Making Many People Happy: Greedy Solutions for Content Distribution

Yunsheng Wang, Yuhong Guo, and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122
Email: {yunsheng.wang, yuhong, jiewu}@temple.edu

***Abstract*—The increase in multimedia content makes providing good quality of service in wireless networks a challenging problem. Consider a set of users, with different content interests, connected to the same base station. The base station can only broadcast a limited amount of content, but wishes to satisfy the largest number of users. We approach this problem by considering each user as a point in a 2-D space, and each type of broadcast content as a circle. A point that is covered by a circle will be satisfied, and the closer the point is to the center of the circle, the higher the satisfaction. In this paper, we first formulate this problem as an optimal content distribution problem and show that it is NP-hard. The optimal problem can also be extended into an m-dimensional (m-D) space, and distance measurements can be expressed in a general p-norm. We then introduce three local greedy algorithms and compare their complexity. The approximation ratio of our greedy algorithms to the optimization problem is also formally analyzed in this paper. We perform extensive simulations using various conditions to evaluate our greedy algorithms. The results demonstrate that our solutions perform well and reflect our analytical results.**

***Index Terms*—Approximation ratio, content distribution, local greedy algorithm, maximum coverage, optimization.**

I. Introduction

The growth in the amount of mobile devices, such as smartphones and tablets, coupled with the popularity of multimedia content, places a significant strain on existing wireless networks. The field of content distribution for wireless networks [1], [2], [3], [4] has emerged in an attempt to address this problem. In this paper, we propose to formulate the content distribution problem as an optimization problem which considers the interest distance between the content and the user's interest.

Fig. 1 illustrates our problem setting. We assume the base station (BS) can only broadcast k times in a given period of time to n number of users. The BS is limited to k broadcasts for performance reasons, where $k < n$. Each user will be happy if the information that was broadcast is close to his interest. We seek to answer the question of choosing the content to broadcast to make the most number of users happy.

This problem can be abstracted as follows. Suppose there are n points in a 2-dimensional (2-D) space to be covered by k circles of radius r. Each point x_i has a maximum reward of w_i, and a point will return a reward if it is covered by a circle. The amount of reward is determined by its distance to the center of the circle. Our objective is get the maximum reward for all of the points. In this abstraction, each point corresponds to a particular user's interest. Two dimensions is considered to be two attributes of an interest. Radius determines the boundary of closeness between the content and matching interest. The reward denotes happiness, but with a given bound.

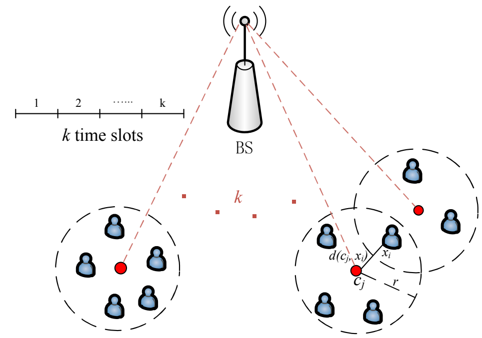


Fig. 1. Content distribution in a sequence of k broadcast in the m-D space.

In the real world, people (users here) have their own interests. If the broadcast contents meet their interests, users will be happy and gain some rewards. For example, one user is interested in classic music. If the BS broadcasts light music, this user will be happy to hear this content. Otherwise, if rock music is broadcast, the user will not gain any rewards. The total rewards that each user can gain is capped. The interest distance is the difference between the broadcasting contents and the users interests, which can be used to measure the gained rewards. Our objective is to maximize the total rewards gained by all of the users. We use multi-dimensional vectors (*m* keywords in *m*-D space) to represent the contents and the users' interests.

In this paper, we first formulate our objective into an optimization problem and prove its NP-hardness. Then we propose three local greedy algorithms to solve the problem. These algorithms can be implemented in the m-D space. The interest distance can be calculated in a general p-norm. Here, we only consider the 1-norm and 2-norm (physical distance in the 2-D space) models. We also design weighted and unweighted schemes to better reflect the relative importance of each node in the network. An analysis of the approximation ratio of the greedy algorithms is also given. Trace-driven evaluations show the good performance of our local greedy algorithms.

The major contributions of our work are as follows:

- We introduce the reward function to measure the quality of content distribution in a content distribution network.
- We prove the optimization problem is an NP-hard problem.
- We develop three local greedy algorithms to solve the problem and calculate the approximation ratio.
- We evaluate the algorithms in synthetic traces. The simulation results show the good performance of the proposed greedy algorithms.

The rest of the paper is organized as follows. Section II discusses relevant previous work. Section III presents the preliminary work, where the problem definition is given and p-norm as interest distance is reviewed. Section IV explains the objective function in a formal model and provides the proof of its NP-hardness. Section V describes our designed local greedy algorithms with the approximation ratio. Section VI analyzes the experiment results. Section VII concludes the paper. The detailed proof of the approximation ratio is given in the appendix of the paper.

II. Related work

A. Content distribution

Content distribution for peer-to-peer (P2P) applications is an important research problem [5], [6], [7], [8]. Content distribution protocols allow personal computers to function in a coordinated manner as a distributed storage medium by contributing, searching and obtaining digital content. There is a rich amount of literature on the design and performance analysis of content distribution algorithms in wireless networks [1], [2], [3], [4], [9], [10], [11], [12]. These protocols mainly rely on flooding, not only to maintain the topology information, but also to distribute the content availability. Our approach differs from these works because we introduce the interest distance, and formulate the problem to a maximum reward problem.

B. Maximum coverage

Our maximum reward content distribution problem is similar to the maximum coverage problem in [13]. This is a classic question in computer science and computational complexity theory. The maximum coverage problem is NP-hard, and has been widely studied in approximation algorithms. Recent research on this topic has proposed budgeted maximum coverage [14] and generalized maximum coverage [15]. The budgeted maximum coverage problem is given a collection S of sets, with associated costs, defined over a domain of weighted elements, and a budget L; find a subset of $S' \subseteq S$ such that the total cost of sets in S' does not exceed L, and the total weight of elements covered by S' is maximized. The generalized maximum coverage problem is an extension of the former one. It has important applications in wireless OFDMA scheduling. In this paper, we prove our optimization problem's NP-hardness by reducing the weighted maximum coverage problem to a finite version of our target problem.

C. Smallest circle problem

In our greedy algorithms, we will use smallest circle solution to find the smallest circle to cover the points in the 2-D space. The smallest circle problem was initially proposed by the English mathematician Sylvester in 1857 [16]. The smallest circle problem in the plane is an example of a facility location problem, in which the location of a new facility must be chosen to provide service to a number of customers, minimizing the farthest distance that any customer must travel to reach the new facility [17]. As Megiddo [18] showed, the minimum enclosing circle can be found in linear time, and the same linear time bound also applies to the smallest enclosing sphere in Euclidean spaces of any constant dimension. Welzl [19] proposed a simple randomized algorithm for the minimum covering circle problem that runs in expected $O(n)$ time, based on a linear programming algorithm of Seidel [20].

