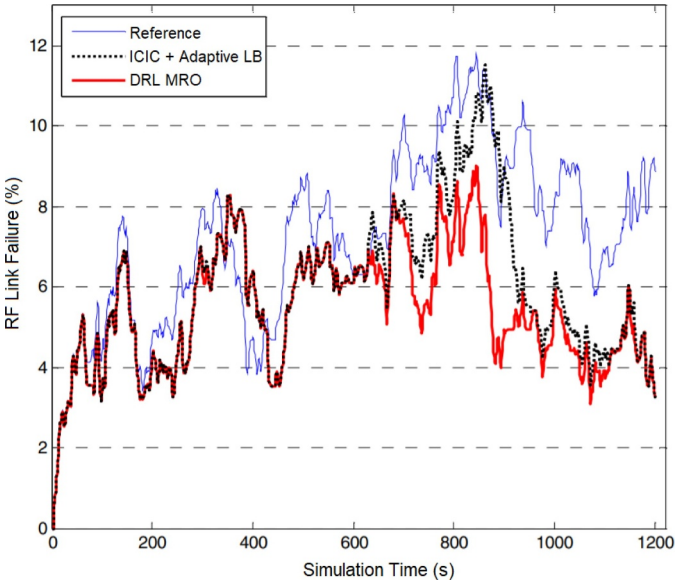

Fig. 2. The ratio of radio frequency link failure during the simulation time.

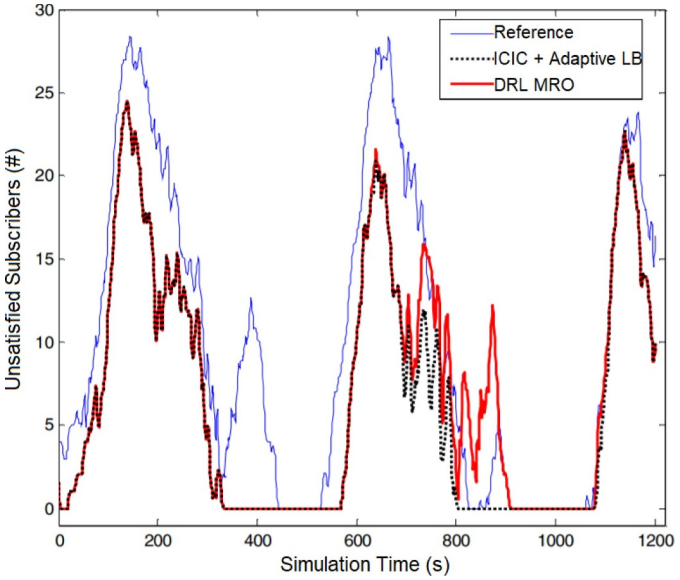

Fig. 3. The number of unsatisfied subscribers during the simulation time.

proposed approach and the number of dissatisfied subscribers (N_{us}) which is considered as the main impact of the algorithm and its effect on subscribers' quality of service. Furthermore, in the next stage of the evaluation, the effectiveness of learning automata load balancing and reactive load balancing algorithms in comparison with the reference is investigated. Note that the *Reference* refers to the primary scenario in which the network is not SON-empowered and works without any self-mobility management capabilities.

In all of the scenarios, all user equipments have independent random-varying speeds. This feature was deployed by assigning accidental speeds to the user equipments at the beginning of the simulation and by accidentally modifying the speeds at the beginning and during every stage by up to 50% in each scenario. i.e. in a network with 300 assigned subscribers located in a suburb with a velocity of 80 km/h, and the velocity vector is continuously changing.

Figure 2 describes the failed radio frequency link ratio during the simulation scenario in comparison with [36] which applied a long short term memory (LSTM) to detect the channel characteristics automatically and suggests a novel solution for inter-cell coordination (ICIC) and mobility load balancing together, in addition to a fixed mobility load balancing approach [37] and the reference network in two typical cases. As it is obvious in the achieved result, the proposed DRL MRO approach has significant less radio link failure and applying this algorithm, the RF link failure is kept less than 8%. Also, during the simulation scenario, the results of the ICIC empowered by the Adaptive LB are aften close to the results of DRL MRO.

