Data-Driven Optimization of Drone-Assisted Cellular Networks

T.R. Pijnappel*, J.L. van den Berg†‡§, S.C. Borst*, R. Litjens†¶
t.r.pijnappel@tue.nl, j.l.vandenberg@tno.nl, s.c.borst@tue.nl, remco.litjens@tno.nl
*Eindhoven University of Technology, Eindhoven, The Netherlands
†TNO, The Hague, The Netherlands
‡University of Twente, Enschede, The Netherlands
§CWI, Amsterdam, The Netherlands
¶Delft University of Technology, Delft, The Netherlands

***Abstract*—Drone base stations can help safeguard coverage and provide capacity relief when cellular networks are under stress. Examples of such stress scenarios are events with massive crowds or network outages. In this paper we focus on a disaster scenario with emergence of a traffic hotspot, where agile drone positioning and load management is a critical issue. In order to address this challenge, we propose and assess a data-driven algorithm which leverages real-time measurements to dynamically optimize the 3D position of the drone as well as a cell selection bias tuned for optimized load management. We compare the performance with three benchmark scenarios: i) no drone; ii) a drone positioned above the failing site; and iii) a drone with a statically optimized position and cell selection bias. The results demonstrate that the proposed algorithm significantly improves the call success rate and achieves close to optimal performance.**

***Index Terms*—Drone-assisted cellular networks, drone positioning, load management, performance assessment**

I. Introduction

In today's digital society wireless cellular networks are a critical infrastructure and need to be highly reliable. When adequately planned, these networks are able to provide reliable coverage and sufficient capacity in normal conditions. In atypical situations however the provided coverage and capacity may no longer be sufficient, e.g. in case of festivals or sports events with massive crowds or network anomalies due to failures or disasters. For these kinds of events the use of drone base stations offers an agile mechanism to provide capacity relief as well as safeguard coverage in the case of network failures [1].

The effective deployment of drone base stations in unpredictable scenarios involves several key challenges. In particular, it is crucial for the drone to find a good 3D position as this will have a critical impact on the performance. For example, a well-positioned drone can significantly increase the call success rate (CSR) [2], while a badly positioned drone may simply be ineffective or even cause high interference and therefore degrade performance. It is also critical to find a good policy or cell selection bias for assigning UEs to regular cells or to the drone base station in order to balance the loads. However, the optimal 3D position and selection bias of the drone cannot be easily determined as these depend on a variety of factors, such as the locations of the base stations, the positions of the UEs, antenna features and propagation characteristics. While some of these aspects may be known from RF planning procedures, other variables may be hard to predict with any accuracy, especially the positions of the UEs which could even be strongly affected by network disruptions. Hence in practice we will likely rely on data-driven algorithms, which leverage online measurement data or network statistics in order to determine an appropriate 3D position and cell selection bias for the drone.

A priori, it is unclear what algorithm to use to steer the drone operation in an online manner, and what the most suitable type of network measurement data is for that purpose. It would be natural to relate the measurement data to a performance indicator like the CSR, but it might simply take too long to accurately measure the CSR. Thus it might be better to consider other metrics as proxies for the CSR, or adopt load-related metrics with the intention to optimize the CSR through load balancing.

A. Contributions

We focus on a network disruption event with a failing base station and the emergence of a traffic hotspot. As alluded to above, it is quite natural for a network disruption event to cause a significant displacement or accumulation of UEs resulting in a traffic hotspot. As a way to restore service, we consider the deployment of a drone base station to take care of the UEs that would normally be served by the failing base station. Motivated by the above-mentioned challenges, we propose and analyze a data-driven algorithm to dynamically adjust the 3D position of the drone as well as a cell selection bias used for load management purposes. The dynamic adjustment procedure leverages measurement data that is easy to obtain in practice, henceforth referred to as control metrics.

The performance achieved by the proposed algorithm using different control metrics is compared with three reference scenarios i) no drone assistance, ii) a drone positioned above the failing site and iii) a drone deployed with a statically optimized position and cell selection bias. The comparison reveals a strong candidate optimization algorithm, which is

978-1-6654-2854-5/21/$31.00 ©2021 IEEE

subsequently further tested in a more extensive sensitivity analysis. The results show a significant improvement with respect to the scenarios without a drone and a drone positioned above the failing site, and indicate close to optimal performance.

B. Related literature

In recent years multiple papers have investigated the deployment of aerial base stations such as drones. In [3] the optimal altitude of a drone is determined for a given maximum allowed path loss. In [4] the authors use Q-learning to determine the optimal position of a drone base station and compare the drone-assisted network with a conventional terrestrial network. The authors of [5] propose a Q-learning algorithm for optimal drone positioning in an emergency scenario where the conventional network is completely destroyed. In a similar setting the authors of [6] use Q-learning to determine the best transmit power allocation and positioning of drones. The disadvantage of the proposed solutions of [5] and [6] is that in each iteration they determine the optimal positions for a snapshot of the environment, while in reality the environment is continuously changing. Therefore the proposed implementation could be too slow in finding good positions in a real scenario. In [7] the authors consider three algorithms (Q-learning, a gradient-based technique and a greedy search) to minimize the path loss. In [8] a heuristic scheme is proposed to find the minimum number of drones and their optimal 3D positions in a scenario without any regular base stations. The authors in [9] use a machine learning algorithm to detect peaks of user demands, and formulate an optimization problem to maximize coverage using drones. The authors of [10] study the placement of a tethered drone with the objective to minimize the average path loss between the drone and a ground user. In [11] an approach using multiple drones to form a communication 'bridge' is considered to offload excessive traffic demand to an underloaded base station.

