Trajectory-Based Multi-Hop Relay Deployment in Wireless Networks

Shilei Tian, Haotian Wang, Sha Li, Fan Wu, and Guihai Chen*

Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{tianshilei,ltwbwht,Zoey.Lee}@sjtu.edu.cn, {fwu,gchen}@cs.sjtu.edu.cn

Abstract. In this paper, we identify a novel problem *Trajectory-Based Relay Deployment* (TBRD) which aims at maximizing user connection time as the users roam through the target area while complying with relay resource constraints. To solve the TBRD, we first propose the concept *Demand Nodes* (DNs). Next, we design a Demand Node Generation (DNG) algorithm that transforms the continuous historical user trajectory into a number of discrete DNs. By generating DNs, we convert the TBRD problem into a Demand Node Coverage (DNC) problem, which is NP-complete. After that, we design an approximation algorithm, named *Submodular Iterative Deployment Algorithm* (SIDA), to solve the DNC problem with the approximation factor $1 - \frac{1}{\sqrt{e \cdot (1-1/k)}}$. The simulation on five real datasets shows that our algorithm can obtain high coverage for users in motion, leading to better user experience.

1 Introduction

With the explosive growth of mobile users, wireless coverage has become an increasingly challenging problem. However, due to transmission distance, limited coverage of Access Point (AP), path loss, and so forth, the signal quality at some locations fails to provide satisfactory Internet access [3]. Deploying relays in multi-hop networks has become an effective method to improve the wireless coverage and service quality [6].

In this paper, we investigate the relay deployment problem under the non-stationary user setting. Due to limited transmission power and path loss, the base station (BS) may fail to cover all users at all times. We hope to deploy a limited number of relays to keep users connected to the Internet as long as possible when they are wandering.

* G. Chen is the corresponding author.
This work was supported in part by Program of International S&T Cooperation (2016YFE0100300), the State Key Development Program for Basic Research of China (973 project 2014CB340303), China NSF Projects (Nos. 61672348, 61672353, 61422208, and 61472252), and CCF-Tencent Open Research Fund.

Existing works designed algorithms according to the exact user locations, they all assumed that users are stationary, which is not realistic in practice. As a result, once a user location changes, the network performance will be affected.

In fact, the movements of users within an area are not completely random. They are strongly affected by people's social demands [4]. Therefore, some *hot spots*, which mean locations where users often pass, or linger around, can be inferred from the user trajectory. Therefore, we consider utilizing the historical user trajectory to infer the tendency of the user movement and deploy the relays.

In this paper, we first define the connectivity of the network. Since relays cannot access the Internet directly, each relay must have a path to the BS. Then we define the *Trajectory-Based Relay Deployment* (TBRD) problem, which aims at maximizing user connection time as the users roam through the target area while complying with relay resource constraints. We introduce a concept *Demand Nodes* (DNs), which are *virtual weighted nodes* representing locations where users often pass or stay for a long time. Next, we propose a matrix-based trajectory representation and design the *Demand Node Generation* (DNG) algorithm. After that, the original TBRD problem is converted to a new problem called *Demand Node Coverage* (DNC). We claim that a DN is covered if its distance to an AP is less than the coverage radius of the AP. The DNC problem is to maximize the total weight of DNs covered by deployed relays and BS. The DNC is NP-complete, which can be reduced from a known NP-complete problem named *budget set cover* (BSC) [2]. To tackle this problem, we propose an approximation algorithm, named *Submodular Iterative Deployment Algorithm* (SIDA), which has an approximation ratio of $1 - \frac{1}{\sqrt{e\cdot(1-1/k)}}$, where e is the mathematical constant, and k is the relay number constraint. Finally, we use real datasets to evaluate our algorithm. The simulation results indicate that our algorithm can perform well.

The paper is organized as follows. The problem statement is given in Section 2. Section 3 describes the DNG algorithm. Section 4 presents the SIDA. Simulations are demonstrated in Section 5. Finally, Section 6 concludes this paper.

2 Problem Statement

2.1 System Model

In this model, the user can either communicate with BS directly, or connect to BS with the help of relays. Since too many hops will lead to a high delay, we limit the number of communication hops to 2.

2.2 Problem Definition

The user trajectory set is denoted by T. P_B and P_R represent the set of BS and relay candidate positions respectively. k is the number of relays we can deploy.