Learning *Trend of DRL MRO:* The *DRL MRO* method was used in different speeds (15 km/h and 45 km/h). All results indicate that the agent learns how to minimize radio link failure due to late handovers (FL) by trading them off with ping-pongs that have less impact on the subscriber's quality of experience (QoE). This exists until link failures remain low due to early handovers. Although in some speed scenarios, *DRL MRO* suffers from high radio link failure at the start, because it sets the settings among a large parameter state. Nevertheless, in continue, the agent continuously decreases link failures due to late handover by trading such degradation with enhancements in ping-pong. Afterward, as soon as it detects dropping returns, these trends are stopped, for example, when each significant reduction in *failure due to late handover* translates into an exceeding ping-pong, or it causes link failure due to early handover. The outcomes indicate that with enough learning time, *DRL MRO* is capable for learning the proper hysteresis and trigger time setting for all mobility profiles.

Figure 3 exhibits the difference in performance between *DRL MRO* and two other algorithms for variable speed case. Although the obtained results proves the effective appropriateness of DRL MRO compared with the Reference, but it is clear in that *DRL MRO* performs poorly some times, which is equivalent to the Limitations of DRL MRO to obtain a better radio link failure ratio, which causes little weakness in load balancing, while ICIC with adaptive LB works without considering quality indexes. Also it can somewhat be relevant to learning stage R_1. Afterwards, the performance is increased during the next learning stages as the DRL MRO focuses on the optimum trigger points. In the next part of the simulation results, we will show how can resolve this issue by enabling stochastic learning automata algorthm for load balancing.

4.2. Mobility-aware load balancing: stochastic learning automata

In comparison with other mobility load balancing approaches for cellular networks, we have compared the performance of the proposed scheme, named Stochastic Load Balancing (SLA-MLB), with [36] which applied a long short term memory (LSTM) to detect the channel characteristics automatically and suggests a novel solution for inter-cell coordination (ICIC) and mobility load balancing together, in addition to a fixed mobility load balancing approach [37].

4.3. Robust load balancing: blocking rate

We have investigated on two major network KPIs for performance comparison between different mobility load balancing approaches which the achieved results shown in Figures 4(a) and 4(b). Hence, these indexes have been simulated in user mobility scenario with Constant bit rate traffic model whose call duration is geometric

- Blocking Rate
- Handover Failure Ratio

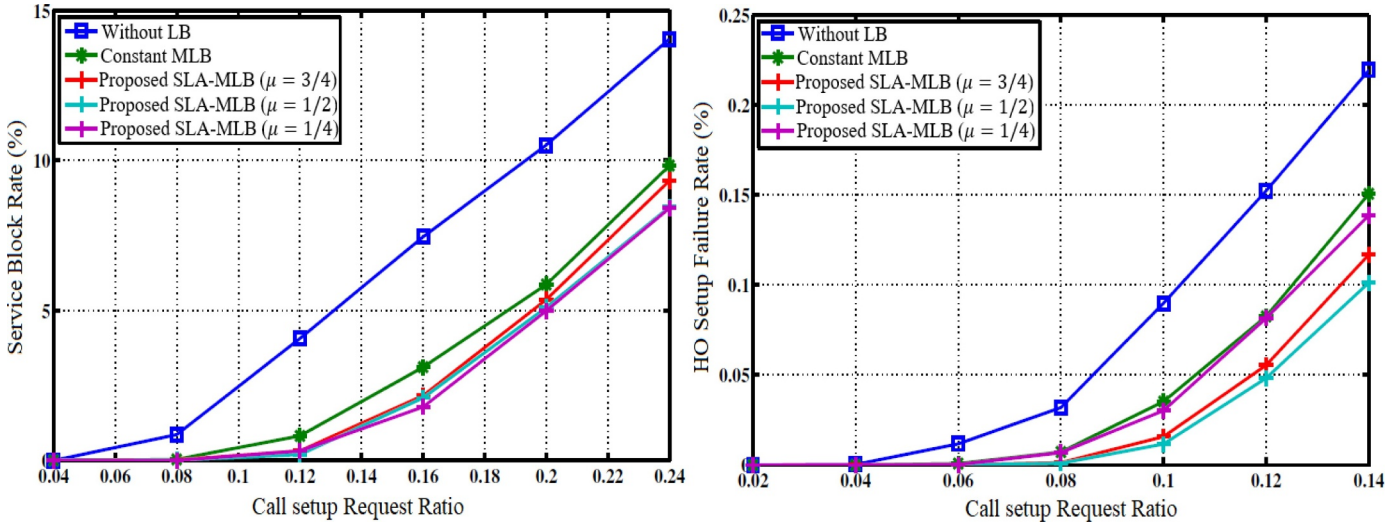

Fig. 4. Evaluation of Stochastic-Learning-Automata load balancing with different service rates: Blocking Probability and Handover failure (%).

