Cost Effective Mobile and Static Road Side Unit Deployment for Vehicular Adhoc Networks

Donghyun Kim*, Yesenia Velasco*, Zishen Yang†, Wei Wang†[1], Rasheed Hussain‡, R.N. Uma*
* Department of Mathematics and Physics, North Carolina Central University, Durham, NC, USA.
E-mail: donghyun.kim@nccu.edu, yvelasco@eagles.nccu.edu, ruma@nccu.edu
† School of Math. and Statistics, Xi'an Jiaotong Univ., Xi'an, China. E-mail: 1164926717@qq.com, wang_weiw@163.com
‡ Department of Computer Science, Innopolis University, Kazan, Russia. E-mail: rasheed1984@gmail.com

***Abstract*—This paper investigates a new strategy to best deploy road side units so that their spatio-temporal coverage is maximized under a limited budget. For the first time in the literature, we consider three different RSU deployment strategies in a single framework, on static locations, public mobile transportation, and fully controllable vehicles. We first introduce a new strategy to abstract a map of city area into a grid graph. Then, we formulate the problem as a new optimization problem and show its NP-hardness. To solve this problem, we transform this problem into another optimization problem and propose a new polynomial running time approximation algorithm and show its performance ratio is at least the half of the best possible ratio.**

***Index Terms*—Vehicular ad-hoc networks, road side unit deployment, approximation algorithm, graph theory, optimization.**

I. Introduction

Vehicular ad hoc network (VANET) refers to a wireless networks of mobile vehicles. Compared to the widely-investigated wireless networks such as wireless ad-hoc networks or wireless sensor networks, VANET is uniquely characterized by the heavy presence of back-end infrastructure, in particular, road side units (RSU), which are relay nodes connecting the VANET nodes to the rest of the network. Naturally, RSU is a key component for cooperative and distributed applications in VANET. Nowadays, RSUs are also considered for different roles such as traffic directories, data disseminators, security management, location servers, and service proxies. Due to the significance of the roles of RSUs in VANET, the proper distribution of RSUs is of great importance to improve the service quality of VANET. Naturally, this issue has attracted a lot of attention. In [1], Barrachina et al. identified that the cost of densely deploying RSUs could be a major hinderance to make VANET service ubiquitously accessible. Their report also shows that as the traffic density of an area differs over time, maximizing the utility of fixed RSUs is very challenging. In [2], Aslam et al. considered to deploy a limited number of fixed RSUs to maximize their utilization over time. Later, Wu et al. [3], Liang et al. [4] considered similar issues of best deploying fixed RSUs. More recently, several researches are conducted to use public infrastructure such as buses and taxis to disseminate messages [5] or route messages by using them as data mules [6].

[1]Corresponding Author.

This paper proposes a new aggressive strategy to unify and complement the existing approaches to best deploy available RSUs to maximize their coverage. More specifically, we assume that there is a budget limitation to deploy RSUs, known costs to deploy each RSU on (a) a fixed location, (b) a public transportation such as a bus and a light rail, whose routes are known in advance, but not controllable, and (c) a fully controllable vehicle, which is owned by the local government, as well as the statistical information of the traffic density over each area. Then, we introduce a new strategy to best deploy RSUs using the three different types of deployment strategies under the limited budget. The list of the main contributions of this paper can be summarized as follows: (a) To the best of our knowledge, this is the first paper in the literature to consider three different RSU deployment strategies on a unified framework; static, mobile but not controllable, and mobile and fully controllable. (b) We introduce a new strategy to abstract a given metropolitan map into a grid graph such that when a fixed RSU is deployed over a point on the grid graph, then the whole corresponding region to the point in the map can be covered by the RSU placed on the center of the region. (c) We convert the problem of our interest into a new NP-hard optimization problem, namely the geometric budget coverage problem (GBCP), and show that it is NP-hard. (d) We transform GBCP to a new optimization problem called the budgeted maximum coverage problem with cardinality constraint and propose a new polynomial time approximation algorithm for it. *We show that the performance ratio of this algorithm is at least the half of the best possible*.

