Abstract—In this paper, the authors present an algorithm for determining the location of wireless network small cells in a dense urban environment. This algorithm uses machine learning, such as k-means clustering and spectral clustering, as well as a very accurate propagation channel created using the ray tracing method. The authors compared two approaches to the small cell location selection process— one based on the assumption that end terminals may be arbitrarily assigned to stations, and the other assuming that the assignment is based on the received signal power. The mean bitrate values are derived for comparing different scenarios. The results show an improvement compared with the baseline results. This paper concludes that machine learning algorithms may be useful in terms of small cell location selection and also for allocating users to small cell base stations.

Keywords—base station selection, k-means clustering, spectral clustering, user equipment allocation.

1. Introduction

With the advent of 5G networks, one may notice increasing interest in the concept of small cells. Additional small cells positioned at locations where services are already available may significantly improve network performance and may boost the quality of service, depending on user needs. For example, deterioration in the quality of network access may often be observed in large gatherings, as most of people present in such scenarios use wireless devices. Such a group of devices connects to the base station and, consequently, either this group or other users of this particular base station are capable of obtaining satisfactory bitrates or service quality levels. An additional base station with a small coverage area (known as a small cell or a pico cell) positioned at the location where such a large group of devices is present may significantly improve the quality of service enjoyed by all users. While the use of small cells is justified in the aforementioned scenario, it is not quite obvious where exactly such cells should be located.

Sometimes, it is quite easy to determine where and for how long increased traffic rates may be expected. For example, a group of people actively using their mobile devices may be presented at a given location only for random periods of time only, or may be expected there periodically (bus stations, airports, etc). The above-mentioned scenarios are directly related to the location at which the increase in traffic takes place. For instance, if increased network traffic is observed at a bus stop—we know the location of the potential small cell base station. However, increased network traffic is not always closely related to a fixed location. Therefore, the authors have designed an algorithm that determines small cell installation locations with a given period of time, to match the highest demand levels. The results obtained with the use of this algorithm may be relied upon in many ways. It is possible to average the results (or to select critical, worst case scenarios), thus selecting a location for a stationary small cell. Such an approach may be used in network coverage planning or improvement processes.

Another approach consists in using drones (UAVs) with a small cell base station hovering overhead. In this scenario, the position of such stations may be changed dynamically. The algorithm presented in this paper works regardless of the way the results are used.

In other publications concerning the application of machine learning techniques for handling small cell traffic two main aspects seem to prevail, namely assignment of user equipment (UE) to a given set of base stations (BSs) and positioning of BSs for best coverage. The articles dealing with the former of those aspects include [1]–[4]. Balapuwaduge et al. [1] focus on smarter assignment of UE to BS by employing an ML algorithm based on the hidden Markov model. The algorithm focuses on reliability and availability of network resources in order to select the best BS for a given piece of UE.

Yang et al. [2] employ reinforcement deep learning (DL) to position small cells in indoor scenarios, with a particular emphasis placed on company small cells. The problem presented may be generalized to the allocation of users whose behavior is predictable and those who behave in a more dynamic manner. The ML algorithm works based on data consisting only of allocation information for each piece of UE. Qi et al. [3] and Xu et al. [4] focus on the k-means clustering and the reinforcement k-means clustering algorithm, respectively. Both of those papers use ML for clustering UEs in order to achieve good load balance.

In the second group of papers which focus on BS positioning in order to achieve the best coverage, the use of drones is a popular solution [5], [6]. In [5], drones are to replace BSs in the event of an emergency. The main problem is
how to ensure the best possible coverage. The reinforce-
ment learning approach, namely the Q-learning algorithm,
is employed to determine the drones’ positions based on
whether a connection has been established between UE
and the drone or not. Wang et al. [6] focus on prob-
lems that drones face while ensuring connectivity, namely
c o-channel interference, limited battery capacity and fast
topology changes. In this case, ML is supposed to control
not only the placement of drones, but also their transmis-
sion power, as it affects the level of interference and battery
lifetime.

To recapitulate, this paper offers the following contribu-
tions:

- it combines the problem of allocating UE to BSs with
  the problem of choosing the stations’ locations,
- it uses two simple unsupervised ML algorithms,
  namely k-means clustering and spectral clustering,
  in order to group UE on the basis of path loss data,
- it chooses the best BS location for each of the groups,
  in order to improve QoS and mean bitrate of the
connections.

In the chapters below, the following are described: the
proposed ML-based small cell deployment algorithm (Sec-
tion 2), the system in which the simulations were performed
(Section 3), detailed simulation results with conclusions
(Section 4), and summary of the work performed.