As shown in 4(a), the stochastic learning automata has a tangible advantage over other algorithms in terms of traffic efficiency. It's handover failure rate and blocking probability are significantly limited compared to the other schemes. However, with limited resource condition, we cannot see any significant difference between this approach's functionality and other schemes.

4.4. Robust load balancing: handover performance

According to Fig 4(b) the load balancing approach based on stochastic learning automata is able to reduce blocking rate about 15% in addition to decreasing the handover failure rate more than 30%.

4.5. Robust load balancing: user satisfaction

The Load Distribution Indicator is a variable which indicates the degree of similarity among cells. If the network load is balanced and the users are distributed in the network, the value of load distribution indicator is close to 1. On the contrary, if network load is distributed completely unbalanced, this value will decrease to a proportion of the total number of cells. Therefore, the goal of load balancing algorithms is maximizing it. According to the obtained results, the proposed robust load balancing approach has better performance in distribution of network load among cells. One of the route cause is the proposed approach can adjust the handover thresholds to load balancing execution although such adjustments is time consuming. Also, Figure 5 demonstrates user satisfaction index which implies the quality of user experience based on the number of satisfied users among all the subscribers. As it is obvious, the total number of unsatisfied users when our proposed load balancing approach activated, is effectively less than other two traditional balancing approaches.

The results denoted that, although all SON functions apply a similar procedure, particular considerations are needed for each SON function based on its specific conditions. For example, each SON function needs a special strategy to recognize its action space. In practice, the higher performance achieved by the proposed method demonstrates the effectiveness of the self-organized cognitive approach. Moreover, stochastic learning provides an appropriate solution for developing self-optimization functions. It is noteworthy that the DRL-MRO is capable to

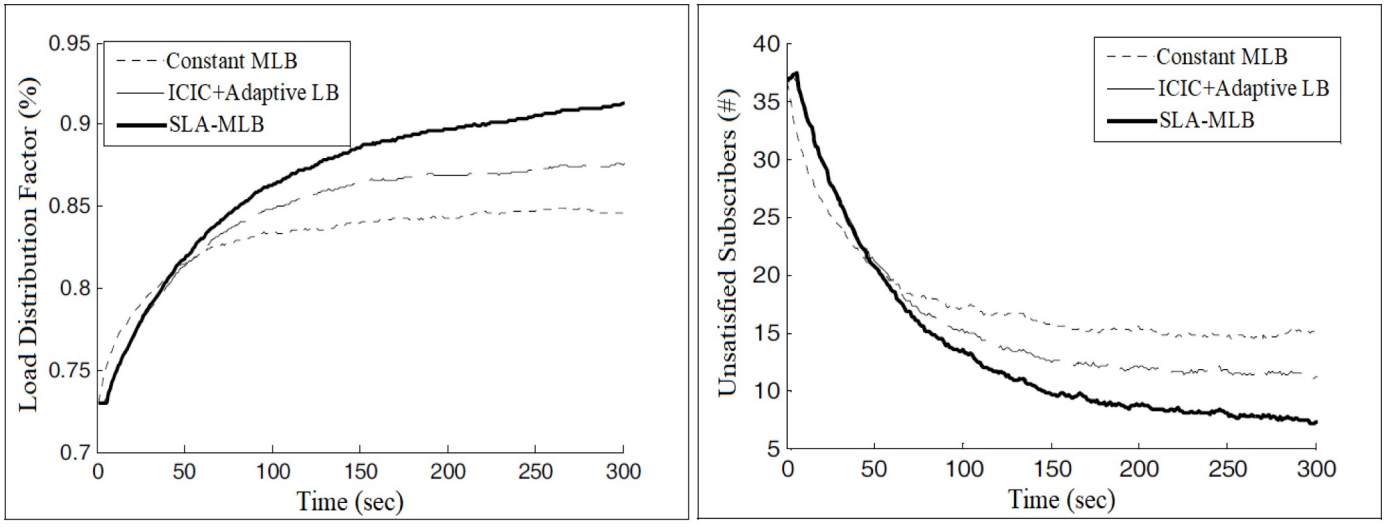

Fig. 5. Load distribution factor and the number of unsatisfied subscribers of three schemes with respect to time.

set the mobility and cell load sharing parameters for each special mobility pattern. when using the cooperative learning method, all base stations work based on learning a single-policy function. Such a dynamic approach is effective in various environments as exhibited by the proper KPI results achieved through the practical scenarios with subscribers with different and variable speeds.