The rest of this paper is organized as follows. The formal definition of the problem and its justification are in Section II. We introduce a transformation of the problem into another optimization problem in Section III and propose a new polynomial time approximation algorithm for it in Section IV. Finally, we conclude this paper in Section V.

II. Problem Statement

In this paper, we investigate how to deploy various static and mobile RSUs on a metropolitan area so that the coverage of RSUs can be maximized under a limited budget. We have the following assumptions: (a) Assumption 1: RSUs can be deployed on a static location (D-Type 1), on mobile public transportation such as buses and light rails, which are

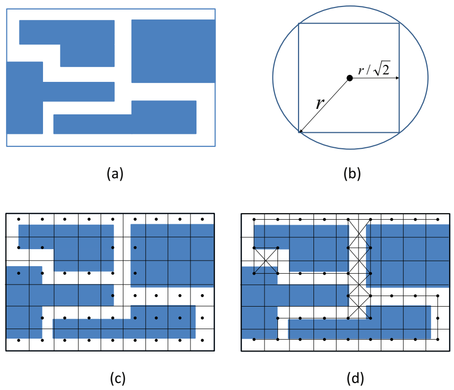

Fig. 1. Abstraction of a metropolitan area map M into a graph.

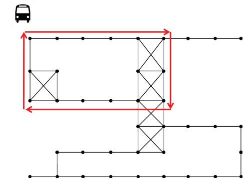

Fig. 2. Location of a mobile public infrastructure varies over time.

mobile but not controllable (D-Type 2), and/or on government vehicles which are fully controllable by need (D-Type 3), (b) Assumption 2: The cost to deploy an RSU on each deployment type is fixed and known in advance, (c) Assumption 3: In case of Deployment D-Type 2, each mobile transportation does not suffer from any delay, and their travel schedule is known. Note that this is mostly true for light rails, as well as for buses within a city area without heavy traffic jam. This assumption implies that the location of each transportation at any moment of a day is known in advance, (d) Assumption 4: D-Type 3 does not suffer from traffic jam. In practice, this can be handled by constructing their travel schedule under very low speed, and (e) Assumption 5: The significance of each region (e.g. traffic load) within the metropolitan area is available. This can be obtained by collecting the relevant statistical information over time. In the following, we first introduce a new way to abstract a metropolitan area map M into a graph model. Then, we formulate the problem of our interest as an optimization problem on the graph. Last, we show the problem of our interest is NP-hard.

A. Abstraction of Topology

In this paper, we assume the shape of the map M of a metropolitan area of our interest is a rectangle, e.g. Fig. 1(a). Suppose the communication range of both VANET nodes and RSUs are equal to r. Next, we partition M into regular squares whose height and width are $(r/\sqrt{2})\times 2$, e.g. Fig. 1(b). Observe that the RSU at the center of the grid square with this length and height is accessible from a VANET node in any location within the grid square. Now, we represent each grid square as a point and obtain a set of grid points representing the whole map, e.g. the points in Fig. 1(c). Finally, we construct a topology graph $G = (V, E)$ such that V is the set of central points of the grid squares. For each pair of points $u, v \in V$, $(u, v) \in E$ if the two squares, whose central points are u, v, are adjacent in M, e.g. Fig. 1(d).

Once the topology graph is constructed, then we assign a weight on each node, which implies the importance of the node, e.g. business of the grid square. By definition, if a grid space includes a popular spot with more traffic, then the corresponding node will get a higher weight. Note that the weight of a node may vary during the day as the traffic situation of a metropolitan area changes over time. Therefore, if we divide a day into T consecutive time slots, we can obtain a set of temporal graphs [9] $\mathcal{G} = \{G_0 = (V_0, E_0, w_0 : V_0 \rightarrow R^+), G_1 = (V_1, E_1, w_1 : V_1 \rightarrow R^+), \cdots, G_T = (V_T, E_T, w_T : V_T \rightarrow R^+)\}$ as an input of our problem of interest. Note that $V_0 = V_1 \cdots = V_T$ and $E_0 = E_1 \cdots = E_T$, but the graphs are only differ in node weights.