III. Preliminary

A. Problem definition

Suppose there are n points in a 2-D space to be covered by k circles of radius r. Each point i has a maximum reward of w_i. A point will return a reward if it is covered by a circle. The reward can be thought of as the satisfaction of the nodes by receiving the content. The amount of reward is based on the distance between the point and the center of the circle. A point i can return multiple rewards, but not exceeding w_i, if it is covered by multiple circles. Note that a larger value of k tends to have a higher average of satisfiability, but it will also have less frequent service in a time-slotted content distribution system.

B. P-norm

The interest distance between content and an interest can be calculated in a general p-norm. In linear algebra, functional analysis and related areas of mathematics, a norm is a function that assigns a strictly positive length or size to all of the vectors in a vector space, other than the zero vector [21].

$$\|x\|_p = \left(\sum_{i=1}^{m} |x_i|^p \right)^{1/p}.$$

In this paper, we will just focus on the 1-norm and the 2-norm conditions. 1-norm is also called Taxicab norm or Manhattan norm. The name relates to the distance a taxi has to drive in a rectangular street grid to get from the origin to the point x: $\|x\|_1 = \sum_{i=1}^{m} |x_i|$. 2-norm is called Euclidean norm: $\|x\|_2 = \sqrt{x_1^2 + \cdots + x_m^2}$. This gives the ordinary distance from the origin to the point x: a consequence of the Pythagorean theorem.

IV. Optimization problem

In this section, we first formulate the objective problem into an optimization problem and prove its NP-hardness. Then we introduce a round-based heuristic algorithm, assuming that the subproblem at each round can be solved optimally. An approximation ratio for this algorithm is also derived. Finally, we show that finding the optimal solution within a round is also an NP-hard problem.

A. Optimization problem

As shown in Fig. 1, c_j is the center j, which has coverage range r. x_i is the point i and $d(c_j, x_i)$ is the interest distance between x_i and c_j. If x_i is in c_j's coverage range, the reward that point x_i received is the inverse of the interest distance between x_i and c_j. Otherwise, point x_i cannot get any reward. The problem can be presented as the equation below:

$$\psi(c_j, x_i) = \begin{cases} w_i(1 - \frac{d(c_j,x_i)}{r}) & d(c_j, x_i) \le r \\ 0 & otherwise \end{cases} \tag{1}$$

One point can belong to different centers, but its received reward can not exceed the maximum reward. Hence,

$$g(c_i) = \sum_{j=1}^{k} \psi(c_j, x_i). \tag{2}$$

$$f(x_i) = \begin{cases} g(x_i), & g(x_i) \le w_i \\ w_i, & otherwise \end{cases} \tag{3}$$

Let $[\cdot]_+$ be a function defined as $[\cdot]_+ = max(\cdot, 0)$. This target problem can be formulated as the following maximization problem:

$$maximize \quad \sum_{i=1}^{n} f(x_i). \tag{4}$$

Then an equivalent equation can be formulated:

$$\max_{\{c_j\}_{j=1}^{k}} \sum_i w_i \; min\left(\sum_j max\left(1 - \frac{d(c_j, x_i)}{r}, 0\right), 1\right). \tag{5}$$

Our target problem can be formulated as the following maximization optimization: problem

$$\max_{C:|C|=k, C\subset \mathcal{V}} f(C) \tag{6}$$

where $\mathcal{V}$ is an infinite set of indices that point to the real vectors in $\mathbb{R}^m$ space, and C is a subset of $\mathcal{V}$. For simplicity, from now on we will directly view $\mathcal{V}$ as an infinite set of real vectors in $\mathbb{R}^m$ space, and represent C as a subset of real vectors in $\mathbb{R}^m$: $C = \{c_1, \ldots, c_k\}$. The objective function in (6) is a function of C, expressed as:

$$f(C) = \sum_{i=1}^{n} w_i \min\left\{\sum_{j=1}^{k}\left[1 - \frac{d(c_j, x_i)}{r}\right]_+, 1\right\} \tag{7}$$

Theorem 0. *The target optimization problem (6) is an NP-hard problem.*

Proof: It is well known that maximizing a submodular function subject to a size constraint is NP-hard [22]. Below we will show that this objective function is a submodular function by proving two lemmas. ■

Lemma 0a. *Given real numbers $a \ge 0$, $b \ge 0$, and $y \ge 0$, we have:*

$$\begin{aligned} g &= \min\{y + a, 1\} - \min\{a, 1\} \\ &- \min\{y + a + b, 1\} + \min\{a + b, 1\} \ge 0. \end{aligned} \tag{8}$$

Proof: Considering different values for a, b and y, the problem above can be solved in three cases. We give the proof below for each case.

- Case 1 — $y + a + b < 1$:
  we have $g = (y+a) - a - (y+a+b) + (a+b) = 0 \ge 0$.
- Case 2 — $y + a + b \ge 1$, $y + a \le 1$:
  if $a + b \le 1$, we have $g = (y + a) - a - 1 + (a + b) = y + a + b - 1 \ge 0$;
  if $a + b > 1$, we have $g = (y + a) - a - 1 + 1 = y \ge 0$.
- Case 3 — $y + a > 1$:
  if $a \le 1$ and $a+b \le 1$, we have $g = 1-a-1+(a+b) = b \ge 0$;
  if $a \le 1$ and $a + b > 1$, we have $g = 1 - a - 1 + 1 = 1 - a \ge 0$;
  if $a > 1$, we have $g = 1 - 1 - 1 + 1 = 0 \ge 0$;

By combing these three cases, (8) is proved. ■

Lemma 0b. *Let $f(\emptyset) = 0$; then the function $f(C)$, defined in (7), is a submodular function.*

Proof: Consider any two subsets $A, B \subseteq \mathcal{V}$ and $A \subset B$. Without loss of generality, we can assume $A = \{a_1, \ldots, a_k\}$ and $B = \{a_1, \ldots, a_k, b_1, \ldots, b_{k'}\}$ for $k \ge 0$ and $k' > 0$. Then for any $s \in \mathcal{V}$ and $s \notin B$, we have:

$$\Big(f(A \cup \{s\}) - f(A)\Big) - \Big(f(B \cup \{s\}) - f(B)\Big) = \sum_{i=1}^{n} w_i \nabla R_i$$

where:

$$\begin{aligned} \nabla R_i &= \min\left\{\left[1 - \frac{d(s, x_i)}{r}\right]_+ + \sum_{j=1}^{k}\left[1 - \frac{d(a_j, x_i)}{r}\right]_+, 1\right\} \\ &- \min\left\{\sum_{j=1}^{k}\left[1 - \frac{d(a_j, x_i)}{r}\right]_+, 1\right\} \\ &- \min\left\{\left[1 - \frac{d(s, x_i)}{r}\right]_+ + \sum_{j=1}^{k}\left[1 - \frac{d(a_j, x_i)}{r}\right]_+\right. \\ &+ \left.\sum_{j=1}^{k'}\left[1 - \frac{d(b_j, x_i)}{r}\right]_+, 1\right\} \\ &+ \min\left\{\sum_{j=1}^{k}\left[1 - \frac{d(a_j, x_i)}{r}\right]_+ + \sum_{j=1}^{k'}\left[1 - \frac{d(b_j, x_i)}{r}\right]_+, 1\right\} \end{aligned} \tag{9}$$

According to Lemma 0a, $\nabla R_i \ge 0$. Therefore, $f(A \cup \{s\}) - f(A) \ge f(B \cup \{s\}) - f(B)$, and function $f(C)$ is a submodular function [22]. ■

It is well known that maximizing a submodular function subject to a size constraint is NP-hard [22]. Thus our target maximization problem is an NP-hard problem.