Definition 1 (Communication Radius). *Two APs can communicate with each other within a communication radius. We use d_B and d_R to denote the communication radius of BS and relay respectively.*

Definition 2 **(2-Hop Relay Connectivity).** *Given the AP candidate position set $P = P_B \cup P_R$, we generate a weighted graph $G = (P, E)$, where $(p_i, p_j) \in E$ if the distance between these two locations is less than the corresponding communication radius. The weight of each edge is set to 1. 2-hop relay connectivity means that in the induced graph $G[F]$, there always exists a path between any selected relay node and the selected BS node, while its distance is less than or equal to 2.*

Definition 3 **(TBRD Problem).** *Given a set of trajectories T, BS candidate locations P_B, relay candidate locations P_R, relay number constraint k, the TBRD problem is to find a BS location $p_B \in P_B$ and relay locations $P_S \subset P_R$ to maximize user connection time. P_S must be subject to $|P_S| = k$, and the induced subgraph $G[\{p_B\} \cup P_S]$ has 2-hop relay connectivity.*

As we mentioned before, hot spots can be inferred from historical user trajectory. We introduce a novel concept called *Demand Node* (DN) to represent them.

Definition 4 **(Demand Node).** *Demand Nodes (DNs) are virtual weighted nodes representing the locations where users often pass or stay for a long time. They are at the center of the grids which are generated by the division of the target area. The weight is the probability of user's appearance in the corresponding location. The larger the weight is, it is more possible that users will pass through or stay at the corresponding location.*

Definition 5 **(Coverage Radius).** *Coverage radius, denoted by r_B for BS and r_R for a relay, is the distance threshold for the BS or relay. Only DNs whose distance to an AP is less than its coverage radius can ensure Internet connection for users. We say that the DN is covered by the corresponding AP.*

Before we introduce the Demand Node Coverage (DNC) problem, we first give some definitions that are used throughout this paper. We use D to denote the DNs set, and W for the weight set of DNs.

Definition 6 **(Covered DNs Set).** *The covered DNs set $C(\cdot)$ is the set of DNs covered by a given AP. For a BS candidate location $p_B^i \in P_B$, $C(p_B^i) = \{d_j | dist(d_j, p_B^i) \leq r_B\}$ where $dist(\cdot)$ denotes the Euclidean distance. For a relay candidate location $p_R^i \in P_R$, $C(p_R^i) = \{d_j | dist(d_j, p_R^i) \leq r_R\}$.*

Definition 7 **(Weight Function).** *The weight function $w(\cdot)$ is the sum of weights of the covered DNs set. For an AP candidate location $p \in P_B \cup P_R$, $w(p) = \sum_{s_i \in C(p)} w_{s_i}$. For an AP candidate location set P, the DNs covered by P are represented as $D_C = \cup_{p_i \in P} C(p_i)$, $w(P) = \sum_{s_i \in D_C} w_{s_i}$.*

Definition 8 **(Residual Weight).** *Considering a selected AP candidate location set S_A, when we continue to select a AP candidate location set S_B, the residual weight of S_B based on S_A is defined as $w_R(S_A, S_B) = w(S_B) - w(S_A \cap S_B)$.*

Assume the width of the target area is w, and the height is h. There is also a filter threshold θ, which constrains the weight of each generated DN to be larger than θ. Now we can define the DNG problem.

Definition 9 (DNG Problem). *Given a user trajectory set T, the width w and height h of the target area, and a filter threshold parameter θ, the DNG problem is to generate a set of DNs D and a relative weight set W. The weight of each DN is in the range of $[\theta, 1]$.*

Now we can define the Demand Node Coverage (DNC) problem.