B. Problem Definition using Abstracted Topology

Previously, we explained how to obtain a set of temporal graphs $\mathcal{G}$ which represents the logical relationships among the subregions (grid squares) in the metropolitan area and their corresponding significance over time. We can easily observe that some deployment strategies (D-Type 1 and D-Type 2) over the set of temporal graphs have the following characteristics: (a) In case of a static RSU deployment (D-Type 1), once we decide to deploy one RSU on a node in G_0, it also represents the deployment of the same node on the rest of the graphs in $\mathcal{G}$ as the RSU is not mobile. This is because by our construction, the graphs are different only in node weights. (b) In case of D-Type 2, as we can see in Fig. 2, the location of a mobile public transportation varies over time. Therefore, once we decide to deploy an RSU on such mobile unit, the exact location of the RSU may be changed in each graph in $\mathcal{G}$. Still, as the travel schedule of the RSU is known in advance (following Assumption 3), we know which node in each $G_i \in \mathcal{G}$ will be covered by an RSU on the mobile transportation exactly. This means that when we deploy multiple RSUs using D-Type 2 strategy, the group of nodes covered by the RSUs changes over time, but we can compute what they are at each moment exactly. Now, let us introduce one related problem.

Definition 1 (Budgeted Maximum Coverage Problem)**.** *Given a positive integer k, a budget B, a set $S = \{s_1, s_2, \cdots, s_n\}$, their corresponding weights $W = \{w_1, w_2, \cdots, w_n\}$, a collection $\mathcal{S}$ of subsets $\{S_1, S_2, \cdots, S_m\}$ of S, and their corresponding cost $C = \{c_1, c_2, \cdots, c_m\}$, the budgeted maximum coverage problem is to find a subset $\mathcal{S}' \subset \mathcal{S}$ to maximize $\sum_{s_i \in S} w_i \cdot y_i$ subject to*

(a) $\sum_{1 \le j \le m} c_j \cdot x_j \le B$, *// the cost of selecting the subsets is no greater than the budget limit.*

(b) $\sum_{s_i \in S_j} x_j \ge y_i$ *for each* s_i, *//* $y_i = 0$ *only if for every* S_j

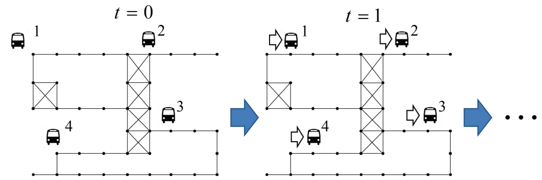

Fig. 3. The subset of nodes covered by the RSUs attached to mobile public infrastructures are changing over time, but can be predicted.

containing s_i*,* $x_j = 0$*.*

(c) $0 \leq y_i \leq 1$*, // $y_i = 1$ only if s_i is covered, i.e. $x_j = 1$ for some S_j including s_i.*

(d) $x_j \in \{0, 1\}$*. // $x_j = 1$ only if S_j is selected, i.e. $S_j \in \mathcal{S}'$.*

Note that the budgeted maximum coverage problem is a known NP-hard problem [8].