B. Round-based heuristic algorithm

Because this is an NP-hard optimization problem, we introduce a round-based heuristic algorithm. There are k rounds (k is same as the number of centers), in each round the reward value can be optimized. The round-based heuristic algorithm is described in Algorithm 1.

Algorithm 1 Round-based Heuristic Algorithm

let $y_i^1 = 1$ for $i = 1 \dots n$.
for $j = 1$ to k **do**
  $\{g(j), c_j, z_1^j, \dots, z_n^j\} \leftarrow$ maximizing $\sum_{i=1}^n w_i z_i^j$ by solving Equation (10).
  update y_i^{j+1}=$y_i^j - z_i^j$, for $i = 1 \dots n$.
end for

In Algorithm 1, the optimization problem involved in the jth round is as follows:

$$\begin{aligned} g(j) = & \max_{c_j} && \sum_{i=1}^n w_i \min\left\{\left[1 - \frac{d(c_j, x_i)}{r}\right]_+, y_i^j\right\} \\ = & \max_{c_j, z_1^j, \dots, z_n^j} && \sum_{i=1}^n w_i z_i^j \qquad (10) \\ & \text{s.t.} && z_i^j \le y_i^j, \forall i = 1, \dots, n \\ & && z_i^j \le \left[1 - \frac{d(c_j, x_i)}{r}\right]_+, \forall i = 1, \dots, n \end{aligned}$$

The optimization problem Equation (10) is equivalent to the following optimization problem.

$$\begin{aligned} \max_{c_j, z_1^j, \dots, z_n^j, s_1, \dots, s_n} \quad & \sum_{i=1}^n w_i s_i z_i^j \qquad (11) \\ \text{s.t.} \quad & z_i^j \le y_i^j, \forall i = 1, \dots, n \\ & z_i^j \le 1 - \frac{d(c_j, x_i)}{r}, \forall i = 1, \dots, n \\ & 0 \le s_i \le 1, \forall i = 1, \dots, n \end{aligned}$$

It is straightforward to show that for any feasible solution $(c, z_1, \dots, z_n)$ of (10), we can simply construct s_i by setting it to 1 if $1 > d(c, x_i)/r$ and to 0 otherwise. Then $(c, z_1, \dots, z_n, s_1, \dots, s_n)$ is one feasible solution for (11) with the same objective value. On the other hand, for any optimal solution $(c^*, z_1^*, \dots, z_n^*, s_1^*, \dots, s_n^*)$ of (11), it is straightforward to see that $(c^*, z_1^*, \dots, z_n^*)$ is one feasible solution for (10) with the same objective value, and therefore is also an optimal solution of (10).

The optimization problem (11) is a quadratic programming. Rewrite its objective in a quadratic form $\alpha Q \alpha^\top$, where $\alpha = [z_1^j, \dots, z_n^j, s_1, \dots, s_n, c_j]$ and:

$$Q = \begin{bmatrix} zeros(n,n) & 0.5diag(\mathbf{w}) & zeros(n,1) \\ 0.5diag(\mathbf{w}) & zeros(n,n) & zeros(n,1) \\ zeros(1,n) & zeros(1,n) & zeros(1,1) \end{bmatrix} \qquad (12)$$

for **w**=$[w_1, \dots, w_n]$. It is easy to see that Q has both positive and negative eigenvalues. Thus Equation (11) is NP-hard according to [23], and Equation (10) is NP-hard as well accordingly.

In the following, we present a theorem for the approximation ratio of the round-based heuristic algorithm.

Theorem 1. *Algorithm 1 (round-based heuristic) achieves an approximation ratio of* $1 - (1 - 1/k)^k$*, where* k *is the number of selected centers.*

We leave the proof of this theorem to the Appendix.

V. Greedy heuristic algorithm

Since the problem, as we discussed in the previous section, is NP-hard, here we will introduce three local greedy algorithms to solve the problem. The idea is to use the greedy approach at each point locally when seeking an optimal solution. To simplify the discussion, we use 2-D and 2-norm to illustrate. "The scope of a content" is represented by a 2-D disk.

We assume that there are *n* number of points in the whole network $\{x_1, x_2, ..., x_n\}$, and each point's maximum reward $\{w_1, w_2, ..., w_n\}$. We need to find k number of disks to cover them. The radius of the disks is predefined as r. Our objective is to find the "best" disks which can get the largest reward as we discussed in the previous section.

A. Local greedy algorithm

The approximation ratio we obtained in Theorem 1 is for the round-based heuristic algorithm; Equation (10) is still an NP-hard problem. Here we solve a local greedy algorithm instead, which picks the point that leads to the largest reward as the center in each round. This algorithm is shown in Algorithm 2.

Algorithm 2 Local Greedy Algorithm

let $y_i^1 = 1$ for $i = 1 \dots n$.
for $j = 1$ to k **do**
  $\{g(j), c_j, z_1^j, \dots, z_n^j\} \leftarrow$ maximizing $\sum_{i=1}^n w_i z_i^j$ by solving Equation (13).
  update y_i^{j+1}=$y_i^j - z_i^j$, for $i = 1 \dots n$.
end for

In Algorithm 2, the optimization problem involved in the jth round is as follows:

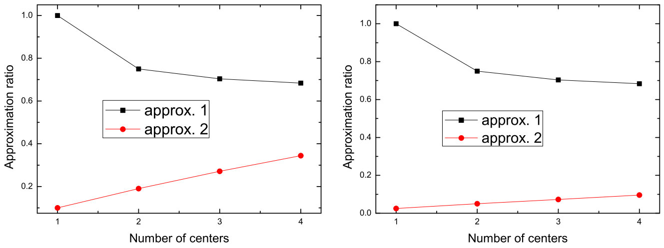
$$\begin{aligned} g(j) & = \max_{c_j \in \{x_1, \dots, x_n\}} \sum_{i=1}^n w_i \min\left\{\left[1 - \tfrac{d(c_j, x_i)}{r}\right]_+, y_i^j\right\} \\ & = \max_{c_j \in \{x_1, \dots, x_n\}, z_1^j, \dots, z_n^j} \sum_{i=1}^n w_i z_i^j \qquad (13) \\ \text{s.t.} \quad & z_i^j \le y_i^j, \forall i = 1, \dots, n \\ & z_i^j \le \left[1 - \tfrac{d(c_j, x_i)}{r}\right]_+, \forall i = 1, \dots, n \end{aligned}$$

In each round of Algorithm 2, all of the points are candidates for the center of the disk. The point that gives the maximum *coverage reward*, which includes all of the rewards contributed by the points covered by this disk, will be selected as the center of the disk. If there are a number of points, which have the same maximum coverage reward, our selection will be based on the index of the points (index refers to the ID of the point).