In a previous paper [2] we determined the optimal position and cell selection bias for the drone by an exhaustive search of many possible configurations. This provides a good benchmark for the performance of online algorithms. However, in practice this would not be a feasible online approach because it would take too long to obtain reliable measurements. Also, it does not provide a viable offline method, since that would involve accurate knowledge of the propagation environment and in particular the spatio-temporal traffic characteristics, which is typically not available for unforeseen disruption events. On the other hand, the results provide useful insights for the development of online algorithms aiming to find a good 3D position and cell selection bias. In the present paper we therefore propose an algorithm which exploits measurements that can be performed more quickly in order to dynamically find a good 3D position and cell selection bias in an online manner.

C. Organization of the paper

The remainder of the paper is organized as follows. In Section II we explain the most important modeling aspects. In Section III we introduce an algorithmic framework and the control metrics that will be considered. In Section IV we present simulation results, and make a comparison of the proposed algorithm using different combinations of the control metrics, followed by a more extensive analysis of the algorithm with the best combination of control metrics. In Section V we summarize the key conclusions and provide recommendations for further research.

II. Modeling

In this section we elaborate on the most important modeling aspects, largely based on [2] to allow for comparison with static drones with an optimized position and cell selection bias. Therefore we repeat the key modeling aspects here in a condensed manner for the paper to be self-contained. A key difference with respect to [2] is that here we consider a drone that dynamically adjusts its position and cell selection bias. These dynamic adjustments affect the channel quality of the UEs, and therefore requires the modeling of additional aspects like handovers and dropping.

A. Network and antenna aspects

We consider a hexagonal layout of twelve three-sectorised sites comprising $12 \times 3 = 36$ cells. Each of these cells is assigned a single 5 MHz FDD carrier in the 2 GHz band, and is served by directional antennas as illustrated in Figure 1. A wraparound feature is applied to mimic an infinite-size network and avoid boundary effects. We assume that the antennas are located at a height of 30 m [12] and that the inter-site distance is 500 m corresponding to a dense urban environment.

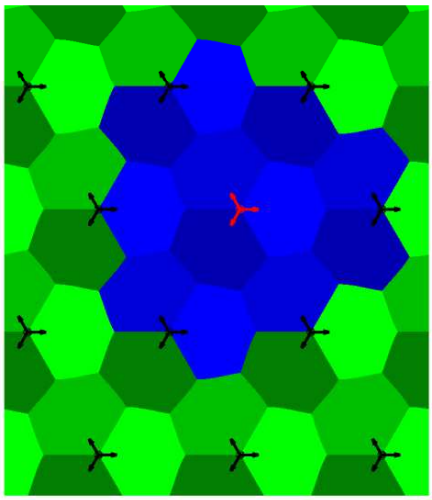

Fig. 1: Best-server area before a disaster event. The different colored areas indicate the cells of the regular base stations, the blue colored cells indicate the evaluation area and the red dot with arrows indicates the failing site.

For the antenna gains of *regular base stations* we use the model proposed by [13] where the total antenna gain is given by $G(\varphi,\theta) = G_h(\varphi) + G_v(\theta)$ with

$$G_h(\varphi) = -\min\left\{12\left(\frac{\varphi}{\text{HPBW}_h}\right)^2,\ \text{FBR}_h\right\} + G_r,$$

$$G_v(\theta) = \max\left\{-12\left(\frac{\theta-\theta_{\text{etilt}}}{\text{HPBW}_v}\right)^2,\ \text{SLL}_v\right\},$$

where $\text{HPBW}_{\{h,v\}}$ denotes the horizontal or vertical half-power beamwidth, G_r the maximum gain in dBi, FBR_h the front back ratio in dB, and SLL_v the side lobe level in dB, both relative to the maximum gain of the main beam. Furthermore φ denotes the horizontal angle relative to the azimuth direction, θ the negative elevation angle relative to the horizontal plane, and θ_{etilt} the electrical downtilt. We assume that these regular base stations each have a transmission power of $P_r^{\text{Tot}} = 20$ W where the power of the reference signal $P_r^{\text{RS}} = 1$ W.

For the *drone base station* we have rotated the model in [14, Table 7.3.1], and adapted the horizontal component to ensure a circular footprint, so that the antenna gain is modeled as

$$G(\theta) = -\min\left\{12\left(\frac{\theta}{\text{HPBW}_d}\right)^2,\ \text{SLL}_d,\ \text{FBR}_d\right\} + G_d,$$

with similar notation as before. We assume that the drone base station has a transmission power of $P_d^{\text{Tot}} = 0.5$ W where the power of the reference signal $P_d^{\text{RS}} = 0.025$ W. The drone is assumed to be wirelessly connected to the backhaul network using a transmission frequency other than that used to serve the UEs.

B. Propagation characteristics

To be able to model different kinds of urban scenarios, the ITU recommends three statistical parameters [15]:

- α: The ratio of built-up land area to the total land area.
- β: The number of buildings per square kilometer.
- γ: A scale parameter describing the buildings' heights according to a Rayleigh distribution.

