Integration of A* and k-means clustering for star network design in environments considering obstacles

Lucas Cardinal, Juliana Menzinger, Rafael A. Penchel, Melissa de Oliveira, Marcelo L. F. Abbade,
Jose Augusto de Oliveira, Mirian P. de Santos, Grethell Perez-Sanchez, and Ivan Aldaya

Abstract— **Multi-level star is a widely adopted topology in many telecommunication systems, including wireless and wired networks. The optimization of this network topology, however, is not trivial, especially in the presence of obstacles. In this paper, we employ the well-known k-means clustering method but, instead of using a distance metric that neglects obstacles, we use the A* algorithm to find trajectories that allow circumventing obstacles. To assess the feasibility of the proposed approach, we apply it to 25 users arbitrarily located in a 20×20 uniform grid. This example indicates that by adopting A* to find obstacle-aware trajectories, the mean distance between the users and their associated centroids is sensibly reduced.**

Keywords— **Multi-level star; network topology optimization; k-means; A*.**

I. Introduction

Topology optimization is a problem that arises in almost all kinds of network design, from energy distribution to transportation systems [1], [2]. It is also a recurrent problem in many telecommunication network designs based on either wired or wireless links [3]–[5]. In telecommunication applications, network topology optimization does not have a general solution due to the heterogeneity derived from the channel characteristics and the particular optimization constraints. For instance, in wireless systems operating in the UHF or VHF bands, diffraction, reflection, and the relatively low attenuation of many building materials allow transmission in not-line-of-sight conditions [6]. On the other hand, networks employing wired channels, such as optical fiber and coaxial cables, and directional wireless links, including point-to-point microwave, millimeter-wave, and optical links, require line-of-sight between the transmitter and receiver [7]. Therefore, the presence of obstacles is particularly critical in this kind of system, which is attracting increasing attention because millimeter and free-space optical systems have been proposed for some operation modes of 5G and 6G systems [8], [9]. In addition, networks may rely on different topologies, for instance, ring, star, and bus, or on their combination in multiple levels [10]. Multi-level star topology is particularly widely used in both wired and wireless networks since it tends to lead to shorter installed cable/fiber compared to other approaches and reduces the distance between nodes in wireless systems [11].

Different techniques have been proposed to address the optimization of star topology. Many initial proposals relied on Lagrangian relaxation or some of its derivatives [12]. Most of these solutions, however, pose serious challenges when they scale up to a large number of nodes [13]. In order to overcome this scalability issue, the k-means clustering technique is particularly interesting [14]. k-means may consider different distances. The most adopted distances are the Euclidean and the Manhattan distance, which are valid in many applications but cannot account for the presence of obstacles. Therefore, employing traditional distance metrics, k-means may lead to a suboptimal topology.

In this work, we address the problem of clustering in the presence of obstacles by integrating k-means with the A* algorithm, which is used to find routes and distances between two positions in a scenario considering obstacles. The A* algorithm was chosen as the metric because it does not result in deadlocks, such as bug algorithms [15], and it is more efficient than Dijkstra [16]. As a proof of concept, we applied the proposed approach to a simple scenario formed by a grid of 20×20 grid where 25 users have been arbitrarily scattered. This case indicates that considering trajectories computed throughout the A* algorithms, the average distance from the users to their respective centroids is significantly reduced. The rest of the paper is organized as follows: in Section II, we overview the basics of k-means, the A* algorithm, and their integration. The results are presented and discussed in Section III. Finally, in Section IV, the most important conclusions are drawn, and some future work is envisaged.

II. Integration of k-means and A*

In this section, we first briefly explain the k-means clustering method and the A* algorithm. Afterward, we describe a possible integration solution.

A. k-means

k-means is a popular clustering method that aims to partition N data points into k clusters. Each cluster is identified by its mass center, denominated centroid, and the elements are assigned to the cluster with the closest centroid. Finding the position of the centroids that minimize the sum of the distances from the centroids to the elements of the cluster is an NP optimization problem that can be mathematically formulated

L. C., J. M., R. A. P., M. O., M. L. F. A., J. A. O., M. P. S. and I. A. are with the Faculdade de Engenharia de São João da Boa Vista, Universidade Estadual Paulista "Júlio de Mesquita Filho" - Unesp, São João da Boa Vista - SP, e-mail: lucas.dantas@unesp.br . G. P. S. is with the División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana, Azcapotzalco, Mexico.

as [17]:

$$\arg\min_{\mu_i} \sum_{i=1}^{k} \sum_{x_j \in S_i} d\{x_j, \mu_i\}, \tag{1}$$

where μ_i is the position of the centroid of the S_i cluster and $d\{x_j, \mu_i\}$ is the distance between the x_j data point and the centroid μ_i.