The approximation ratio of this local greedy algorithm is given in Theorem 2.

Theorem 2. *Algorithm 2 (local greedy) achieves an approximation ratio of* $1 - (1 - 1/n)^k$*, where* k *is the number of selected centers and* n *is the number of points.* $n > k$ *is assumed by default.*

We leave the proof of this theorem to the Appendix.



(a) 10-node (b) 40-node

Fig. 2. Comparison of approximation ratio in 10-node and 40-node environments.

In each round, it takes up to n steps to select the center point in the approach, and there are up to n steps to get the coverage reward by each candidate center. Hence, the complexity of this algorithm is $O(kn^2)$.

B. Alternative local greedy algorithms

In Fig. 2, approx. 1 is the approximation ratio $1-(1-1/k)^k$ from Theorem 1 using Algorithm 1, and approx. 2 in the figures is the approximation ratio $1-(1-1/n)^k$ from Theorem 2 using Algorithm 2, where k is the number of centers and n is the number of points. We compare these two approximation ratios in 10-node and 40-node environments. We can see that approx. 1 is much smaller than approx. 2. In the following, we propose the other two algorithms: one reduces the asymptotic complexity (simple local greedy algorithm) and another one increases the approximation ratio (complex local greedy algorithm).

Simple local greedy algorithm: picks the largest single reward point as the center in each round, which is described in Algorithm 3:

Algorithm 3 Simple Local Greedy Algorithm

let $y_i^1 = 1$ for $i = 1 \ldots n$.
for $j = 1$ to k **do**
  $c_j \leftarrow x_{i^*}$ for $i^* = \arg\max_i w_i y_i^j$.
  $\{g(j), z_1^j, \ldots, z_n^j\} \leftarrow$ maximizing $\sum_{i=1}^n w_i z_i^j$ by solving Equation (14).
  update y_i^{j+1}=$y_i^j - z_i^j$, for $i = 1 \ldots n$.
end for

In Algorithm 3, the optimization problem involved in the jth round is as follows:

$$\begin{aligned} g(j) = & \quad \sum_{i=1}^{n} w_i \min\left\{\left[1-\frac{d(c_j,x_i)}{r}\right]_+, y_i^j\right\} \\ = \quad & \max_{z_1^j,\ldots,z_n^j} \quad \sum_{i=1}^{n} w_i z_i^j \\ & \text{s.t.} \quad z_i^j \le y_i^j, \forall i = 1,\ldots,n \\ & \qquad z_i^j \le \left[1-\frac{d(c_j,x_i)}{r}\right]_+, \forall i=1,\ldots,n \end{aligned} \tag{14}$$

In each round, we will pick the point with the maximum *single-point reward* as the center of the disk. If there are a number of points, which have the same maximum single-point reward, our selection will be based on the index of the nodes.

Theorem 3. *The complexity of Algorithm 3 (simple local greedy) is $O(kn)$.*

Proof: In each round, there are up to n steps to get the coverage reward by the selected center. Hence, the complexity of this algorithm is $O(kn)$. ■

It is easy to show that the approximation ratio in Theorem 3 still holds for the simple local greedy algorithm.

Complex local greedy algorithm: the complex local greedy algorithm is described in Algorithm 4:

Algorithm 4 Complex Local Greedy Algorithm

let $y_i^1 = 1$ for $i = 1 \ldots n$.
for $j = 1$ to k **do**
  for $i = 1$ to n **do**
    Initially $x_i^1 \leftarrow x_i$.
    update x_i^{l+1}= new-center (x_i^l), for $l = 1 \ldots (n-1)$.
  end for
  $\{g(j), c_j, z_1^j, \ldots, z_n^j\} \leftarrow$ maximizing $\sum_{i=1}^n w_i z_i^j$ by solving Equation (15).
  update y_i^{j+1}=$y_i^j - z_i^j$, for $i = 1 \ldots n$.
end for

In Algorithm 4, the optimization problem involved in the jth round is as follows:

$$\begin{aligned} g(j) \quad & = \max_{c_j \in \{x_1^n,\ldots,x_n^n\}} \sum_{i=1}^n w_i \min\left\{\left[1-\frac{d(c_j,x_i)}{r}\right]_+, y_i^j\right\} \\ & = \max_{c_j \in \{x_1^n,\ldots,x_n^n\}, z_1^j,\ldots,z_n^j} \sum_{i=1}^n w_i z_i^j \\ \text{s.t.} \quad & \qquad z_i^j \le y_i^j, \forall i = 1,\ldots,n \\ & \qquad z_i^j \le \left[1-\frac{d(c_j,x_i)}{r}\right]_+, \forall i=1,\ldots,n \end{aligned} \tag{15}$$

We use a 2-D disk to illustrate the new-center (x_i^l) in a 2-D and 2-norm system:

1) Start with the disk D centered at an x_i^l.
2) Consider the remaining heaviest point j (i.e., $\max w_j z_j^l$).
3) If j is outside D, return the center of D and stop.
4) Otherwise, define the center for the new disk D' by including j in D. This center is the smallest disk that covers all points in D plus point j.
5) If the coverage reward of D' is larger than the coverage reward of D, then return the center of D'; otherwise, return the center of D.

In Algorithm 4, $g(j)$ is the coverage reward. It is so presented to keep all of the algorithms in a uniform way. The new center (x_i^l) in 1-norm can be easily calculated through projections on different dimensions. Note that the major difference here is that the position of a center can be anywhere in the complex local greedy algorithm.

The complexity for this greedy heuristic algorithm is described in Theorem 4:

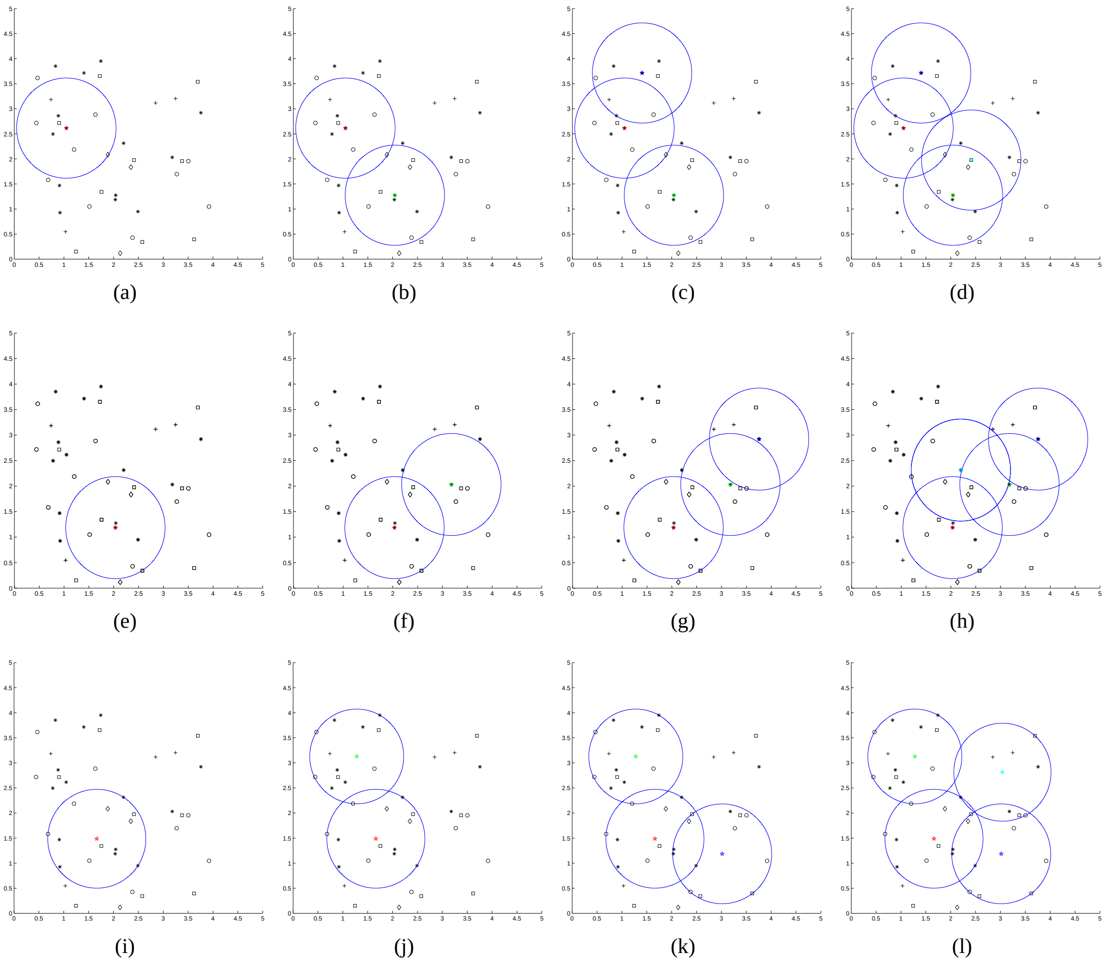


Fig. 3. Greedy algorithms: greedy 2: (a) - (d); greedy 3: (e) - (h); greedy 4: (i) - (l) (different symbols of the points mean different weights: 5: $*$; 4: $\square$; 3: $\diamond$; 2: $+$; 1: $\bigcirc$. $\star$ is the centers).

Coverage reward	1	2	3	4	Total
Greedy 2	14.3145	11.2969	9.778	9.2406	44.6301
Greedy 3	11.2969	9.7395	8.4636	8.3435	37.8435
Greedy 4	20.3867	19.1588	13.5481	10.4635	63.5571

TABLE I
COMPARISON OF THREE GREEDY HEURISTIC ALGORITHMS.

Theorem 4. *The complexity of Algorithm 4 (complex local greedy) is $O(kn^3)$ for 2-norm in a 2-D space. The complexity is $O(kmn^3)$ for 1-norm in the m-D space.*

Proof: For 2-norm in a 2-D space, there are k rounds in the outer loop. Also, each point (of n) performs (1), (2) and (3), that we discussed above, of the following:

Suppose the size of D is i in the current round. (2) takes $(n-i)$ steps. (3) consumes $(i+1)$ steps (to find the smallest disk to cover D and the newly selected point). Up to n rounds of (2) and (3) have a total of n^2 steps. Therefore, the overall complexity is $O(kn^3)$.

For 1-norm in the m-D space, in each round, up to n^2 steps are needed as all remain the same except (3), which consumes $m \cdot (i+1)$. Along each dimension, the the boundary can be determined through a projection on the dimension. The min and max values are determined. The center position along this dimension is $(\min + \max)/2$. Therefore, its complexity is $O(kmn^3)$. ■

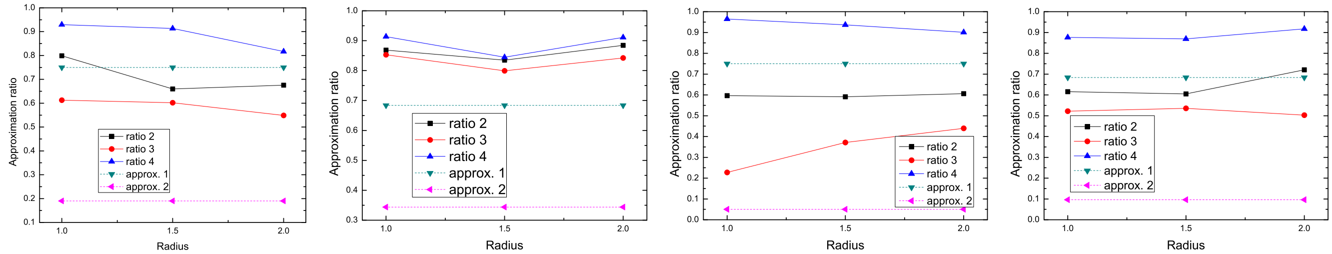
The approximation ratio for the complex local greedy algorithm is still an open problem.

C. *p-norm in the m-D space*

Previously, we used 2-norm and a 2-D space to explain our algorithms. The disk and its radius we mentioned are based on 2-norm to calculate the interest distance between two points in a 2-D space.

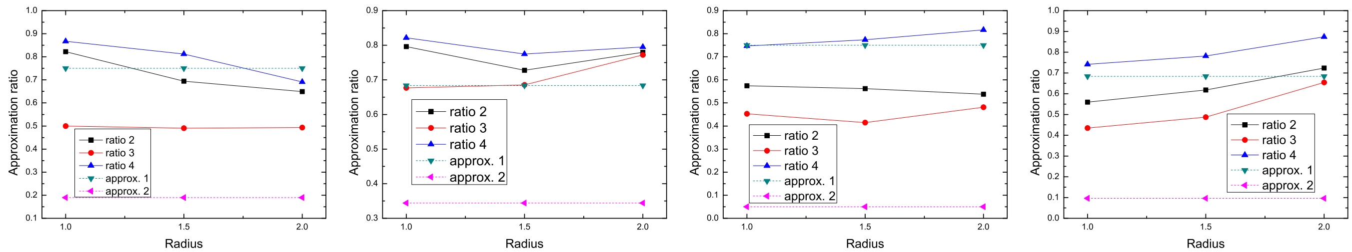
In the 2-D space, if we use the 1-norm to calculate the interest distance, we can use a square to cover the points instead of the disk, and the "radius" would then be the square's side length.

In the 3-D space, in a 1-norm system, we use a cube to cover the points, while in a 2-norm system, we use a ball to cover the points.