In the learning automata load balancing, the agents are able to set the best value of the cell parameters required to decrease overload in various statuses. Mobility load balancing also learns the various cell individual offset adjustments which are needed for different load conditions. One other impact of load redistribution is increasing number of the satisfied subscribers from the perspective of data rate. In terms of convergence and complexity, it is proven that with similar costs, the algorithm has a linear order of complexity to the cell density. Also, there is always a trade-off between adaptation velocity and complexity.

5. Conclusion

In this paper we presented a Deep Learning-based Mobility Robustness Optimization solution (DRL-MRO), which learns the required parameter's appropriate values for each mobility pattern in individual cells. Optimal mobility setting for Handover parameters also depends on the user distribution and their velocities in the network. In this framework, an effective mobility-aware load balancing approach applied for autonomous methods of configuring the parameters in congruence to the mobility patterns in which approximately the same quality level is provided for each subscriber. We compared the proposed approach against the best static reference configuration (Ref) that is obtained by sweeping the parameter space. Our results show that DRL-MRO is able to learn parameter settings that achieve better performance than Ref in a realistic network environment where users have dynamically varying velocities. The results denoted that, although all SON functions apply a similar procedure, particular considerations are needed for each SON function based on its specific conditions. For example, each SON function needs a special strategy to recognize its action space. In practice, the higher performance achieved by the proposed method demonstrates the effectiveness of the self-organized cognitive approach. Moreover, stochastic learning provides an appropriate solution for developing self-optimization functions. It is noteworthy that the DRL-MRO is capable to set the mobility and cell load sharing parameters for each special mobility pattern.

Declaration of Competing Interest

None.

References

[1] Asghar Ahmad, Farooq Hasan, Imran Ali. Self-Healing in emerging cellular networks: review, challenges, and research directions. IEEE Communications Surveys & Tutorials 2018;20(3):1682–709.

[2] Lohmüller Simon. Cognitive Self-Organizing Network Management for Automated Configuration of Self-Optimization SON Functions. Universität Augsburg; 2019.

[3] Ganesan R, Sowmya B. Enhanced Fuzzy Rule with Modified Particle Swarm Optimization Based Handoff Algorithm in Wireless Mobile Communication Network. Asian Journal of Research in Social Sciences and Humanities 2016;6(12):15–29.

[4] Liu Qianyu, Kwong Chiew-Foong, Zhang Sibo, Li Lincan. Fuzzy-TOPSIS based optimal handover decision-making algorithm for fifth-generation of mobile communications system. Journal of Communications 2019;14(10):945–50.

[5] Singh Avinash, Singh Surya Pratap, Tripathi Upendra Nath, Mishra Manish. Optimizing Call Drops in Cellular Network using Artificial Intelligence based Handover Schema. intelligence 2017;6:1.

[6] Mahajan Payal. Review Paper on Optimization of Handover Parameter in Heterogeneous Networks. 2018 3rd International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH). IEEE; 2018. p. 1–5.

[7] Mohajer Amin, Barari Morteza, Zarrabi Houman. Big data based self-optimization networking: A novel approach beyond cognition. Intelligent Automation & Soft Computing 2017:1–7.

[8] Nikjoo Faramarz, Mirzaei Abbas, Mohajer Amin. A Novel Approach to Efficient Resource Allocation in NOMA Heterogeneous Networks: Multi-Criteria Green Resource Management. Applied Artificial Intelligence 2018;32(7-8):583–612.

[9] Zhang Jianchun, Zhao Yu, Ma Xiaobing. Reliability modeling methods for load-sharing k-out-of-n system subject to discrete external load. Reliability Engineering & System Safety 2020;193:106603.