We will consider a dense urban scenario, and therefore set $\alpha = 0.5$, $\beta = 300$ and $\gamma = 20$ [16].

For the link between UEs and *regular base stations*, we model the path loss according to the COST 231 Walfisch-Ikegami model [17]. We derive the parameters of this model using the statistical parameters α, β and γ. First the value of γ implies an average building height of $\gamma\sqrt{\pi/2}$. Following the reasoning in [16], the average width of the roads and the building separation are given by $1000/\sqrt{\beta} - 1000\sqrt{\alpha/\beta}$ and $1000/\sqrt{\beta}$, respectively. Lastly, we take the road orientation with respect to the direct radio path to be 90 degrees as suggested in [17].

As the COST 231 Walfisch-Ikegami model is only valid for transmitter heights up to 50 m, and we would like the drone to fly at higher altitudes, a different model is needed for the path loss between a UE and the *drone base station*. A model that is better suited for this purpose is the model described in [3], which considers two types of links, either having Line-of-Sight (LoS) or not (NLoS), with the LoS probability given by

$$p_{\text{LoS}} = 1 - p_{\text{NLoS}} = \frac{1}{1 + \xi\exp\left(-\psi[\arctan(h/r) - \xi]\right)},$$

where h denotes the difference in height, r represents the horizontal distance between the drone and UE, and $\xi = 12.081$ and $\psi = 0.1139$ are environment parameters calculated according to the model in [3], which also uses the statistical parameters α, β and γ. Now the path loss is assumed to be the free-space path loss plus the excessive path loss η which depends on the type of link (LoS or NLoS). We take $\eta_{\text{LoS}} = 1.6$ dB and $\eta_{\text{LoS}} = 23$ dB corresponding to a dense urban environment with a carrier frequency $f = 2000$ MHz as reported in [16]. Thus the path loss is given by

$$L = 20\log_{10}\left(\frac{4\pi df}{c}\right) + p_{\text{LoS}}\eta_{\text{LoS}} + (1 - p_{\text{LoS}})\eta_{\text{NLoS}},$$

where c denotes the speed of light and d the 3D-distance between the drone and UE.

We further impose a minimum coupling loss of 70 dB for all links [12].

C. Traffic characteristics

UEs are assumed to initiate calls according to a spatial Poisson process with intensity λ. The occurrence of the disruptive event could result in an increased intensity of active UEs in an area close to the location of this event. We model such a hotspot as a circle with a radius of 100 meters, and assume that the intensity of initiated calls in the hotspot is a factor ρ times higher than elsewhere.

The calls have an exponentially distributed duration with a mean equal to τ seconds, and are characterized by a minimum bit rate requirement of R Mb/s.

D. Resource management aspects

Besides considering the Reference Signal Received Power (RSRP), we adopt the Cell Individual Offset (CIO) as a cell selection bias to be able to steer traffic away from the drone when it has a high load or to attract more load when it has little load. Specifically, upon initiation of a call we assign the cell with the highest RSRP + CIO as the call's serving cell, provided that the UE initiating the call has an RSRP > -120 dBm meaning that it has coverage. If the latter is the case, an admission control mechanism accepts the newly requested call when the estimated fraction of downlink resource blocks needed at the serving cell to satisfy the minimum bit rate requirements of all currently active UEs plus the new UE does not exceed 98%. Here we leave a 2% margin to cope with possible changes in the amount of interference due to the movement of the drone. To guarantee the minimum required bit rate of R Mb/s, we calculate the fraction of downlink resources that a UE needs by $R/\left(B\log_2(1 + \text{SINR})\right)$, where B denotes the available bandwidth which is set to be 5 MHz and SINR the Signal-to-Interference-plus-Noise Ratio of that

UE. For the calculation of the SINR, we assume a thermal noise of -106.94 dBm and a noise figure of 8 dB for each UE.

Any surplus resources not needed to satisfy the minimum bit rate requirements are divided among the active UEs in a proportional fair way.

Although we do not consider user mobility, the dynamic optimization of the drone position and the CIO may cause active UEs to request handovers. Such a handover is attempted when the RSRP + CIO of a candidate target cell exceeds that of the current serving cell by at least 3 dB (hysteresis value) for at least 200 ms (time to trigger or TTT). A handover request is submitted to the admission control algorithm of the target cell (to avoid overloading the target cell) where we do not consider the margin of 2%. When a handover request is denied, it will be repeated after 50 ms provided that the conditions triggering the handover are still satisfied.

Lastly, changes in the 3D drone position may affect whether active calls still experience coverage, as specified by the -120 dBm RSRP threshold. In case an active call looses coverage and cannot be handed over to another covering cell, it is dropped.

E. Performance measure

We adopt the Call Success Rate (CSR) as a key performance measure. For this metric we consider a call to be successful if it has coverage, is admitted, has received its required minimum bit rate for the entire call duration and is not dropped. For the evaluation area of the CSR we consider the cells of the failing site and the nine cells adjacent to this site (indicated in blue in Figure 1). This way, we only consider UEs in the vicinity of the failed based station, and UEs connected to one of the cells that now handles calls that would normally be handled by the failing site.

III. Algorithms

In this section we introduce the general framework and control metrics (CMs) of the proposed algorithm. In order to optimize the CSR, this data-driven algorithm uses the CMs to dynamically adjust the x, y and z coordinates and CIO as the control parameters (CPs) of the drone.