Based on our discussion so far (without considering D-Type 3), we can construct a budgeted maximum coverage problem instance as follows: Given a budget B, we first construct a set $S = V(G_0) \bigcup V(G_1) \bigcup \cdots \bigcup V(G_T)$, where $V(G_i)$ is the set of nodes in G_i with corresponding node weights. Note that the weight of each node in S is known in advance and we have W. Then, we create an empty collection $\mathcal{S}$ of subsets of S. Then, for each node $v_i \in G_0$ and its identical nodes in the rest of the temporal graphs $G_1, \cdots, G_T$, we create a subset S_i and add it to $\mathcal{S}$. Then, each S_i represents a grid point covered by a static RSU deployed on v_i over time period $[0, T]$. Next, for each public mobile transportation, we collect the set of grid nodes covered over time period $[0, T]$ by an RSU over the mobile transportation and construct a new subset. Then, we also add this subset to $\mathcal{S}$. This subset represents the set of grid points covered by the RSU attached to the mobile transportation over time. For each subset in $\mathcal{S}$, we assign the known cost to the subset which is corresponding to the cost to deploy and operate an RSU to cover the nodes in the subset over time, then we have C. Then, we obtain a budgeted maximum coverage problem instance.

Finally, let us discuss how the consideration of D-Type 3 deployment strategy (fully controllable) will impact our formulation so far. Given a starting location of fully controllable node with an RSU in G_0, the node can always stay at the same location or move to adjacent location in the next graph in the set of temporal graphs. From this observation, we can construct a new directed acyclic graph (DAG) $G_U = (V_U, E_U)$ such that $V(G_U) \leftarrow \bigcup_{G_i \in \mathcal{G}} V(G_i)$ and for each $v \in V(G_i)$ and $u \in V(G_{i+1})$ pair, there exists a directional edge from v to u in E_U only if $v = u$ or v and u are adjacent in G_i (which also means that they are adjacent in G_{i+1}). Fig. 4 shows that under such construction, a feasible path of a mobile node which is located at $g_{1,0}$ is a path from G_0 to G_T, and the number of such paths is exponential. This means that while our problem of interest is similar to the budgeted maximum coverage problem, it is significantly more challenging as there are so many choices to construct a subset for each fully controllable mobile node, which becomes a new subset into

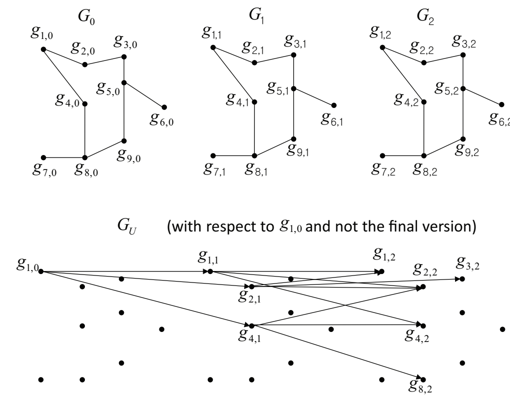

Fig. 4. This figure shows the movement of a fully controllable mobile node over time becomes one subset in S in the budgeted maximum coverage problem instance. For instance, if we decide to move the node through $g_{1,0} \rightarrow g_{1,1} \rightarrow g_{2,2}$, a corresponding subset $\{g_{1,0}, g_{1,1}, g_{2,2}\}$ is added to $\mathcal{S}$. This movement implies that a mobile node starts from $g_{1,0}$ and stay there for one more time unit and then finally move to $g_{2,3}$.

$\mathcal{S}$ in the formulation of the budgeted maximum coverage problem. Below is the formal definition of our problem of interest.

Definition 2 (Geometric Budgeted Maximum Coverage Problem)**.** *Given (a) a DAG $G_U = (V_U, E_U)$ with their corresponding node weights W, a collection $\mathcal{S}$ of subsets of V and their corresponding cost $\mathcal{C}$, and a budget B, the geometric budget coverage problem is to construct a subcollection of subsets of $\mathcal{S}$ by (a) computing subsets, each of which represents the grid points covered by fully controllable mobile nodes with an RSU which moves over G_U and (b) selecting additional subsets, each of which represents the set of grid points covered by either a static RSU or an RSU attached to a public mobile transportation, such that the total cost to deploy the RSUs is under the limited budget B and the coverage of RSUs over time is maximized, which is the sum of the weight of each grid point in any $G_i \in \mathcal{G}$ covered by the subsets in the subcollection.*

It is worthwhile to notice that the conditions (a) and (b) are not necessarily in that order. That is, we may compute the paths for fully controllable mobile nodes first and then select some more subsets from $\mathcal{S}$, or vice versa. Also, it is allowed to do interchangeably. Meanwhile, it is easy to see that the geometric budget coverage problem is NP-hard as its simplest version without any fully controllable mobile node (this is possible by assuming the cost to operate each fully controllable mobile node is greater than the given budget), is equivalent to the budgeted maximum coverage problem.