[10] Phan NhuQuan, Bui ThiOanh, Jiang Huilin, Li Pei, Pan Zhiwen, Liu Nan. Coverage optimization of LTE networks based on antenna tilt adjusting considering network load. China Communications 2017;14(5):48–58.

[11] Troussas Christos, Virvou Maria. Advances in Social Networking-based Learning: Machine Learning-based User Modelling and Sentiment Analysis. Springer Nature 2020;181.

[12] Boutaba Raouf, Salahuddin Mohammad A, Limam Noura, Ayoubi Sara, Shahriar Nashid, Estrada-Solano Felipe, Caicedo Oscar M. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications 2018;9(1):16.

[13] Angus Ara, Murudkar Chetana V, Gitlin Richard D. "Machine Learning for QoE Prediction and Anomaly Detection in Self-Organizing Mobile Networking Systems. International Journal of Wireless & Mobile Networks (IJWMN) 2019;11.

[14] Zhang Lin, Tan Junjie, Liang Ying-Chang, Feng Gang, Niyato Dusit. "Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks. IEEE Transactions on Wireless Communications 2019;18(6):3281–94.

[15] Amiri Roohollah, Almasi Mojtaba Ahmadi, Andrews Jeffrey G, Mehrpouyan Hani. Reinforcement learning for self organization and power control of two-tier heterogeneous networks. IEEE Transactions on Wireless Communications 2019;18(8):3933–47.

[16] Mohajer Amin, Mazoochi Mojtaba, Niasar Freshteh Atri, Ghadikolayi Ali Azami, Nabipour Mohammad. "Network Coding-Based QoS and Security for Dynamic Interference-Limited Networks. International Conference on Computer Networks. Springer; 2013. p. 277–89.

[17] Neapolitan Richard E, Jiang Xia. Artificial intelligence: With an introduction to machine learning. Chapman and Hall/CRC; 2018.

[18] Liu Libin, Hodgins Jessica. "Learning to schedule control fragments for physics-based characters using deep q-learning. ACM Transactions on Graphics (TOG) 2017;36(3):1–14.

[19] Mohajer Amin, Barari Morteza, Zarrabi Houman. "QoSCM: QoS-aware Coded Multicast Approach for Wireless Networks. TIIS 2016;10(12):5191–211.

[20] Shah Brijesh, Dalwadi Gaurav, Pandey Anupkumar, Shah Hardip, Kothari Nikhil. "Online CQI-based optimization using k-means and machine learning approach under sparse system knowledge. International Journal of Communication Systems 2020;33(3):e4200.

[21] Kakadia Deepak, Ramirez-Marquez Jose Emmanuel. "Quantitative approaches for optimization of user experience based on network resilience for wireless service provider networks. Reliability Engineering & System Safety 2020;193:106606.

[22] Du Zhiyong, Sun Youming, Guo Weisi, Xu Yuhua, Wu Qihui, Zhang Jie. "Data-driven deployment and cooperative self-organization in ultra-dense small cell networks. IEEE Access 2018;6:22839–48.

[23] Hoseinitabatabei Seyed Amir, Mohamed Abdelrahim, Hassanpour Masoud, Tafazolli Rahim. "The Power of Mobility Prediction in Reducing Idle-State Signalling in Cellular Systems: A Revisit to 4G Mobility Management. IEEE Transactions on Wireless Communications 2020.

[24] Necker Marc C, Gauger Christoph M, Kiesel Sebastian, Reiser Ulrich. "Ikremulib: A library for seamless integration of simulation and emulation. 13th GI/ITG Conference-Measuring, Modelling and Evaluation of Computer and Communication Systems. VDE; 2006. p. 1–18.

[25] Tong Yanjie, Tien Iris. "Analytical probability propagation method for reliability analysis of general complex networks. Reliability Engineering & System Safety 2019;189:21–30.

[26] Zhang Haijun, Jiang Chunxiao, Hu Rose Qingyang, Qian Yi. "Self-organization in disaster-resilient heterogeneous small cell networks. IEEE Network 2016;30(2):116–21.

[27] Ahmed Furqan, Tirkkonen Olav. "Simulated annealing variants for self-organized resource allocation in small cell networks. Applied Soft Computing 2016;38:762–70.