A. General framework

For our algorithm we combine the optimization of z and CIO. This design choice is motivated by the observation that the optimal z and CIO for a static drone are strongly related. In particular, since increasing z or CIO both have the effect of steering more load towards the drone, therefore a higher optimal value of z corresponds to a lower optimal CIO value and vice versa [2]. Given this strong relation, we propose an algorithm which tries to keep the z or CIO at a default value until the other CP has reached a pre-determined lower or upper limit. Note that this yields two distinct options in terms of whether z or CIO is the primary CP.

The algorithm repeatedly optimizes the x, y and z/CIO one at a time. In this algorithm we measure two distinct CMs: one for xy-adjustments and one for z/CIO-adjustments, which will be defined in Section III-B. The algorithm starts by measuring the CMs during 200 ms. Based on these CMs and the previous action taken, our algorithm decides what action to take next (increase/decrease x, y or z/CIO). Then the action is executed which means that the drone moves to a new position or that the CIO is adjusted. Now we wait until the TTT has passed, to allow for possible handovers induced by the control action, and then measure the CMs again. This iterative cycle of measuring, adjusting the CP and waiting is illustrated in Figure 2.

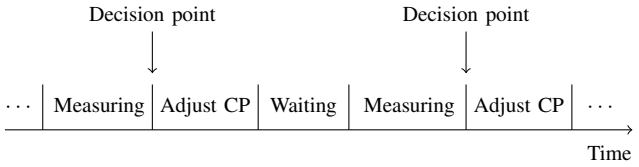

Fig. 2: Illustration of the optimization process and its decision points.

As mentioned above, we repeatedly optimize the CPs x, y and z/CIO, meaning that we adjust only one of the four CPs at a time. For the algorithm we consider two optimization methods. The first method is an *improvement*-based method which can be used for all CPs (x, y and z/CIO), and works as follows: Given a baseline measurement of a CM, we first take an action to increase the current CP. At the next decision point (after the next measurement period), we check whether the CM has improved. If that is the case, we keep increasing the CP until the CM does not improve anymore. If the CM has degraded, we reverse the CP adaptation direction and hence in this case decrease the CP, and continue taking this action until the CM does not improve anymore. At that point, we start optimizing one of the other CPs (in the order x, y, z/CIO).

Since a change in the CIO does not necessarily affect the currently active UEs, we cannot conclusively infer from the measurements of the CMs whether or not the adjustment was an improvement. Therefore we also consider a different approach for the optimization of z/CIO, which is a *value*-based method. In this method we measure the CM, and determine whether the CP should increase or decrease based on the CM value itself, rather than on whether it improved or degraded. So depending on the value of the CM, we will either increase or decrease the z/CIO in order to direct additional or less load to the drone cell or do not adjust z/CIO. For this method we also limit the number of actions, meaning that we allow at most one adjustment in z/CIO every minute. The reason for this is that an adjustment of the CIO does not affect most active UEs, except UEs that request a handover due to this change. As only a few (if any) UEs request a handover, we choose to incorporate the impact of new and finished calls on the CM, and therefore need to wait some time before adjusting the z/CIO again.

B. Control metrics

As our ultimate goal is to optimize the CSR, it might seem logical to use the CSR as CM. However, it would take too long

to measure the CSR accurately enough due to the relatively large time scale at which call arrivals and terminations take place, and hence making the adjustments based on the CSR would likely be too slow. Therefore, the adjustments are driven by other specific CMs, which are easy to obtain and can be measured well more quickly.

For the improvement-based optimization method we consider the following CMs which are measured over the cells of the evaluation area:

- C_1: the total amount of resources needed to satisfy the minimum bit rate requirement of all currently assigned UEs to a cell.
- C_2: the average fraction of resources needed per UE to satisfy the minimum bit rate requirement.
- C_3: the absolute difference of the 75th percentile of the fraction of resources needed to satisfy the minimum bit rate requirements of the currently assigned UEs to one of the cells adjacent to the failing site and the fraction of resources that the drone needs to satisfy the minimum bit rate requirements of the connected UEs.
- C_4: same as C_3 but now using the 50th percentile (median).
- C_5: the total number of currently assigned UEs that do not receive their minimum bit rate.
- C_6: the total number of currently assigned UEs that do not receive their minimum bit rate divided by the total number of assigned UEs.

CMs C_1-C_4 are chosen with the intention to balance the load, while C_5 and C_6 are more closely related to the CSR as these measure the number/fraction of currently successful calls.

For the value-based optimization method (which is only used for adjusting z/CIO) we consider the following CMs:

- C_7: the difference of the 75th percentile of the smoothed fraction of resources needed to satisfy the minimum bit rate requirements of the cells of the regular cells and the smoothed fraction of resources that the drone needs to satisfy the minimum bit rate requirements.
- C_8: same as C_7 but now with the 50th percentile (median).

For these last CMs we use a smoothing parameter such that 70% of the measured value is based on the last minute.

IV. Simulation setup and results

In this section we first discuss the setup of the simulation experiments. Then we quantify how well (in terms of the CSR) different combinations of the CMs (one for the xy-adjustments, and one for the z/CIO-adjustments) work for our algorithm, and use that to select a good combination of CMs. Finally, we conduct a more extensive analysis for the algorithm with this selected combination of CMs. Specifically, we examine the convergence speed and compare the performance with baseline scenarios for different hotspot locations and traffic loads.