Among the algorithms developed to resolve this optimization problem, Lloyd's algorithm is the widest adopted [17]. Lloyd's algorithm is composed of the following steps:

1) The position of each centroid is initialized either randomly or by applying some smarter initialization algorithm, such as k-means++ [18].
2) Iterate until the positions of the centroids do not change:
    a) Assign each element to the cluster with the closest centroid.
    b) Recompute the position of each centroid by finding the point that minimizes the distance to all the points assigned to this centroid.

When the adopted distance metric is Manhattan or Euclidean (L^1 and L^2 distances, respectively), Step 2.b. is generally accomplished by simply calculating the vectorial average values of the positions of the elements of the cluster. However, for more general distance metrics, this shortcut is not valid anymore.

B. A* algorithm

A* is a heuristic path finding algorithm developed by Peter Hart, Nils Nilsson, and Bertram Raphael, which is employed in many applications where scenarios with obstacles are present [19]. This algorithm can be considered as a heuristic version of the well-known Dijkstra's algorithm, thus generally leading to a lower complexity [20]. To find the shortest path between initial and final nodes, we split the scenario into a rectangular uniform grid. A* then explores the intermediate nodes of the grid employing a heuristic cost function given by:

$$f(n) = g(n) + h(n), \tag{2}$$

where $g(n)$ and $h(n)$ are the cost from the initial node to n and a heuristic estimate of the cost from n to the final goal, respectively. To control the visited intermediate nodes, two different lists are implemented: an open node list and a closed node list. The algorithm stops when the final node is reached. Once the path between the initial and final nodes is found, the distance between the elements can be easily computed. For simplicity, we will denominate this distance as *A* distance*.

C. Integration of k-means and A*

To perform clustering in environments with obstacles, the distance metric used in Step 2 should account for valid paths, which can be found via A*. Step 2.a is relatively straightforward since it is only necessary to calculate the A* distance from each element to all the centroids and select the centroid with the shortest computed distance. Step 2.b, on the other hand, is not so trivial since, as mentioned, the position that minimizes the distance to all the cluster elements cannot be computed by averaging the position in the two axes. In order to find the updated position of a certain centroid, an extensive search can be adopted. This approach, although intuitive, may result in multiple path-finding requests, many of which can be repetitive. A possible solution to this issue is to precompute an A* distance matrix that stores the distance (considering the obstacles) between each pair of possible nodes in the scenario. It is important to note that even if the computation of this matrix may result in time-consuming, its implementation is highly parallelizable. The flow diagram of the integration of A* distance and k-means is depicted in Fig. 1.

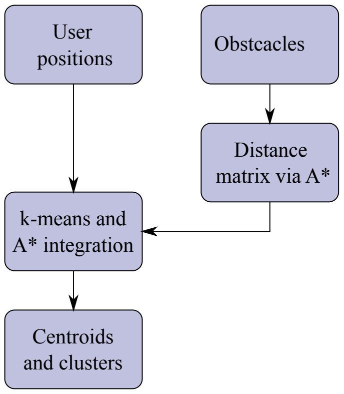

Fig. 1. Flow diagram for integrating A* distance metric into k-means clustering algorithm.

III. Results

In this section, we test the correct operation of the A* algorithm. and its application to find the centroid of a group of nodes. Then, we applied the combination of k-means and A* to a simple scenario with an arbitrary shape obstacle (represented in white) and 25 scattered users.