2. ML-based Small Cell Deployment

The first thing one needs to do in order to successfully
deploy small (pico) cells is to choose their optimized loca-
tions. In this paper, we propose the use of machine learning
algorithms for this purpose. Such an algorithm will decide
which small cells to use and which pieces of UE to as-
sign to them. The main problem is what algorithm to use,
considering the limitations of training data. In this section,
different approaches to artificial learning are discussed and
the best solutions are presented. We also describe how we
employ the chosen ML algorithms for selecting BSs and
assigning UE. Additionally, a detailed description of the
input data that the presented algorithms rely on is given as
well.

When it comes to selecting the right ML algorithm, one
has to consider what types of data are available. In most
cases, it is hard to obtain a labeled set of training data.
Labeled data is a term used to describe data that consists
of input features (usually referred as X labels), but also of
their corresponding categories, or desired outputs, known
as y labels. In order to obtain such a data set, it is usually
necessary to manually label each input feature set. In the
case of a computer simulation, it is much easier to generate
training data along with their corresponding labels, but this
is not always true.

In the system considered in this paper, labeled data would
be generated for a set of many different combinations of
user positions within the considered space. The input fea-
ture data could consist of the users’ coordinates and other
additional features, while output labels would indicate to
which BS they are connected. It would be necessary to
calculate, for all of the user locations considered, all bi-
trates to all of the possible BSs, while taking into ac-
count interference from all other BSs in order to determine
how to allocate users to BSs. It is easy to imagine how
computationally-intensive and time-consuming it would be
to generate such a dataset. In order to address these issues,
the authors propose to use ML algorithms that are not su-
ervised and are able to learn based on data without any
specified output labels. The algorithms that are explored in
this paper are: k-means clustering and spectral clustering.

2.1. K-means Clustering

As the system under consideration consists of scattered
users in who are in need of being allocated to a BS, clus-
tering algorithms come to mind first. Clustering algorithms
groups similar feature data points together. The resulting
groups are called clusters. In this paper, the k-means al-
gorithm has been proposed as a grouping method, as it is
simple, yet effective.

The grouping process is performed in the following man-
ner: initially, a random placement of centroids is picked
(points around which clusters are centered). Then, all in-
put instances are assigned to the closest centroid [7]. Then,
the centroids are updated by minimizing the inertia crite-
non IC, given by:

\[
IC = \sum_{n=0}^{N} \min_{c \in C} \left( ||x_n - c_i|| \right), \tag{1}
\]

where \(x_n\) is an instance from input dataset \(X\), and \(c_n\) is the
\(n\)-th centroid from the chosen centroid set \(C\) consisting of
\(N\) centroids. The process of categorizing input data and
assigning such data to clusters is repeated until the cen-
troids stop moving. In the k-means algorithm, it is initially
necessary to specify the number of clusters.

2.2. Spectral Clustering

Spectral clustering is another unsupervised grouping ML
algorithm used in the experiments. Compared to the k-
means algorithm, spectral clustering is capable of perform-
ing better on non-convex data, which is quite helpful in
solving the problem presented in the paper. Spectral clus-
tering creates a similarity matrix between the input data and
then reduces the dimensionality of this matrix. After that,
another clustering algorithm is used on the obtained ma-
trix [8]. In the algorithm implemented for the experiments,
spectral clustering performs k-means after dimensionality
reduction. As it is the case with the k-means algorithm,
spectral clustering requires that the number of clusters be
specified before data grouping.
2.3. Clustering Algorithm Input Data

The algorithms outlined above require correctly defined input data. The input dataset consists of the pathloss values between each user and each of the potential BSs. In the analyzed simulation scenario, the authors had to limit the list of small cell locations to 28 potential sites. For the sake of simplicity, path attenuation was analyzed, calculated as the average attenuation for all resource blocks. Hence, there are 28 potential locations of pico-type BSs, and one feature instance representing a features dataset for one user has 29 values – 1 pathloss between the user and the macro BS, and 28 pathlosses between the user and each of the pico BSs. To sum up, the \( i \)-th input instance may be presented as the following vector: \( [PL_{\text{macroBSi}}, PL_{\text{picoBS1i}}, PL_{\text{picoBS2i}}, \ldots, PL_{\text{picoBS28i}}] \), where \( PL_{\text{macroBSi}} \) is a pathloss value between \( i \)-th user and the macro BS, \( PL_{\text{picoBSni}} \) is a pathloss value between \( i \)-th user and the \( n \)-th pico BS. The data has been pre-processed before being used as input data. All of the pathloss values have been normalized and scaled to the 0–1 range, except for \( PL_{\text{macroBSi}} \) that has been scaled to the 0–2 range in order to place a greater emphasis on this particular feature. Thanks to such alterations, algorithm should prefer to connect users to the macro BS, connecting them to pico BSs only in those cases in which such a step is required.

2.4. ML-based Algorithm

As explained in the previous section, both clustering algorithms group the input data into \( k \) groups based on their pathloss values concerning all BSs. The next step is to determine which BSs should be assigned to the created groups. The performance of the small cell location selection algorithm is considered in two scenarios, namely Scenario 1 and Scenario 2.