(a) 10-node with 2 centroids (b) 10-node with 4 centroids (c) 40-node with 2 centroids (d) 40-node with 4 centroids

Fig. 4. Comparison in 2-norm in a 2-D space using different weights.



(a) 10-node with 2 centroids (b) 10-node with 4 centroids (c) 40-node with 2 centroids (d) 40-node with 4 centroids

Fig. 5. Comparison in 2-norm in a 2-D space using the same weight.

VI. Simulation

In this section, we compare the performance of our greedy algorithms. The metric we used is the approximation ratio. This is the ratio of our greedy algorithms' reward and the exhaustive reward.

A. Simulation methods and setting

We randomly put the nodes into a (4×4) 2-D space and a $(4 \times 4 \times 4)$ 3-D space. In a 2-D space, we have 10-node and 40-node environments. In a 3-D space, we have 40-node and 160-node environments.

The weight of each node is an integer. We also have two schemes: one is that the weight of each node is the same, which is 1; another is that the weight of each node is different, which is a random integer between 1 and 5.

In our simulation, the calculation of the interest distance is based on 1-norm and 2-norm. We compare the greedy algorithms in different number of centers (2, 4) and different radius of the centers (1, 1.5, 2).

We denote that the original local greedy algorithm (Algorithm 2) is called greedy 2, the simple local greedy algorithm (Algorithm 3) is called greedy 3, and the complex local greedy algorithm (Algorithm 4) is called greedy 4.

In the 2-D space comparison, ratios 2, 3, and 4 present the approximation ratio among these three greedy algorithms (Algorithms 2, 3, and 4) with the exhaustive optimal solution, respectively. We also compare these three ratios with approx. 1 from Theorem 1 and approx. 2 from Theorem 2. Note that approx. 2 is the worst case for Algorithm 2. Therefore it should correspond to the smallest number (in ratio). Approx. 1 is the worst case for an iterative approach with the optimal local solution. In the 3-D space comparison, we just compare these three greedy algorithms' gained rewards.

B. Simulation results

In this section, we will discuss the simulation results using the greedy algorithms we mentioned in the previous sections.

1) Example: First, we give an example to illustrate how these three local greedy algorithms work.

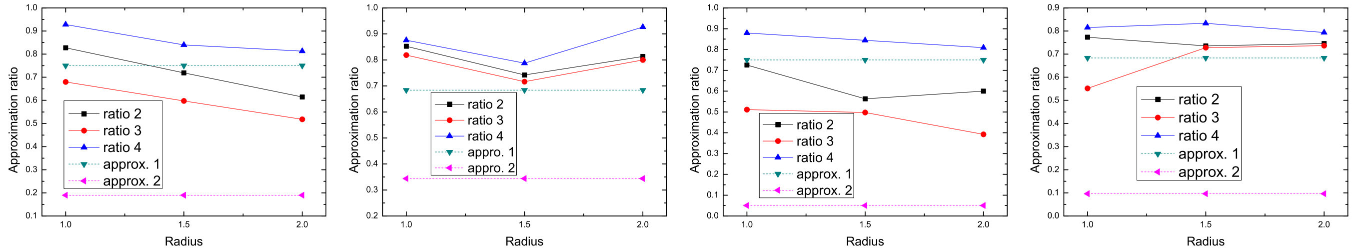
In Figs. 3, we implement our three local greedy algorithms in a (4×4) 2-D space with 40 nodes based on 2-norm for interest distance calculation. Table I shows the coverage reward gain in each round using these algorithms. It is clear that the third one is much better than other two in each round.

2) Results in a 2-D space using 2-norm: in a 2-D space, we first use 2-norm to calculate the interest distance between two points. We compare the approximation ratio among these three algorithms with the exhaustive solution, as shown in Figs. 4 and 5.

From Figs. 4 and 5, we can see that our proposed three local greedy algorithms' approximation ratios are all larger than approx. 2. This validates Theorem 2. Greedy 3 is better than the other two, and its approximation ratios are above approx. 1 most times in different conditions.

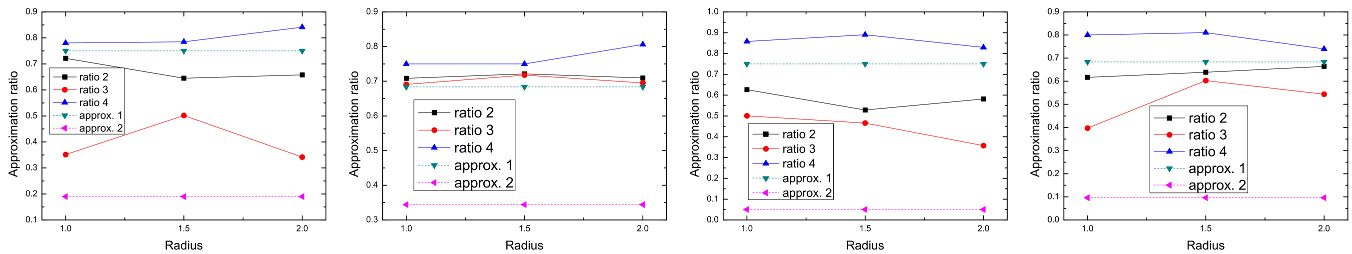
Overall, in a 2-D space for 2-norm, we find that with greedy 3, the approximation ratio is about 84.22% which is the best out of all three of the greedy algorithms. Greedy 1's the approximation ratio is about 68.87% and approximation ratio for greedy 2 is about 55.97%.

3) Results in a 2-D space using 1-norm: In a 2-D space, we use 1-norm to calculate the interest distance between two nodes. We compare the approximation ratio among these three algorithms with the exhaustive solution.



(a) 10-node with 2 centroids (b) 10-node with 4 centroids (c) 40-node with 2 centroids (d) 40-node with 4 centroids

Fig. 6. Comparison in 1-norm in a 2-D space using different weights.



(a) 10-node with 2 centroids (b) 10-node with 4 centroids (c) 40-node with 2 centroids (d) 40-node with 4 centroids

Fig. 7. Comparison in 1-norm in a 2-D space using the same weight.

From Figs. 6 and 7, we find that with greedy 3, the approximation ratio is about 82.76% which is the best out of all three of the greedy algorithms. Greedy 1's approximation ratio is about 68.77% and the approximation ratio for greedy 2 is about 57%.

4) Results in a 3-D space using 1-norm: in a 3-D space, we use 1-norm to calculate the interest distance between two nodes. We compare the total reward that these three greedy algorithms gain with same weight and different weight schemes, as shown in Figs. 8 and 9.

We find using greedy 3 will get the highest reward. Using greedy 1 gets about 61.04% of the reward that greedy 3 gets, and greedy 2 gets about 31.14%.