A. Operation of A*

First, we tested the operation of the A* algorithm by selecting four pairs of points formed by initial and final points. The results are shown in Fig. 2. For each pair, the initial and final nodes are identified with light green and red markers, respectively, whereas the path found using the A* algorithm is represented in white. As can be seen, the path connecting the initial and final nodes indeed circumvents the obstacles. In Fig. 2, we also included a colormap showing the A* distance between the initial point and all the nodes in the grid, showing that the computed distance is strongly affected by the presence of obstacles. Therefore, as expected, two geometrically close points may present a large distance if their positions are the two sides of an obstacle. This can be clearly appreciated, for instance, in Fig. 2(b). This initial analysis shows that the A* algorithm operated correctly.

B. Calculation of the centroid of a set of nodes

Once the proper operation of the A* was verified, we calculated the distance between each pair of nodes in the grid. These distances were stored in a distance matrix that was then employed to find the position of the node that minimizes the distance to all the nodes, which is equivalent to finding the centroid of these points. In Fig. 3(a) and Fig. 3(b) we show two examples. On the one hand, in Fig. 3(a), four points in an obstacle-free region are considered. In this case, the node coordinates that minimize the distance correspond to the mean value of the coordinates of the nodes. The colomap indicates the mean distance from each node to the four considered nodes. In the case of four points positioned around the obstacles, the position of the centroid is not as trivial as in the previous case and an extensive search is required.

C. *k-means with A* distance*

After testing that the centroid updating stage via distance minimization was operating properly, we executed Lloyd's algorithm on 25 users randomly scattered on the grid. Fig. 4(a) and Fig. 4(b) show the clustering of the users in 4 clusters employing the k-means method with the traditional Euclidean distance and with the A* distance, respectively. A first qualitative comparison between Fig. 4(a) and Fig. 4(b) shows that the generated clusters are completely different. This difference, nevertheless, can be attributed to the stochastic nature of the k-means algorithms. Looking carefully it is possible to observe that if Euclidean distance is considered during the clustering process, some users are assigned to centroids on the other side of some obstacle. This is not the case when the A* distance is considered as the metric for the clustering algorithm. To have a quantitative comparison, in Table I and Table II, we show the assigned cluster ID for each user alongside the mean distance between the users associated with each cluster and the corresponding centroid (this distance was calculated using A* even for the case of k-means based on Euclidean distance). It is important to note that when Euclidean distance is adopted, clusters with large mean intra-cluster distance (the distance between the elements of the cluster and its centroid) are generated. This can be explained by noting that when obstacles are neglected during the clustering process, the positions of the centroids are clearly far from optimal. If A* is adopted, on the other hand, k-means does not generate clusters with such a large mean intra-cluster distance. Considering the whole set of users, therefore, the integration of the A* algorithm with k-means leads reduces the mean distance between the users and their relative centroids from 4.7 to 3.08, showing that for this particular scenario, the use of A* during the clustering processing outperforms the use of Euclidean distance. The impact of the achieved mean distance reduction depends on the particularities of the network. For instance, in a wireless network, this may result in lower transmission power and the subsequent power consumption reduction. In the case of wired networks, on the other hand, the distance reduction can be interpreted as a lower cable installation.