In Scenario 1, pieces of UE are directly associated with BSs indicated by the ML-based algorithm. After the piece of UE have been grouped, a comparative algorithm is implemented that searches for the best BS for a given UE group by checking which BS has the best (lowest) mean pathloss for the assigned users. One BS is assigned to each of the created clusters. The chosen BSs are dedicated to one cluster only, so if the number of clusters is \( k \), the number of BSs used in the network is \( k \) as well.

In Scenario 2, ML algorithms perform clustering on the pieces of UE as well. Then, just as it was the case in Scenario 1, BSs are picked for each of the groups in the same manner. The main difference is that after the BSs have been chosen, the pieces of UE are associated with BSs based on the best received signal strength, just as in a typical LTE network.

3. System Description

In the model of their system, the authors analyzed a typical fragment of an urban environment in Madrid. It consists of several buildings of different heights, a grid of streets, a wide pavement typical of shopping districts, and a park. The method that was used for generating the radio environment relied on the ray tracing method which enabled to obtain a very precise fragment of the channel coefficients. This allowed for a good representation of the actual wave propagation conditions observed in a typical radio environment (in a dense urban area). At the same time, due to the high computational complexity of this method, the authors were forced to significantly reduce the potential locations of pico base stations to 28 points. In Fig. 1, the area of the analyzed network, with individual buildings marked, is presented. The macro BS covering a large part of this area, and the potential locations for pico BSs for which the channel was generated, are marked as well. Additionally, the locations of UE have been marked in the same figure.

![Network topology subjected to analysis.](image)
power of the macro BS is 46 dBm, and the transmit power of each of the pico BSs is 30 dBm.

Algorithm 1 Network simulation scheme

**Input:** small cell location set
**Result:** KPI set for all devices

Generator channel coefficient between all (BS, UE) pair
Associate each UE with BS

for time slot \( t \) to simulation duration do
  if \( \text{mod}(t, t_{\text{assoc}}) = 0 \) then
    Associate each UE with BS
  end

Allocate RBs to UEs
Calculate interference
Calculate SINR
Calculate throughput
Save KPIs

end

A detailed description of the simulator’s operation is presented as Algorithm 1. The positions of small cells derived from the ML-based position selection algorithm from Scenario 2 are fed to the simulator as input data. In Scenario 1, information about the pattern of direct association of UE to BSs is an additional source of input data. At the initial phase of the simulator’s operation, channel coefficients are generated between each piece of UE and a BS, separately for each RB. Then, depending on the scenario, pieces of UE are connected to their respective BSs on the basis of the received signal power or based on a direct indication from the proposed algorithm. Within the main simulation loop, where the simulation duration is set to 100 ms, the following operations are performed in sequence. Every \( t_i \) (in the simulation \( t_i \) equals 20 ms), the procedure of assigning pieces of UE to BSs is commenced. For each BS separately, the RBs are allocated, using the round robin algorithm, to all pieces of UE attached to a given BS. Then, interference is calculated separately for each UE and RB, and SINR for the allocated RBs is determined. Using the Shannon formula, throughput is calculated separately for each UE and RB and is then added up for all allocated RBs. The relevant metrics – key performance indicators (KPIs) – are saved for each time slot.

Once the simulation has been completed, the average throughput, as well as the first and the third quartiles of throughput are compared to evaluate the performance of the proposed solutions. The last two values allow to evaluate transmission performance for worst case and best case scenarios, respectively.

4. Experiment Setup and Results

Here, the results obtained for each of the proposed algorithms are presented. Transmission bitrate is the key value that is compared.

For both ML algorithms and for a number of \( k \) clusters, two results are compared for Scenario 1 and Scenario 2.

Bitrates related to Scenario 1 are marked blue, while Scenario 2 results are presented with the use of yellow bars. The results of both scenarios are compared with the bitrate for \( k = 1 \), where there is only macro BS in use. All pieces of UE are assigned to this macro BS and the UE assignment method does not have any impact on the resulting bitrate. The dashed line presented in the graphs shows the bitrate level for \( k = 1 \) and is considered to be a benchmark value. In the following sections, the results for both ML algorithms and both scenarios are presented. First, results pertaining to the k-means algorithm are presented.

4.1. K-means Clustering Algorithm

First, IC values were calculated for each of the \( k \) values in order to see when the best value of \( k \) may be expected. Figure 2 shows the inertia values for different numbers of clusters \( k \). One may observe that the best results should be obtained for \( k = 2 \), since for that value of parameter \( k \), a peculiar, sudden change in the course of the inertia line is visible.