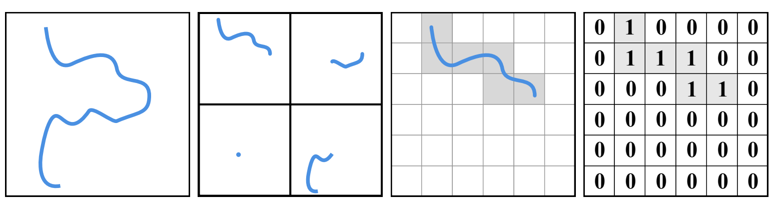
Definition 10 (DNC Problem). *Given a set of DNs D and the corresponding weight set W, BS candidate locations P_B, relay candidate locations P_R, relay number constraint k, the DNC problem is to find a location $p_B \in P_B$, and relay candidate locations subset $P_S \subseteq P_R$ to maximize $w(F)$, where $F = \{p_B\} \cup P_S$ while $|P_S| = k$. The induced subgraph complies with the 2-hop relay connectivity constraint.*

3 Demand Node Generation

In this section, we show how to extract "hotspots" which we refer to as Demand Nodes (DNs) from user trajectories. The Demand Node Generation (DNG) algorithm consists of three major steps: (1) trajectory matrix generation; (2) prediction matrix; (3) filtering.

3.1 Trajectory Matrix Generation

Since the DNs depend on both the temporal and spatial information of the user trajectory, we segment each trajectory according to a fixed time span t and record the location of each segment where the user appears in the target area by a binary matrix. Fig. 1 illustrates the details of converting a trajectory into a binary matrix. Fig. 1(a) shows a trajectory in the area.



(a) Original (b) Segments (c) 6×6 grids (d) Matrix

Fig. 1. An illustration of the process of a trajectory.

Firstly, we divide the trajectory into a number of segments, and each segment shows the trajectory of a user at the corresponding time span t, as shown in Fig. 1(b). Then, the target area is further partitioned into small sizes of grids which are the candidate locations for the demand nodes. Fig. 1(c) shows the distribution of the upper left segment. Lastly, Fig. 1(d) shows the binary matrix of the trajectory at one time segment. The whole trajectory area is seen as a matrix and entries of the matrix represent the partitioned grids. If the trajectory passes through the grid, the corresponding entry of the matrix is set to 1.

After the conversion, we obtain numerous binary matrices for the target area.

3.2 Prediction Matrix

Since the value of a grid x_{ij} is 0 or 1, we assume the probability distributions of these grids are independent Bernoulli distributions, which can be written as $x_{ij} \sim p(x_{ij}|\mu_{ij}) = \mu_{ij}^{x_{ij}}(1-\mu_{ij})^{1-x_{ij}}$, where the parameter $\mu_{ij} \in [0,1]$ is the probability of $x_{ij} = 1$.

We can estimate the μ_{ij} by maximizing likelihood estimation. However, this may lead to over-fitted results for small datasets [1]. In order to alleviate this problem, we first introduce a prior distribution $p(\mu_{ij}|a_{ij}, b_{ij})$, *beta* distribution, over the parameter μ_{ij}, which is easy to interpret while having some properties. The posterior distribution of μ_{ij} is now obtained by Bayesian theorem

$$p(\mu_{ij}|X_{ij}) = \frac{p(X_{ij}|\mu_{ij})p(\mu_{ij}|a_{ij}, b_{ij})}{\int p(X_{ij}|\mu_{ij})p(\mu_{ij}|a_{ij}, b_{ij})d\mu_{ij}}. \quad (1)$$

Then, we estimate the value of μ_{ij} by maximizing the posterior distribution $p(\mu_{ij}|x_{ij})$. We see that this posterior distribution has the form

$$p(\mu_{ij}|X_{ij}) \propto \mu_{ij}^{m+a_{ij}-1}(1-\mu_{ij})^{n-m+b_{ij}-1}. \quad (2)$$

Finally, maximizing Eq. (2) with respect to μ_{ij}, we obtain the maximum posterior solution given by $\mu_{ij} = \frac{m+a_{ij}}{n+a_{ij}+b_{ij}}$.

3.3 Filtering

After the prediction matrix of the target area is determined, the DNs are at the center of those grids with higher probabilities for 1. In our model, a threshold θ is set, and the grids whose probabilities for 1 are not less than θ are DNs.

4 Submodular Iterative Deployment Algorithm (SIDA)

We now focus on selecting the locations for APs from the candidate location set. It is clear that the weight function $w(\cdot)$ is a submodular function.

4.1 The SIDA

The main idea of SIDA is as follows. First, we construct an undirected graph $G = (P, E)$, where $P = P_B \cup P_R$. For any two nodes $p_i, p_j \in P$, $(p_i, p_j) \in E$ if $dist(p_i, p_j)$ is less than the corresponding communication radius. Then, we scan each p_B^i sequentially, and generate a subgraph with its 2-hop neighbors. The following operations are taken within this subgraph.