A. Setup of simulation experiments

Table I provides a list of simulation parameters that have not been specified before. Based on regulatory restrictions the maximum altitude of the drone is set at 120 m [18]. For the movement of the drone we assume a maximum speed of 24 km/h. Furthermore, we set the default CIO and altitude equal to 0 dB and 120 m, respectively, and assume the drone to begin above the failing site with these default settings.

In the simulation experiments we consider a time period of more than 2.5 hours and measure the CSR over the last hour. The measured CSR will serve as a basis for comparing the performance achieved by the different configurations of the algorithm. In addition, we have investigated the evolution of the CSR averaged over a moving two-minute window starting when the drone is activated. Inspection of these results showed that the average CSR over the last hour was generally approached well within the first 1.5 hours, indicating that an equilibrium situation had thus been reached at the start of the CSR assessment period. Obviously, conditional on good CSR performance in equilibrium, it remains crucially important to reach that service level as quickly as possible, and we will further examine the convergence speed of the deemed best configuration of the algorithm in Section IV-C.

TABLE I: Simulation parameters.

	General parameters	
h_{UE}	Height of UE	1.5 m
λ	Arrival intensity (outside the hotspot)	2.1383 arrivals/s/km^2
τ	Average call duration	120 s
R	Minimum bit rate	0.4 Mb/s
	Antenna gain - link with regular BS	
G_r	Maximum antenna gain	18 dBi
HPBW$_h$	Half-power beam width (horizontal)	65°
HPBW$_v$	Half-power beam width (vertical)	6.2°
FBR$_h$	Front back ratio	30 dB
SLL$_v$	Side lobe level	-18 dB
θ_{etilt}	Electrical downtilt	8°
	Antenna gain - link with drone BS	
G_d	Maximum antenna gain	8 dBi
HPBW$_d$	Half-power beam width	65°
FBR$_d$	Front-back ratio	30 dB
SLL$_d$	Side lobe level	30 dB

B. Selection of best algorithm configuration

To evaluate the performance of the algorithm using different CM combinations, we assumed that the CMs used for x and y-adjustments are the same, but could be different for the z/CIO-adjustments. This implies a total of $6 \times 8 \times 2 = 96$ (six for xy-adjustments, eight for z/CIO-adjustments, and the two choices z or CIO as the primary optimization variable for z/CIO-adjustments) combinations of CMs and choices for the primary optimization variable of z/CIO. Moreover, we also consider the six possible CMs for the xy-adjustments where we perform no adjustments of the z/CIO (indicated by 'None'). Together with these configurations, we also consider three

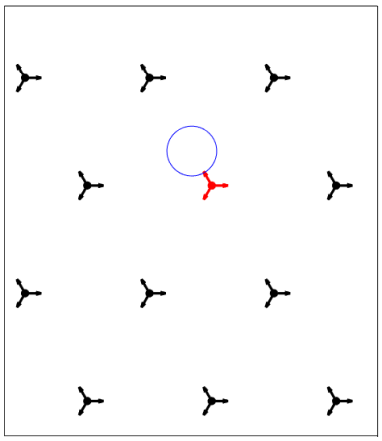

Fig. 3: Illustration of simulation scenario. The black dots indicate the regular base stations (with their azimuth directions indicated by the arrows), the red dot with arrows indicates the failing site, and the blue circle indicates the hotspot area.

benchmarks i) no drone, ii) a static drone above the failing site, and iii) a static drone with optimal values for x, y, z and CIO obtained by simulating many possible configurations).

Let us first consider a scenario where the load in the hotspot is $\rho = 4$ times higher than elsewhere, and where the center of the hotspot is located 160 m from the failing site along one of its azimuth directions as illustrated in Figure 3. For each of the combinations of CMs we conduct twenty independent simulation runs and report the average CSR in Table II.

Each entry in this table indicates the CSR of one configuration of the algorithm. Here each row indicates a CM for the z/CIO-adjustments and each column a CM for the xy-adjustments. We see that C_3 and C_4 for xy-adjustments perform worst among all CMs for the xy-adjustments, while C_6 yields the best performance in almost all cases for each z/CIO-control metric. But more interestingly, for xy-adjustments based on CMs C_1, C_2, C_5 and C_6, we notice that the improvement-based method for z/CIO (z/CIO-control metrics C_1 to C_6) is outperformed by the configuration without z/CIO-adjustments. The latter is outperformed in most cases by C_7 and C_8. A possible explanation for this is that it is better to incorporate some information about new and finished calls in the CM before adjusting the z/CIO again. This is because the drone would ideally be configured to fly at an altitude above 120 m. Since this is not allowed, we mostly adjust the CIO which has no direct consequences for the SINRs and thus resource usage of the active UEs except for a few possible handovers, but does have consequences for new calls. Moreover, we also simulated C_7 and C_8 for z/CIO-adjustments where we allowed an adjustment every 10 and 30 seconds (instead of 60) and found worse performance. This indicates that we should not adjust the z/CIO too often to better incorporate the effect of new and finished calls on the CM.

Even though the differences in Table II are small, it seems best to take C_4 for the xy-adjustments, which is noted to be most closely related to the CSR. For the z/CIO-adjustments we select C_8-CIO as this performs best in combination with C_4 for the xy-adjustments. In the remainder of the paper we focus on this algorithm configuration and the three benchmarks.