Remark 1. *The cost to deploy a fully controllable mobile node with an RSU and operating them for whole year can be much higher than deploying an RSU on a fixed location or a mobile public transportation. This means that the maximum number k of fully controllable mobile nodes under a limited budget may not be huge.*

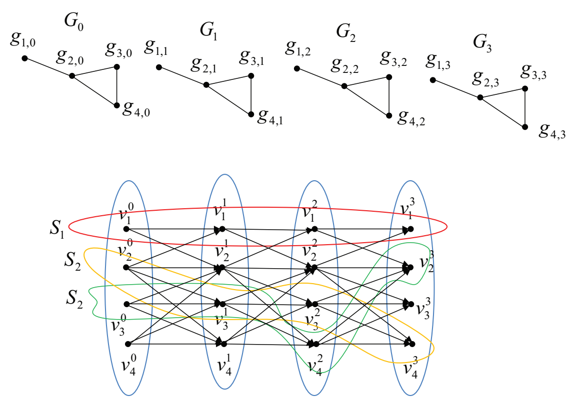

Fig. 5. This figure shows how $\mathcal{G}$ from a geometric budgeted maximum coverage problem instance is used to construct Γ in a budgeted maximum coverage problem with cardinality constraints instance.

Due to Remark 1, in the rest of this paper, we will focus on a variation of geometric budgeted maximum coverage problem, in which the number of available fully controllable mobile nodes is specifically given as a positive integer k. Clearly, by solving this problem in polynomial time, we can solve the original version in polynomial time by selecting the best result among all possible choices of the number of fully controllable mobile nodes, which is bounded by the limited budget B divided by the cost of deploying one fully controllable mobile node.

III. Budgeted Maximum Coverage Problem with Cardinality Constraints

Now, we reformulate the geometric budgeted maximum coverage problem, whose input is $\langle \mathcal{G}, W, C, k, B, T\rangle$ to a new optimization problem namely the budgeted maximum coverage problem with cardinality constraints (BMCP-CC) as follows. First, from $\mathcal{G}$ and W, we can construct a DAG $\Gamma = (V', E')$ as follows: $V' \leftarrow V(G_0) \cup V(G_2) \cup \cdots \cup V(G_T)$. For any two nodes u and v, there exists a direct edge from u to v in E' only if (a) $u \in G_i$ is same as (a copy of) $v \in G_{i+1}$ or (b) $u \in G_i$ and $v' \in G_i$ are adjacent in G_i, where v' is a copy of $v \in G_{i+1}$. Note that this construction is similar to the construction of G_U. Without loss of generality, suppose $V' = \{v_1^0, v_2^0, \cdots, v_n^0; v_1^1, v_2^1, \cdots, v_n^1; \cdots, v_1^T, v_2^T, \cdots, v_n^T\}$, where each v_i^j represents the i-th node of V in the j-th moment $t = j$ for $j = 0, 1, \cdots, T$, i.e. v_i^j is a copy of v_i in $V(G_j)$. There is a directed edge from v_i^j to v_k^{j+1} only if either nodes v_i and v_k are adjacent in graph G or $i = k$; see Fig 5 for example. Now, we are given a collection of subsets $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 = \{S_1, S_2, \cdots, S_m\} \cup \{S'_1, S'_2, \cdots, S'_l\}$, (a) where each subset S'_i in $\mathcal{S}_2$ contains $T+1$ nodes from V', which forms a directed path from some nodes in time $t = 0$ to some nodes in time $t = T$ in graph Γ corresponding to the trajectory of the stationary s(and mobile) RSUs. The cost $c(S'_i)$ of S'_i is given in advance, the coverage benefit of S'_i is $w(S'_i) = \sum_{v \in S'_i} w(v)$, (b) while $\mathcal{S}_1$ **represents the set of all possible trajectories for fully controlled sensors in** Γ, which consists of all possible subsets of nodes constituting a directed path from some nodes in time $t = 0$ to time $t = T$ in graph Γ. Note that the cardinality of $\mathcal{S}_1$ can grow exponentially in terms of T and n, and cannot be given explicitly.