Algorithm 1: SIDA

Input: An instance of DNC problem, $\langle P_B, P_R, k, w(\cdot), w_R(\cdot)\rangle$
Output: The final solution F

$D \leftarrow \emptyset$;
for $b \in P_B$ **do**
  $k' \leftarrow k$; $S \leftarrow b$; $V_t \leftarrow \{v : hop(v, b) \leq 2, v \in P_R\}$; `// hop(v,b) is the least hop number from v to b, the same as below.`
  while $k' > 0$ *and* $V_t \neq S$ **do**
    $j \leftarrow 0$;
    while $j \leq \lfloor k'/2 \rfloor$ **do**
      Find $\max\{w_R(S, S \cup \{v\}) : v \in V_t\}$; $S \leftarrow S \cup \{v\}$; $j \leftarrow j + 1$;
    for $v \in S$ **do**
      if *v is not connected with b* **then**
        $V_d \leftarrow \{u|u$ is one hop neighbor of v that also one hop neighbor of $b\}$; Find $\max\{w_R(S, S \cup \{u\}) : u \in V_d\}$; $S \leftarrow S \cup \{u\}$;
    $k' \leftarrow k - |S|$;
  if $w(F) \leq w(S)$ **then**
    $F \leftarrow S$;
return F;

Next, we repeatedly select $\lfloor k/2 \rfloor$ candidate locations with maximum residual weight in the subgraph. For each selected candidate location p_i, check whether it is the 1-hop or 2-hop neighbor of the BS. If it is a 2-hop neighbor, then we check whether those selected locations can construct a path from p_i to the BS. If not, we need to select another one p_j from the 1-hop neighbors of p_i that brings the maximum residual weight while ensuring that $p_i \to p_j \to BS$ is a path. In this way, the number of all selected locations is at most $\lfloor k/2 \rfloor \times 2 \leq k$. It is very likely that we still have available relays. Therefore, assume that we have selected g relays, and $g < k$, then we run the same procedure on this subgraph with $k = k - g$. Repeat this procedure and use S to record all the selected locations, and it will terminate once $|S| = k$ and S is a feasible solution.

Finally, choose the solution with the maximum total weight. The details of SIDA are shown in Algorithm 1.

4.2 Performance Analysis

In this subsection, we analyze the performance guarantee of SIDA. We consider a BS location, its 2-hop neighbors and the generated subgraph. We propose two lemma for this subgraph.

Lemma 1. *After each greedy iteration l_i, $i = 2, \ldots, t$, $t \leq \lfloor k/2 \rfloor$, the inequality $w(G_i) - w(G_{i-1}) \geq \frac{1}{k}\left[w(OPT') - w(G_{i-1})\right]$ holds, where G_i is the selected set after i-th iteration, and OPT' is the optimal solution within the current subgraph.*

Proof. First, we denote $w(G_i) - w(G_{i-1})$ as W'_i, which is the maximum residual weight in ith iteration according to the greedy strategy. Clearly, $w(OPT') - w(G_{i-1})$ is no more than the weight of the elements covered by OPT', but not covered by G_{i-1}, i.e. $w(OPT') - w(G_{i-1}) \leq w(OPT' \backslash G_{i-1})$. Since the size of the set $OPT' \backslash G_{i-1}$ is bounded by the budget k, the total weight of DNs covered by $OPT' \backslash G_{i-1}$ and not covered by G_{i-1}, is at most kW'_i. Hence we get $w(OPT') - w(G_{i-1}) \leq kW'_i$. Substituting $w(G_i) - w(G_{i-1})$ for W'_i, and multiplying both sides by $1/k$, we get the required inequality.

Lemma 2. *After each iteration* l_i, $i = 2, \ldots, t$, $t \leq \lfloor k/2 \rfloor$, *the inequality* $w(G_i) \geq [1 - (1 - 1/k)^i] w(OPT')$ *holds.*

Proof. According to Lemma 1, we have:

$$k(w(G_i) - w(G_{i-1})) \geq w(OPT') - w(G_{i-1}) \Rightarrow \frac{w(G_i) - w(OPT')}{w(G_{i-1}) - w(OPT')} \leq 1 - 1/k.$$

Therefore, let $j = 1, 2, \ldots, i$, and multiply those inqualities, we can get:

$$\prod_{j=1}^{i} \frac{w(G_j) - w(OPT')}{w(G_{j-1}) - w(OPT')} \leq (1 - 1/k)^i \Rightarrow \frac{w(G_i) - w(OPT')}{w(G_0) - w(OPT')} \leq (1 - 1/k)^i.$$

Since $G_0 = \emptyset$, thus $w(G_0) = 0$, then we have $w(G_i) \geq [1 - (1 - 1/k)^i] w(OPT')$.