TABLE II: CSR of three benchmarks and the algorithm using different combinations of CMs, where the appended 'CIO' or 'z' label indicates the assumed primary control variable in the z/CIO plane.

Control metric for z/CIO \ Control metric for x,y	C_1	C_2	C_3	C_4	C_5	C_6
None	0.928	0.929	0.914	0.931	0.929	0.931
C_1-CIO	0.865	0.867	0.827	0.828	0.922	0.921
C_2-CIO	0.870	0.867	0.825	0.817	0.923	0.923
C_3-CIO	0.895	0.894	0.827	0.859	0.922	0.925
C_4-CIO	0.892	0.915	0.848	0.842	0.922	0.926
C_5-CIO	0.876	0.870	0.827	0.819	0.921	0.923
C_6-CIO	0.898	0.914	0.859	0.907	0.924	0.927
C_7-CIO	0.931	0.932	0.825	0.936	0.936	0.937
C_8-CIO	0.924	0.930	0.823	0.832	0.939	0.940
C_1-z	0.919	0.922	0.917	0.922	0.920	0.925
C_2-z	0.921	0.923	0.916	0.922	0.920	0.922
C_3-z	0.913	0.913	0.829	0.839	0.922	0.924
C_4-z	0.918	0.921	0.919	0.922	0.920	0.924
C_5-z	0.916	0.919	0.912	0.922	0.920	0.920
C_6-z	0.921	0.922	0.922	0.923	0.920	0.923
C_7-z	0.929	0.931	0.826	0.929	0.928	0.932
C_8-z	0.925	0.927	0.823	0.870	0.926	0.923

Benchmarks:	
No drone	0.820
Static at failing site	0.898
Static at optimal position	0.940

C. Convergence speed

In the event of a network disruption it is crucial that the impact on the service level is minimized, meaning that it is important to find a good position and CIO for the drone as quickly as possible. In order to examine the convergence speed of our candidate algorithm, we consider three load intensities $\rho \in \{2, 4, 8\}$ where the center of the hotspot is located at 40 or 160 m from the failing site along one of its azimuth directions. For these scenarios we simulated 50 independent runs and determined the CSR over two-minute windows. In Figure 4 we plot how the window-averaged CSR evolves over time. We observe that it takes more time for the CSR to approach the long-term average for higher values of ρ (meaning a higher load in the hotspot). However, in all cases we see that after approximately twenty minutes the CSRs are close to the equilibrium values. This suggests that our algorithm is able to find a good position and CIO in twenty minutes or less.

D. Comparison for different hotspot and traffic scenarios

As the algorithm should be able to consistently find a good 3D position and CIO for the drone in a wide range

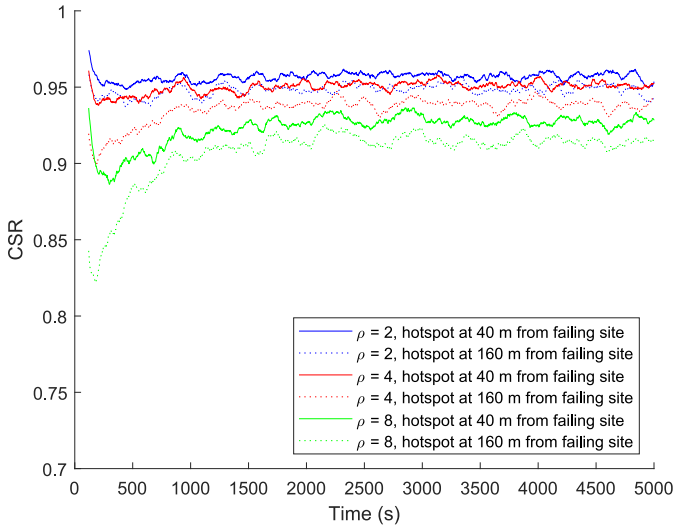

Fig. 4: Evolution of the CSR over time for different scenarios, from the moment that the drone becomes active.

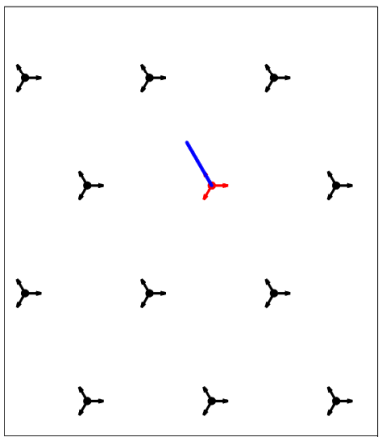

Fig. 5: Illustration of simulation scenarios. The black dots indicate the regular base stations, the red dot indicates the failing site and the blue line indicates the possible locations of the center of the hotspot area.

of network environments, we consider different locations of the hotspot and different traffic intensities in the hotspot. For this sensitivity analysis we consider locations where the center of the hotspot lies along one of the azimuth directions of the failed site (the blue line in Figure 5), and we consider $\rho \in \{1, 2, 4, 8\}$.

In Figure 6 we observe that the attained CSR for the benchmark without a drone (dash-dotted lines) increases as the distance between the hotspot and the failing site increases. This can be explained by the fact that the UEs in the hotspot are closer to the adjacent cells. Therefore they experience better signal strengths from these cells which means that the UEs need fewer resources when connected to these cells. This in turn leaves more resources for other UEs, thus reducing the number of blocked/dropped calls.