Definition 3 (Budgeted Maximum Coverage Problem with Cardinality Constraints, BMCP-CC)**.** *Using notations above, given a a positive integer k and a budget B, we are asked to select k subsets from $\{S_1, S_2, \cdots, S_m\}$ and some subsets from $\{S'_1, S'_2, \cdots, S'_l\}$ with total cost no more than B, such that the total weights of nodes covered is maximized.*

IV. A New Approximation Algorithm for BMCP-CC

In this section, we will design a $\frac{1}{2}(1 - \frac{1}{e})$-approximation for BMCP-CC. The basic idea follows from the existing work [8]. However, since the number of subsets in $\mathcal{S}_2$ can be exponentially large, even a simple greedy strategy does not work (it takes exponential time if we simply enumerate all possibilities). Fortunately, we manage to make the greedy strategy work, by exploiting the special structures of Γ (which is a directed acyclic graph), based on dynamic programming strategy and a clever node weight reassignment procedure.

The basic idea of our algorithm is based on the greedy strategy. The algorithm mainly consists of two independent stages. In the first stage, we apply the greedy algorithm for the Maximum k Coverage Problem with $\mathcal{S}_1$ as the input. The second stage uses greedy strategy to solve the budgeted maximum coverage problem with input $\mathcal{S}_2$ and budget B. The algorithm takes the union of solutions obtained in two stages as the outputs.

Algorithm 1. Algorithm for BMCP-CC

Input: $(G = (V, E), T, B, k, \mathcal{S}_1)$

Output: k subsets from $\mathcal{S}_1$ and a sub-collection $\mathcal{S}' \subset \mathcal{S}_2$ with cost at most B.

- Step 1 (**Greedy Algorithm for Maximum k Coverage**). $\mathcal{A}_1 \leftarrow \emptyset$; Select subsets in $\mathcal{S}_1$ into $\mathcal{A}_1$ in a greedy manner, i.e., first find a subset $S \in \mathcal{S}_1$ which covers nodes with maximum total weight, then at each round, pick a subset in the remaining subsets in $\mathcal{S}_1$ which covers the uncovered nodes in V' with maximum total weight, until there are k subsets selected in $\mathcal{A}_1$. Let $\mathcal{A}_1 = \{S_1, S_2, \cdots, S_k\}$ be the collection of selected subsets in this step.
- Step 2 (**Greedy Algorithm for the Budgeted Maximum Coverage Problem**). Apply the $(1-1/e)$-approximation algorithm in [8] for the budgeted maximum coverage problem over BMCP-CC with $k = 0$ (which means we have no fully controllable mobile nodes. Under this condition, BMCP-CC becomes a traditional budgeted maximum coverage problem). Suppose after running this algorithm, we obtain $\mathcal{A}_2 = \{\hat{S}_1, \hat{S}_2, \cdots, \hat{S}_p\}$, for some $p \le l$.
- Step 3. Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, output $\mathcal{A}$.

The main consideration is how to efficiently select subsets from $\mathcal{S}_1$ such that the greedy strategy works. Next, we show Step 1 can be done in polynomial time, by using by dynamic programming. We remark, however, that finding the longest paths in general graphs is NP-hard.

Step 1(a). Finding the longest directed path in Γ (starting from a node in $t = 0$ and ending at a node in $t = T$).