Theorem 1. *SIDA achieves an approximation factor of* $1 - \frac{1}{\sqrt{e \cdot (1 - 1/k)}}$ *for the DNC problem.*

Proof. For each BS candidate location, the algorithm iterates for at least $\lfloor k/2 \rfloor$ times. We suppose OPT is the optimal solution of the DNC problem. For the subgraph which contains OPT, we denote the set of locations selected by SIDA as F_{OPT}. Then in the light of Lemma 2, we could get:

$$w\big(F_{OPT}\big) \geq [1 - (1 - 1/k)^{\lfloor k/2 \rfloor}] w(OPT) \geq \left[1 - \frac{1}{\sqrt{e \cdot (1 - 1/k)}}\right] w(OPT).$$

5 Simulations

In this section, we conduct extensive simulation experiments to evaluate our algorithm via C++. We evaluate our entire procedure including trajectory processing, DNG, and SIDA on five real GPS data [5] from CRAWDAD: NCSU and KAIST, New York City, Orlando, and North Carolina state fair. We randomly divide each dataset into training and validation group.

The parameters of each dataset are shown in Table 1. We divide the map into grids of $g_B \times g_B$ and set the candidate BS locations to the center of these grids. Similarity, the candidate relay locations are set to the center of $g_R \times g_R$ grids. d_B is set to $r_R + r_B$, and d_R is set to $2 \times r_R$. The number of relays we

Table 1. Parameters of each dataset

Dataset	s	r_B	r_R	g_B	g_R	θ
KAIST	200	1200	600	3000	500	0.14
NCSU	200	1200	600	3000	500	0.21
New York	400	2400	1200	6000	1000	0.21
Orlando	300	1500	1000	5000	1000	0.20
Statefair	20	150	75	350	50	0.35

can deploy is $k = 5$. For simplicity, both the two parameters a_{ij} and b_{ij} of beta distribution are set to 5. The time slot is set to $t = 200$.

We repeated the partition of the validation set and ran the procedure for 1000 times, and then took the average. The coverage performance for the five datasets are 95.10%, 85.83%, 62.52%, 85.44%, and 60.70%, respectively.

6 Conclusion

In this work, we have proposed the *Trajectory-Based Relay Deployment* (TBRD) problem in wireless networks, which aims at maximizing user connection time as the users roam through the target area while complying with relay resource constraints. We first transform the trajectories into a number of virtual weighted discrete Demand Nodes (DNs). In this way, the original TBRD problem is converted to an NP-complete problem called Demand Node Coverage (DNC) problem, which is to maximize total covered DN weight. Then, we design an approximation algorithm named Submodular Iterative Deployment Algorithm (SIDA) to solve the DNC problem, with an approximation ratio of $1 - \frac{1}{e \cdot \sqrt{(1-1/k)}}$. The simulation on five real datasets results show that our algorithm can obtain high coverage performance and thus significantly improve the user experience. To the best of our knowledge, we are the first to consider user trajectories for relay deployment.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
2. Hochba, D.S.: Approximation algorithms for np-hard problems. ACM SIGACT News 28(2), 40–52 (Jun 1997)
3. Ma, L., Teymorian, A.Y., Cheng, X.: A hybrid rogue access point protection framework for commodity wi-fi networks. In: IEEE International Conference on Computer Communications (INFOCOM) (2008)
4. Musolesi, M., Hailes, S., Mascolo, C.: An ad hoc mobility model founded on social network theory. In: MSWiM. pp. 20–24 (2004)
5. Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S., Chong, S.: CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23). Downloaded from http://crawdad.org/ncsu/mobilitymodels/20090723/GPS (July 2009)
6. Wang, H., Tian, S., Gao, X., Wu, L., Chen, G.: Approximation designs for cooperative relay deployment in wireless networks. In: ICDCS. pp. 2270–2275 (2017)