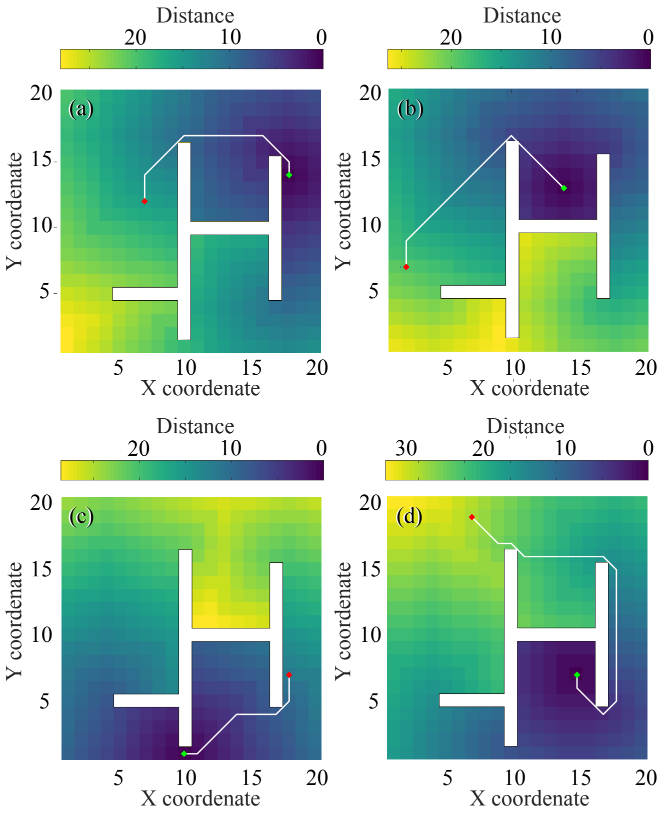

Fig. 2. Application of A* algorithm to four arbitrary pairs of initial and final nodes, which are identified with light green and red markers, respectively. The found trajectory is shown in white, and the distance from the initial node to all the grid nodes is also shown. (a) Case I, (b) Case II, (c) Case III, and (d) Case IV. (all coordinates and distances are in arbitrary units).

IV. Conclusions

In this paper, we propose to integrate the popular k-means clustering method with the A* pathfinding algorithm to assist in designing star networks in scenarios with obstacles. After showing the correct operation of the A* algorithm and how to find the centroid of a set of nodes in presence of obstacles, we applied the proposed approach to a set of 25 users randomly located in a 20×20 rectangular grid. For this case study, numerical results indicate that by adopting A* distance during the clustering process, clusters with large intra-cluster mean distance are avoided, resulting in a significant reduction of the total mean distance. These results, although promising, should be statistically validated and extended to a larger number of users. In addition, it would be interesting to consider other

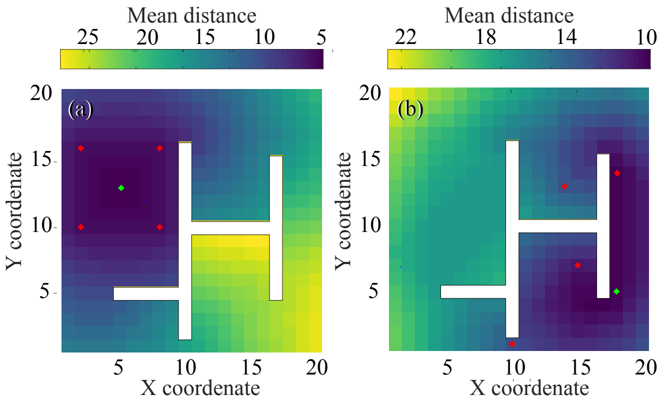

Fig. 3. Calculation of the centroid (light green) of four points (red) for points in (a) a region without obstacles and (b) points around an obstacle. The superposed colormap indicates the mean distance from the node to the four nodes indicated in red.

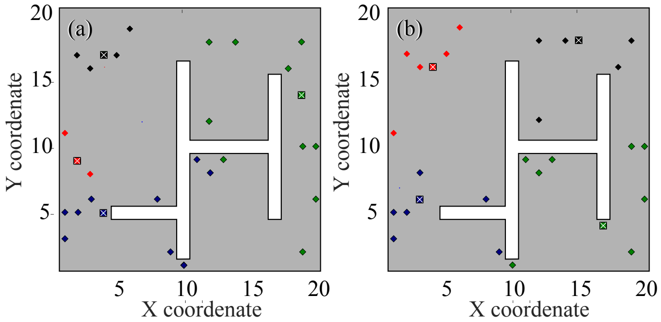

Fig. 4. Clustering of 25 users arbitrarily scattered in a grid of 20×20 using k-means based on (a) Euclidean distance and (b) A* distance. The number of clusters was set to 4. The found centroids are depicted as a square with a superimposed white x.

TABLE I

RESULTS EMPLOYING K-MEANS BASED ON EUCLIDEAN DISTANCE.

User	X coord.	Y coord.	Clust. ID	Mean dist.	Tot. mean dist.
1	1	3	1	5.4	4.7
2	1	5	1		
3	2	5	1		
4	3	6	1		
5	8	6	1		
6	9	2	1		
7	10	1	1		
8	11	9	1		
9	12	8	1		
10	3	8	2	0.5	
11	1	11	2		
12	3	16	3	0.6	
13	2	17	3		
14	5	17	3		
15	6	19	3		
16	19	2	4	6.6	
17	20	6	4		
18	13	9	4		
19	20	10	4		
20	19	10	4		
21	12	12	4		
22	18	16	4		
23	19	18	4		
24	14	18	4		
25	12	18	4		

TABLE II

RESULTS EMPLOYING K-MEANS BASED ON A*.