Input: $(G = (V, E), w, T,)$

Output: the longest path staring from $t = 0$ and ending at $t = T$.

(i) Transform Γ into an edge-weighted graph as follows: Let $e_{ij} = (v_i, v_j)$ be a directed edge. Then $w(e_{ij}) = w(i) + w(j)/2$ if $i = 0$ and $j \neq T$; $w(e_{ij}) = (w(i) + w(j))/2$, if $1 \leq i, j \leq T - 1$; $w(e_{ij}) = w(i)/2 + w(j)$, if $i \neq 0$ and $j = T$; and $w(e_{ij}) = w(i) + w(j)$ if $i = 0$ and $j = T = 1$.

(ii) (Dynamic Programming) We shall find the longest path between any pair of nodes starting from time $t = 0$ and ending at time $t = T$. First, define the outgoing neighbors of a node v in Γ to be $N_{out}(v) = \{u|(v, u) \in E(\Gamma)\}$ and the incoming neighbors of v to be $N_{in}(v) = \{u|(u, v) \in E(\Gamma)\}$.
Let $dist(v_i^0, v_k^j)$ be the distance of the longest path between v_1^0 and v_k^j. Then $dist(v_i^0, v_k^0) = 0$, $\forall i, k = 1, 2, \cdots, n$. Generally, $dist(v_i^0, v_k^j) = \max_{r \in N_{in}(v_k^j)}\{dist(v_i^0, r) + w(r, v_k^j)\}$, for $i, k = 1, 2, \cdots, n$ and $j = 1, \cdots, T$. Based on above recursive relation, we can find all the longest paths from v_i^0 to any node v_k^T at time $t = T$.

(iii) After computing all the longest distance between v_i^0 and v_j^T for any $i, j = 1, 2, \cdots, n$. We can choose among them the longest paths from a node in $t = 0$ to a node in $t = T$.

Step 1(b). Procedure(Γ, T, B, k)

Input: (Γ, T, B, k)

Output: k subsets in $\mathcal{S}_1$.

(i) In the edge weighted graph Γ, find the first longest path, say $P_1 = v_{i_1}^0 v_{i_2}^1 \cdots v_{i_n}^T$;

(ii) Reset the node weights of P_1 to be zero in graph Γ, then reconstruct the edge weight of Γ according to the rule in Step 1(a). Find the longest path, say P_2, in Γ with new edge weights, by using Step 1(a);

(iii) Repeat the above process until k paths having been selected.

Now we have the following theorems.

Theorem 1. *Algorithm 1 runs in polynomial time.*

Proof. The time complexity of Algorithm 1 is dominated by Step 1 in computing k longest paths in graph Γ with varied edge weights. The time complexity of Step 1(a) for computing one longest paths is $O(n^3(T + 1))$. So the total complexity is $O(kn^3(T + 1)$. $\square$

Theorem 2. *Algorithm 1 is a $\frac{1}{2}(1 - \frac{1}{e})$-approximation for BMCP-CC, which is guaranteed to produce a solution at least $\frac{1}{2}(1 - \frac{1}{e})$ times the optimal solution.*

Proof. Let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 = \{S_1, S_2, \cdots, S_k\} \cup \{\hat{S}_1, \hat{S}_2, \cdots, \hat{S}_p\}$ be the solution of BMCP-CC obtained by Algorithm 1. Let $\mathcal{A}^* = \mathcal{A}_1^* \cup \mathcal{A}_2^* = \{S_1^*, S_2^*, \cdots, S_k^*\} \cup \{\hat{S}_1^*, \hat{S}_2^*, \cdots, \hat{S}_q^*\}$ be an optimal solution of BMCP-CC. Let OPT_1 be an optimal solution of the Maximum Coverage Problem by selecting k subsets from $\mathcal{S}_1$, and let OPT_2 be an optimal solution of the Budgeted Maximum Coverage Problem by only selecting some subsets from $\mathcal{S}_2$ with total costs at most B. Then it follows from [7] and [8] respectively that

$$w(\mathcal{A}_1) = w(S_1 \cup S_2 \cup \cdots \cup S_k) \geq (1 - 1/e)OPT_1, \quad (1)$$

$$w(\mathcal{A}_2) = w(\hat{S}_1 \cup \hat{S}_2 \cup \cdots \cup \hat{S}_p) \geq (1 - 1/e)OPT_2. \quad (2)$$