![Fig. 2. Inertia of the k-means algorithm for different numbers of clusters. The graph the number of clusters for which the clustering results should be the best.](Image 351x384 to 521x518)

Figure 3 shows the mean bitrate for different numbers of clusters and for both Scenarios. For \( k > 1 \), there is a significant improvement in the mean bitrate. The best results of the k-means clustering algorithm (Scenario 1 results) have been obtained for \( k = 5 \). For that number of clusters, the macro BS and four pico BSs have been assigned to five clusters, and the mean bitrate improved 4.2 times compared to the mean bitrate benchmark value (results for \( k = 1 \) are presented). The best results in terms of the assignment of users to the same BSs without the k-means-based user grouping algorithm (Scenario 2) have been achieved also for \( k = 5 \), and the mean bitrate has improved 4.7 times. One can see, that the mean bitrate is better for Scenario 1-based allocations for \( k = 2 \) only. This means that only for a network with one macro BS and one pico BS the bitrate with k-means is better than the bitrate for Scenario 2 with the assignment to the same two stations.
From Fig. 4, presenting results for the worst 25% of the connections, it is quite clear that Scenario 2 performs better for all BS numbers (all $k$ values). The best results for Scenario 1 have been achieved for 2 BSs ($k = 2$), with bitrate improving 52.7 times. With the growing number of clusters, the results tend to get worse, although bitrate still remains better than for one cluster only. The assignment to the same BSs in Scenario 2 renders much better results, and the best outcomes have been achieved for $k = 3$ groups, with the bitrate improving 111.9 times.

Although Scenario 2 performs, in most cases, in terms of bitrate for all users and in terms of transmission parameters for the weakest 25% of connections, Fig. 5 shows greater improvement for Scenario 1. For the best 25% of connections, the advantage caused by using more BSs is the greatest, and bitrate may be improved by up to 54.3 times for the best case of k-means-based grouping for $k = 4$. The best results for Scenario 2 have been achieved for $k = 5$, and bitrate has been improved 35.6 times.

### 4.2. Spectral Clustering Algorithm

The second set of results has been obtained using the spectral clustering algorithm. Similarly to the k-means algorithm, for the mean bit rate of connections (Fig. 6) only for two groups or two BSs ($k = 2$) the Scenario 1 grouping is better than the Scenario 2, where Scenario 1 achieved 2.6 times better bitrate and Scenario 2 achieved 2.3 better bitrate comparing with reference bit rate for $k = 1$.
to the results from Scenario 1. For \( k = 6 \) and \( k = 7 \), Scenario 2 results for spectral clustering achieve a bitrate that is over 5 times better, while in Scenario 1, results achieved with the k-means method peak at \( k = 6 \) and reach a bitrate that is 4.5 times better.

The last set of results concerns the best 75% of connections – see Fig. 8. Here, the results are also better when compared to those obtained using the k-means method. Scenario 1-based grouping achieved better results than Scenario 2 for each \( k \) value, and there are four \( k \) values for which Scenario 1 improved the bitrate 31 times (for \( k = 4, 6 \) and 7 – even 35 times).

5. Conclusion

The paper presents an algorithm for selecting the location of small cells using the ML technique. The presented simulation results showed that the choice of BS locations is performed with best users (75th percentile of throughput) preferred. However, average and the weakest (25th percentile of throughput) network users achieve lower bitrates in such a scenario. The presented algorithm is not universal and is effective in specific cases only, but it offers a promising point of departure for further studies. As an extension of the algorithm, the usage of the CRE parameter related to small cells may be considered. The application of other ML methods, such as reinforcement ML, could be taken into consideration as well.

6. Acknowledgements

This work was supported by the DAINA project no. 2017/27/L/ST7/03166 “Cognitive engine for radio environment awareness in networks of the future” (CERTAIN) funded by the National Science Centre, Poland.

References


Małgorzata Wasilewska and Łukasz Kućacz


Małgorzata Wasilewska received her M.Sc. degree in Telecommunications from Poznań University of Technology, Poland, in 2017. She is currently pursuing her Ph.D., working at the Institute of Wireless Communications, PUT. Her main field of interest are wireless communications and machine learning.

[https://orcid.org/0000-0002-3471-0516]

Łukasz Kućacz received his M.Sc. degree in Telecommunications from Poznań University of Technology, Poland, in 2018. He is currently pursuing his Ph.D., working at the Institute of Wireless Communications, PUT. His main fields of interest are programming, wireless communications, and algorithm design.

[https://orcid.org/0000-0002-3434-1917]

E-mail: malgorzata.wasilewska@put.poznan.pl
Institute of Radiocommunications
Faculty of Computing and Telecommunications
Poznań University of Technology
Polanka 3
60-995 Poznań, Poland

E-mail: lukasz.kulacz@put.poznan.pl
Institute of Radiocommunications
Faculty of Computing and Telecommunications
Poznań University of Technology
Polanka 3
60-995 Poznań, Poland