Furthermore, we observe that the benchmark without a drone performs worst, which is as expected. Comparing the

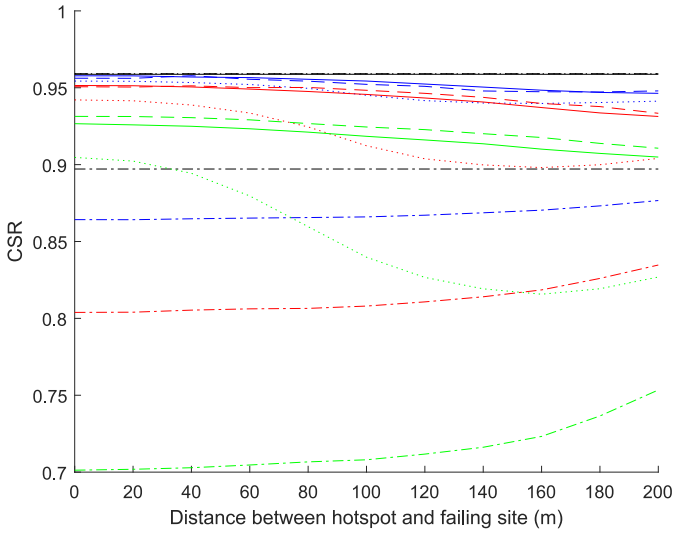

Fig. 6: CSR for different scenarios. The black, blue, red and green lines correspond to the different values of $\rho \in \{1, 2, 4, 8\}$ respectively. The solid, dashed, dotted and dash-dotted lines correspond to the proposed algorithm, optimal static drone, static drone above failing site and no drone respectively.

proposed algorithm using the selected CMs, we see that it improves the CSR with at least 0.06, 0.06, 0.09 and 0.15 compared to the benchmark without a drone for $\rho \in \{1, 2, 4, 8\}$ respectively. Moreover, our algorithm nearly achieves the performance of the static drone with optimized position and CIO.

In order to take a closer look at the improvement with respect to the fraction of unsuccessful calls, we can define the improvement factor as $(1-\text{CSR}_{\text{benchmark}})/(1-\text{CSR}_{\text{algorithm}})$. This ratio represents the number of unsuccessful calls of the benchmark for each unsuccessful call using the proposed algorithm. Plotting the improvement factor for the benchmark without a drone and with a static drone at the failing site at an altitude of 120 m and with a CIO of 0 dB yields Figures 7 and 8, respectively.

In Figure 7 we observe that the fraction of failed calls is more than twice as high in a scenario without a drone compared to a scenario where the proposed algorithm is used.

When we look in Figure 8 at the improvement factor when comparing our algorithm with the static drone, we see that this value depends on the location of the hotspot and the intensity of initiated calls in the hotspot. When the center of the hotspot is located at the failing site (distance 0) and $\rho = 8$, we see that the benchmark has still 30% more unsuccessful calls, due to the fact that using our algorithm the CIO is adjusted while the benchmark has a fixed CIO.

Furthermore, we observe that the improvement factor attains a maximum value when the hotspot is located between 110 and 150 m from the failing site. The explanation for this is that the performance of the benchmark (static drone above the failing site) degrades as the hotspot is located further away

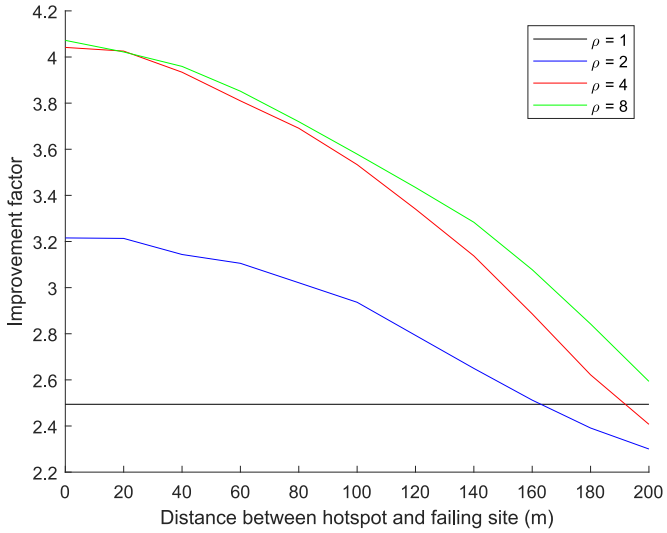

Fig. 7: Improvement factor of our algorithm compared to a scenario without a drone.

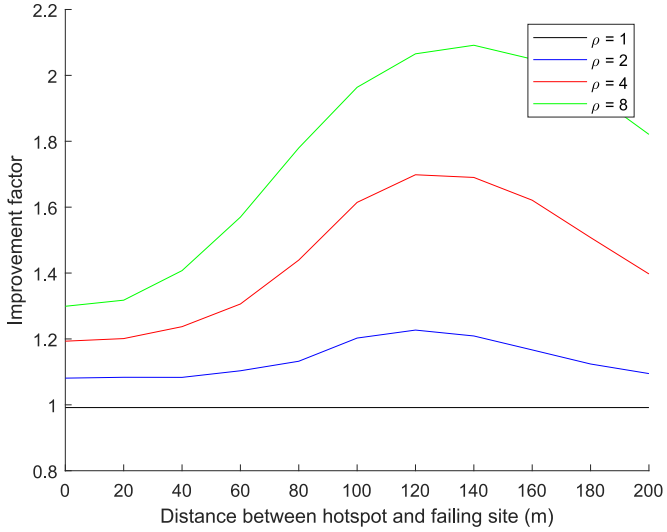

Fig. 8: Improvement factor of our algorithm compared to a scenario with a static drone above the failing site at an altitude of 120 m and with a CIO of 0 dB.

from the failing site. However, at approximately 150 m, the performance starts improving. This in turn is caused by the fact that the regular base stations will be able to serve the UEs in the hotspot more efficiently, thus reducing the number of unsuccessful calls.