Note $w(\mathcal{A}_1 \cup \mathcal{A}_2) \geq \max(w(\mathcal{A}_1), w(\mathcal{A}_2)) \geq \frac{w(\mathcal{A}_1)+w(\mathcal{A}_2)}{2}$. It follows that $w(\mathcal{A}_1 \cup \mathcal{A}_2) \geq \frac{w(\mathcal{A}_1)+w(\mathcal{A}_2)}{2} \geq \frac{1}{2}(1 - 1/e)(OPT_1 + OPT_2) \geq \frac{1}{2}(1 - 1/e)(w(\mathcal{A}_1^*) + w(\mathcal{A}_2^*)) \geq \frac{1}{2}(1-1/e)w(\mathcal{A}_1^* \cup \mathcal{A}_2^*) = \frac{1}{2}(1-1/e)w(\mathcal{A}^*)$, where $w(\mathcal{A}_1^*) = w(S_1^* \cup S_2^* \cup \cdots \cup S_k^*) \leq OPT_1$ and $w(\mathcal{A}_2^*) = w(\hat{S}_1^* \cup \hat{S}_2^* \cup \cdots \cup \hat{S}_q^*) \leq OPT_2$ follow from the fact that OPT_1 and OPT_2 are optimal solutions, respectively. This completes the proof. $\square$

V. Conclusion

In this paper, we propose a new strategy to deploy RSUs under the limited budget. As a future work, we plan to investigate the tightness of our algorithm and further investigate the existence of approximation algorithms with better performance ratio to close the gap.

Acknowledgement

This work was supported in part by US National Science Foundation (NSF) HRD-1345219 and HRD-1533653. This research was jointly supported by National Natural Science Foundation of China under grants 11471005.

References

[1] J. Barrachina, P. Garrido, M. Fogue, F.J. Martinez, J.C. Cano, C.T. Calafate and P. Manzoni, "Road Side Unit Deployment: A Density-Based Approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 30-39, 2013.

[2] B. Aslam, F. Amjad and C.C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," *Proc. of 2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 423-429, 2012.

[3] T.-J. Wu, L. Wanjiun and C. Chung-Ju, "A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks," IEEE Transactions on Communications, vol. 60, no. 8, pp. 2295-2303, 2012.

[4] L. Yingsi, L. Hui and D. Rajan, "Optimal Placement and Configuration of Roadside Units in Vehicular Networks," *Proc. of 2012 IEEE Vehicular Technology Conference (VTC 2012-Spring)*, pp. 1-6, 2012.

[5] Z. Lei, Z. Yanyan, P. Jianping, L. Kaur and Z. Hongzi, "Multi-modal Message Dissemination in Vehicular Ad-hoc Networks," *Proc. of 2012 1st IEEE International Conference on Communications in China (ICCC)*, pp. 670-675, 2012.

[6] L. Kun-chan and W. Ze Ming, "On the Feasibility of using Public Transport as Data Mules for Traffic Monitoring," *Proc. of 2008 IEEE Intelligent Vehicles Symposium*, pp. 979-984, 2008.

[7] D.S. Hochbaum, ed. "Approximation Algorithms for NP-Hard Problems", PWS Publishing Company, 1997.

[8] S. Khuller, A. Moss, and J. Naor, "The Budgeted Maximum Coverage Problem," *Information Processing Letters*, vol. 7, issue 1, pp. 39-45, 1999.

[9] V. Kostakos, "Temporal graphs, *Physica A: Statistical Mechanics and its Applications*, vol. 388, issue 6, 1007-1023, 2009.