C. Summary of simulation

We use these three greedy algorithms to calculate the approximation value for the optimization problem. In 2-D and 3-D spaces for 1-norm and 2-norm, we can see that greedy 3 gets the best results. Its approximation ratio is above 80%, which is higher than Theorem 1's approximation ratio, which considers that each round is optimal. Greedy 1's approximation ratio is larger than 60%, while greedy 2's approximation ratio is about 56% in a 2-D space and 31% in a 3-D space. These two greedy algorithms' results also reflect the analytical results (Theorem 2). Although our optimization problem is an NP-hard problem, our proposed greedy 3 still gets an acceptable approximation ratio.

VII. Conclusion

In this paper, we studied the content distribution in wireless networks. We first formulated the problem into an optimal content distribution problem and proved it as an NP-hard problem. Three greedy algorithms have been proposed to solve the optimal problem. The greedy algorithms can be implemented in the m-D space using p-norm to calculate the interest distance. We analyzed the approximation ratio of the greedy algorithms and its complexity. We then turned to studying the performance of these three local greedy algorithms in 2-D and 3-D space with 1-norm and 2-norm. The simulation results have shown that our proposed algorithms perform well. To the best of our knowledge, this is the first study on the optimal content distribution problem in wireless networks.

Acknowledgments

This research was supported in part by NSF grants CCF 1028167, CNS 0948184, and CCF 0830289.

References

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2508–2530, June 2006.

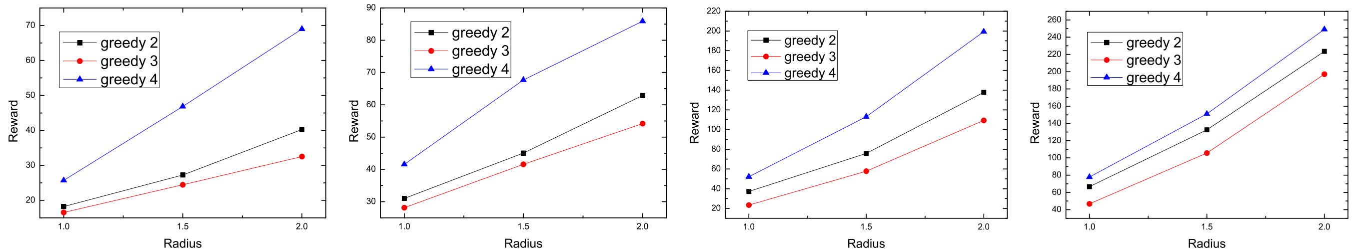
[2] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: enhancing mobile networks with infrastructure," in *Proc. of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2008, pp. 81–91.

[3] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic, "The age of gossip: spatial mean field regime," in *Proc. of the 11th ACM International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2009, pp. 109–120.

[4] P. Denantes, F. Bnzit, P. Thiran, and M. Vetterli, "Which distributed averaging algorithm should I choose for my sensor network," in *Proc. of the 27th IEEE Conf. Computer Communications and Networks (INFOCOM)*, 2008.

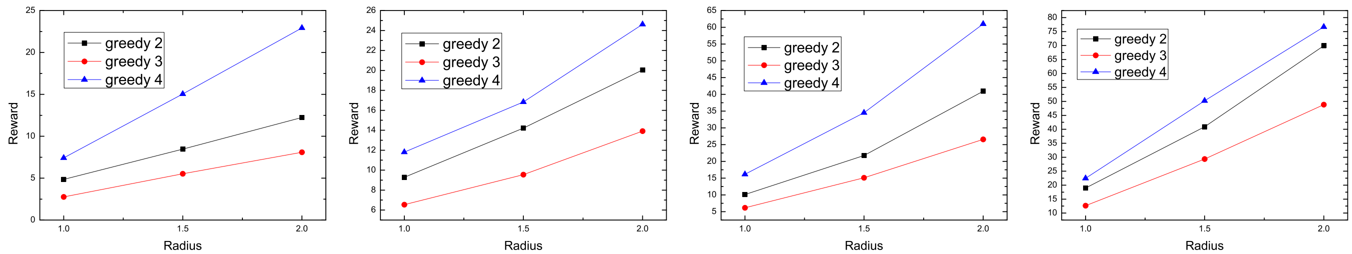
[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. of ACM SIGCOMM*, 2001, pp. 149–160.

[6] B. Yang and H. Garcia-molina, "Improving search in peer-to-peer networks," in *Proc. of the 22nd International Conference on Distributed Computing Systems (ICDCS)*, 2002, pp. 5–14.



(a) 10-node with 2 centroids (b) 10-node with 4 centroids (c) 40-node with 2 centroids (d) 40-node with 4 centroids

Fig. 8. Comparison in 1-norm in a 3-D space using different weights.



(a) 10-node with 2 centroids (b) 10-node with 4 centroids (c) 40-node with 2 centroids (d) 40-node with 4 centroids

Fig. 9. Comparison in 1-norm in a 3-D space using the same weight.

[7] Y. Chen, Y. H. Katz, and J. D. Kubiatowicz, "Scan: A dynamic, scalable, and efficient content distribution network," in *Proc. of the IEEE International Conference on Pervasive Computing*, 2002.

[8] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 20, pp. 1489–1499, 2002.

[9] S.-H. Lee, U. Lee, K.-W. Lee, and M. Gerla, "Content distribution in vanets using network coding: The effect of disk i/o and processing o/h," in *Proc. of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2008.

[10] B. Molina, S. F. Pileggi, C. E. Palau, and M. Esteve, "A social framework for content distribution in mobile transient networks," in *Proc. of the 3rd ACM International Workshop on Use of P2P, Grid and Agents for the Development of Content Networks*, 2008, pp. 29–36.

[11] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proc. of the 21st IEEE Conf. Computer Communications and Networks (INFOCOM)*, 2002.

[12] A. Klemm, C. Lindemann, and O. P. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," in *Proc. of the IEEE Vehicular Technology Conference: Symposium on Data Base Management in Wireless Network Environments (VTC)*, 2003.

[13] D. S. Hochbaum, *Approximating Covering and Packing Problems: Set cover, Vertex cover, Independent Set, and Related Problems.* Approximation Algorithms for NP-hard Problems, 1997, pp. 94–143.

[14] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, pp. 39–45, April 1999.

[15] R. Cohen and L. Katzir, "The generalized maximum coverage problem," *Inf. Process. Lett.*, vol. 108, pp. 15–22, September 2008.

[16] J. J. Sylvester, "A question in the geometry of situation," *Quarterly Journal of Mathemaitics*, vol. 1, p. 79, 1857.

[17] R. Francis, L. F. McGinnis, and J. White, *Facility Layout and Location: An Analytical Approach, Second edition.* Prentice-Hall, Inc., 1992.

[18] N. Megiddo, "Linear-time algorithms for linear programming in r3 and related problems," *SIAM Journal on Computing*, vol. 12, pp. 759–776, 1983.

[19] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," *Results and New Trends in Computer Science*, vol. 555, pp. 359–370, 1991.

[20] R. Seidel, "Linear programming and convex hulls made easy," in *Proc. of the 6th ACM annual Symposium on Computational Geometry*, 1990, pp. 211–215.