V. Conclusion

We have designed and analyzed a data-driven algorithm to dynamically adjust the 3D position and cell selection bias of a drone base station in a scenario with a failing site and emerging hotspot. The proposed algorithm does not require any explicit knowledge of the antenna features, propagation characteristics, location of the hotspot or traffic density, and leverages easily available measurement data in order to steer the drone operation in an online manner. While the CSR serves as the ultimate optimization objective, other metrics which are easily measurable turn out to provide a more effective basis for driving the dynamic adjustments towards CSR optimization. Remarkably, with the right selection of control metrics and associated time scales, the proposed algorithm is able to quickly reach a stable and near-optimal operating point. Indeed, the performance in terms of CSR is comparable to the performance of a static offline optimization with full knowledge of the relevant system attributes. In future research we plan to expand the analysis of the convergence speed, and also consider the tracking capability of the proposed algorithm in situations where the location of the hotspot and traffic density may change over time. Another topic that we would like to investigate in future research is the optimal control of multiple drones.

References

[1] 5G!Drones, EU H2020 project, www.5gdrones.eu, 2019-2022.
[2] T.R. Pijnappel, J.L. van den Berg, S.C. Borst and R. Litjens, "Drone-assisted cellular networks: optimal positioning and load management", *2021 IEEE VTC-Spring*, 2021.
[3] A. Al-Hourani, S. Kandeepan and S. Lardner, "Optimal LAP altitude for maximum coverage", in *IEEE Wireless Communications Letters*, vol. 3, no. 6, 2014.
[4] R. Ghanavi, E. Kalantari, M. Sabbaghian, H. Yanikomeroglu and A. Yongacoglu, "Efficient 3D aerial base station placement considering users mobility by reinforcement learning", *2018 IEEE WCNC*, Barcelona, Spain, 2018.
[5] P.V. Klaine, J.P.B. Nadas, R.D. Souza and M.A. Imran "Distributed drone base station positioning for emergency cellular networks using reinforcement learning", *Cognitive Computing* vol. 10, 2018.
[6] R. de Paula Parisotto, P.V. Klaine, J.P.B. Nadas, R.D. Souza, G. Brante and M.A. Imran, "Drone base station positioning and power allocation using reinforcement learning", *2019 ISWCS*, Oulu, Finland, 2019.
[7] D. Prado, S. Inca, D. Martín-Sacristán and J.F. Monserrat, "Comparison of optimization methods for aerial base station placement with users mobility", *2019 EuCNC*, Valencia, Spain, 2019.
[8] E. Kalantari, H. Yanikomeroglu and A. Yongacoglu, "On the number and 3D placement of drone base stations in wireless cellular networks", *2016 IEEE VTC-Fall*, Montreal, Canada, 2016.
[9] C. Boucetta, A. Dridi, H. Moungla, H. Afifi and A.E. Kamal, "Optimizing drone deployment for cellular communication coverage during crowded event",*2019 MILCOM*, 2019.
[10] M.A. Kishk, A. Bader and M.-S. Alouini, "On the 3-D placement of airborne base stations using tethered UAVs", in *IEEE Transactions on Communications*, vol. 68, no. 8, 2020.
[11] N. Tafintsev, D. Moltchanov, S. Andreev, S. Yeh, N. Himayat, Y. Koucheryavy and M. Valkama, "Handling spontaneous traffic variations in 5G+ via offloading onto mmWave-capable UAV 'bridges'", in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, 2020.
[12] 3GPP TR36.942: "Evolved universal terrestrial radio access (E-UTRA); radio frequency (RF) system scenarios", *3GPP Technical Report*, v16.0.0, 2020.
[13] F. Gunnarsson, M.N. Johansson, A. Furuskär, M. Lundevall, A. Simonsson, C. Tidestav and M. Blomgren, "Downtilted base station antennas - a simulation model proposal and impact on HSPA and LTE performance", *2008 IEEE VTC-Fall*, Calgary, Canada, 2008.
[14] 3GPP TR38.901: "Study on channel model for frequencies from 0.5 to 100 GHz", *3GPP Technical Report*, v16.1.0, 2019.
[15] ITU-R, Rec. P1410-2 "Propagation data and prediction methods for the design of terrestrial broadband millimetric radio access systems." P Series, *Radiowave Propagation*, 2003.
[16] A. Al-Hourani, S. Kandeepan and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," *2014 IEEE GLOBECOM*, Austin, USA, 2014.
[17] COST 231, Final report: "*Digital mobile radio towards future generation systems*", 1999.
[18] The European Commission, Official journal of the European Union: "*L152*", vol. 62, June 11, 2019.