User	X coord.	Y coord	Clust. ID	Mean dist.	Tot. mean dist.
1	1	3	1	2.7	3.1
2	1	5	1		
3	2	5	1		
4	3	6	1		
5	3	8	1		
6	8	6	1		
7	9	2	1		
8	10	1	1		
9	1	11	2	1.8	
10	2	17	2		
11	3	16	2		
12	5	17	2		
13	6	19	2		
14	12	12	3	2.8	
15	12	18	3		
16	14	18	3		
17	18	16	3		
18	19	18	3		
19	11	9	4	4.7	
20	12	8	4		
21	13	9	4		
22	19	2	4		
23	20	6	4		
24	19	10	4		
25	20	10	4		

path-finding algorithms that allow moving not only in vertical, horizontal, and diagonal but also in other directions.

ACKNOWLEDGMENTS

The authors thank the National Council for Scientific and Technological Development (CNPq, grant numbers 432303/2018-9 and 311035/2018-3) and the São Paulo Research Foundation (FAPESP, grant 2018/25339-4).

REFERENCES

[1] A. Ogunbanwo, A. Williamson, M. Veluscek, R. Izsak, T. Kalganova, and P. Broomhead, "Transportation network optimization," in *Encyclopedia of Business Analytics and Optimization*. IGI Global, 2014, pp. 2570–2583.

[2] R. Sharma, *Network Topology Optimization: The Art and Science of Network Design*, ser. VNR computer library. Van Nostrand Reinhold, 1990.

[3] M. Resener, S. Rebennack, P. M. Pardalos, and S. Haffner, *Handbook of Optimization in Electric Power Distribution Systems*. Springer, 2020.

[4] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical networks: a practical perspective*. Morgan Kaufmann, 2009.

[5] M. Min and A. Chinchuluun, "Optimization in wireless networks," in *Handbook of Optimization in Telecommunications*. Springer, 2006, pp. 891–915.

[6] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[7] S. R. Saunders and A. Aragón-Zavala, *Antennas and Propagation for Wireless Communication Systems*. John Wiley & Sons, 2007.

[8] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5G: opportunities and challenges," *Wireless networks*, vol. 21, no. 8, pp. 2657–2676, 2015.

[9] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[10] A. Kershenbaum, *Telecommunications network design algorithms*. McGraw-Hill, Inc., 1993.

[11] G. Castañón, G. Campuzano, and O. Tonguz, "High reliability and availability in radio over fiber networks," *Journal of Optical Networking*, vol. 7, no. 6, pp. 603–616, 2008.

[12] S. Park, J. Lee, and J. Hong, "A Lagrangean based heuristic for assigning of base stations to mobile switching centers in PCS networks," in *5th International Conference of Telecommunication Systems*, 1997, pp. 224–233.

[13] J. Petrek, "A new assignment algorithm for star network topology design," in *9th International Conference on Electronics, Circuits and Systems*, vol. 2. IEEE, 2002, pp. 789–792.

[14] P. Lafata, "Advanced algorithm for optimizing the deployment cost of passive optical networks," *Advances in Electrical and Electronic Engineering*, vol. 11, no. 1, pp. 36–45, 2013.

[15] H. T. Nguyen and H. X. Le, "Path planning and obstacle avoidance approaches for mobile robot," *arXiv preprint arXiv:1609.01935*, 2016.

[16] L.-s. Liu, J.-f. Lin, J.-x. Yao, D.-w. He, J.-s. Zheng, J. Huang, and P. Shi, "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[17] E. Alpaydin, *Introduction to Machine Learning*. MIT press, 2020.

[18] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[19] M. Nosrati, R. Karimi, and H. A. Hasanvand, "Investigation of the A*(star) search algorithms: Characteristics, methods and approaches," *World Applied Programming*, vol. 2, no. 4, pp. 251–256, 2012.

[20] S. J. Russell, *Artificial Intelligence: a Modern Approach*. Pearson Education, Inc., 2010.