[28] Misra Aradhana, Sarma Kandarpa Kumar. "Self-organization and optimization in heterogenous networks. Interference Mitigation and Energy Management in 5G Heterogeneous Cellular Networks. IGI Global; 2017. p. 246–68.

[29] Shayea Ibraheem, Ismail Mahamod, Nordin Rosdiadee, Mohamad Hafizal, Rahman Tharek Abd, Abdullah Nor Fadzilah. "Novel handover optimization with a coordinated contiguous carrier aggregation deployment scenario in LTE-advanced systems. Mobile Information Systems 2016:2016.

[30] Zakaria Eman, Awamry Amr A, Taman Abdelkerim, Zekry Abdelhalim. "A novel vertical handover algorithm based on Adaptive Neuro-Fuzzy Inference System (ANFIS). International Journal of Engineering & Technology 2018;7(1):74–8.

[31] Fedrizzi Riccardo, Goratti Leonardo, Rasheed Tinku, Kandeepan Sithamparanathan. "A heuristic approach to mobility robustness in 4G LTE public safety networks. 2016 IEEE Wireless Communications and Networking Conference. IEEE; 2016. p. 1–6.

[32] Gijón Carolina, Toril Matías, Luna-Ramírez Salvador, Marí-Altozano María Luisa. "A

data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks. IEEE Transactions on Vehicular Technology. 68. 2019. p. 9414–24.
[33] Anannya Mehrin, Shourov Riad Mashrub. "Performance Measurement Model of Mobile User Connectivity in Femtocell/Macrocell Networks using Fractional Frequency Re-use Scheme. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 2018;9(5):355–62.
[34] Bassoy Selcuk, Jaber Mona, Imran Muhammad Ali, Xiao Pei. "Load aware self-organising user-centric dynamic CoMP clustering for 5G networks. IEEE Access 2016;4:2895–906.
[35] Mohajer Amin, Mazoochi Mojtaba, Niasar Freshteh Atri, Ghadikolayi Ali Azami, Nabipour Mohammad. "Network Coding-Based QoS and Security for Dynamic Interference-Limited Networks. International Conference on Computer Networks. Springer; 2013. p. 277–89.
[36] Tuncel Nur Oyku, Koca Mutlu. "Joint mobility load balancing and inter-cell interference coordination for self-organizing OFDMA networks. 2015 IEEE 81st Vehicular Technology Conference (VTC Spring). IEEE; 2015. p. 1–5.
[37] Lobinger Andreas, Stefanski Szymon, Jansen Thomas, Balan Irina. "Load balancing in downlink LTE self-optimizing networks. 2010 IEEE 71st Vehicular Technology Conference. IEEE; 2010. p. 1–5.
[38] Du Zhiyong, Sun Youming, Guo Weisi, Xu Yuhua, Wu Qihui, Zhang Jie. "Data-driven deployment and cooperative self-organization in ultra-dense small cell networks. IEEE Access 2018;6:22839–48.

Amin Mohajer received the B.Eng. degree in electrical engineering in 2009, the M.S. Eng. degree in Telecommunications from Malek Ashtar University of Technology, Tehran, in 2011, and the Ph.D. degree in Telecommunications and Computer engineering from Malek Ashtar University of Technology, Tehran, in 2015. He is currently Research Fellow with the Telecommunication and Network Management Laboratory in the Department of Communications Technology at the ICT Research Institute (ITRC), Tehran, Iran. His research interests include intelligent resource management in wireless communication systems, mobile computing and application of robust optimization theory and recommender systems in self-organized wireless networks. His current works are more related to designing a Self-Optimization Networking system in Next Generation Mobile Networks using data analysis and artificial intelligence approaches.

Maryam Bavaghar received her B.Sc. degree from Birjand University, and her M.Sc. degree from Malek Ashtar University of Technology, Tehran, Iran, both in Information Technology and Information Security in 2010 and 2013, respectively. She is currently with the department of Network Security and Information Technology, ICT Research Institute (ITRC), Tehran, Iran. Her main research interests include Wireless Communication, Network Security, Intrusion Detection Systems, Data Analysis.

Hamid Farrokhi Since September 2002, he has been with the Communications Technology department of ICT Research Institute (ITRC), Tehran, Iran, where he became an associate professor on January 2006 and he is currently a full professor. His current research interests include data mining, distributed wireless networking, digital signal processing, wireless communications, and estimation and detection theories.