[21] N. Bourbaki, *Topological Vector Spaces*. Springer, 1987.

[22] U. Feige and V. S. Mirrokni, "Maximizing non-monotone submodular functions," in *In Proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS*, 2007, p. 2007.

[23] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *Journal of Global Optimization*, vol. 1, no. 1, pp. 15–22, 1991.

APPENDIX

In the appendix, we will present the proof of Theorem 1 in Section IV and Theorem 2 in Section V.

Proof of Theorem 1: We will prove Theorem 1 by proving the following Lemma.

Lemma 1. *Let $g(j) = \sum_{i=1}^{n} w_i z_i^j$ be the optimal objective function value for the optimization problem (10) in the jth round of the round-based heuristic algorithm. We then have the following results: (a) $g(1) \geq \frac{1}{k} f_{opt}$; (b) $g(j) \geq \frac{1}{k}(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1))$, for $j = 2 \ldots k$.*

Proof: The proof for (a) is straightforward. From the round-based heuristic algorithm, we can see that $kg(1)$ can be obtained by:

$$\begin{aligned} kg(1) &= k \max_{c_1} \sum_{i=1}^{n} w_i \min\left\{\left[1 - \frac{d(c_1, x_i)}{r}\right]_+, 1\right\} \\ &= \max_{c_1,\ldots,c_j} \sum_{i=1}^{n} w_i \sum_{j=1}^{k} \min\left\{\left[1 - \frac{d(c_j, x_i)}{r}\right]_+, 1\right\} \\ &\geq \max_{c_1,\ldots,c_j} \sum_{i=1}^{n} w_i \min\left\{\sum_{j=1}^{k}\left[1 - \frac{d(c_j, x_i)}{r}\right]_+, 1\right\} \\ &= f_{opt} \end{aligned} \tag{16}$$

Thus (a) $g(1) \geq \frac{1}{k} f_{opt}$ is proved.

Let the reward obtained on point i from the first $j-1$ rounds of the round-based heuristic algorithm is $h_i^{j-1} = \sum_{\ell=1}^{j-1} z_i^\ell$, thus $y_i^j = 1 - h_i^{j-1}$, and $\sum_{i=1}^n h_i^{j-1} = \sum_{\ell=1}^{j-1} g(\ell-1)$. Let $\{c_1^*, \ldots, c_k^*\}$ be the optimal solution returned by the optimization. Therefore:

$$\begin{aligned}
kg(j) &= k \max_{c_j} \sum_{i=1}^n w_i \min\left\{\left[1-\frac{d(c_j,x_i)}{r}\right]_+, y_i^j\right\} \\
&= \max_{c_{j_1},\ldots,c_{j_k}} \sum_{i=1}^n w_i \sum_{\ell=1}^k \min\left\{\left[1-\frac{d(c_{j_\ell},x_i)}{r}\right]_+, y_i^j\right\} \\
&\geq \sum_{i=1}^n w_i \sum_{j=1}^k \min\left\{\left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, y_i^j\right\} \\
&\geq \sum_{i=1}^n w_i \min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, y_i^j\right\} \\
&= \sum_{i=1}^n w_i \min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, 1-h_i^{j-1}\right\} \\
&\geq \sum_{i=1}^n w_i \left(\min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, 1\right\} - h_i^{j-1}\right) \\
&= f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1) \qquad (17)
\end{aligned}$$

Thus (b) $g(j) \geq \frac{1}{k}(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1))$ is proved. ■

Now let $f(j) = \sum_{\ell=1}^j g(j)$. Then $f(k)$ is the reward value, i.e. the sum of objective values in all k rounds, returned by the round-based heuristic algorithm above. According to Lemma 1, we have:

$$\begin{aligned}
f(1) &= g(1) \geq \frac{1}{k} f_{opt}; \qquad (18) \\
f(j) &= f(j-1) + [f(j) - f(j-1)] \\
&= f(j-1) + g(j) \\
&\geq f(j-1) + \frac{1}{k}(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1)) \\
&= f(j-1) + \frac{1}{k}(f_{opt} - f(j-1)) \\
&= (1-\frac{1}{k}) f(j-1) + \frac{1}{k} f_{opt}, \\
&\quad \text{for } j = 2, \ldots, k. \qquad (19)
\end{aligned}$$

Combining the base case inequation (18) and the recursive inequation (24), we can get:

$$f(k) \geq (1-(1-1/k)^k) f_{opt} \geq (1-1/e) f_{opt}. \qquad (20)$$

Therefore, the approximation ratio of the round-based heuristic algorithm is $(1-(1-1/k)^k)$, and it is bounded by $(1-1/e)$.

Proof of Theorem 2: We will prove Theorem 2 by proving the following Lemma.

Lemma 2. *Let $g(j) = \sum_{i=1}^n w_i z_i^j$ be the optimal objective function value for the optimization problem (13) in the jth round of the local greedy algorithm. We then have the following results: (a) $g(1) \geq \frac{1}{n} f_{opt}$; (b) $g(j) \geq \frac{1}{n}(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1))$, for $j = 2 \ldots k$.*

Proof: It is obvious $f_{opt} \leq \sum_{i=1}^n w_i$. The proof for (a) is straightforward. From the local greedy algorithm, we have $g(1) \geq \max_i w_i$. Therefore:

$$g(1) \geq \frac{\max_i w_i}{\sum_{i=1}^n w_i} f_{opt} \geq \frac{1}{n} f_{opt} \qquad (21)$$

Thus (a) $g(1) \geq \frac{1}{n} f_{opt}$ is proved.

Let the reward obtained on point i from the first $j-1$ rounds of the round-based heuristic algorithm is $h_i^{j-1} = \sum_{\ell=1}^{j-1} z_i^\ell$, thus $y_i^j = 1 - h_i^{j-1}$, and $\sum_{i=1}^n h_i^{j-1} = \sum_{\ell=1}^{j-1} g(\ell-1)$. Let $\{c_1^*, \ldots, c_k^*\}$ be the optimal solution returned by the optimization. Therefore:

$$\begin{aligned}
g(j) &= \max_{c_j \in \{x_1,\ldots,x_n\}} \sum_{i=1}^n w_i \min\left\{\left[1-\frac{d(c_j,x_i)}{r}\right]_+, y_i^j\right\} \\
&\geq \max_i w_i y_i^j \\
&\geq \max_i w_i \min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, y_i^j\right\} \\
&\geq \frac{1}{n} \sum_{i=1}^n w_i \min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, y_i^j\right\} \\
&= \frac{1}{n} \sum_{i=1}^n w_i \min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, 1-h_i^{j-1}\right\} \\
&\geq \frac{1}{n} \sum_{i=1}^n w_i \left(\min\left\{\sum_{j=1}^k \left[1-\frac{d(c_j^*,x_i)}{r}\right]_+, 1\right\} - h_i^{j-1}\right) \\
&= \frac{1}{n}\left(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1)\right) \qquad (22)
\end{aligned}$$

Thus (b) $g(j) \geq \frac{1}{n}(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1))$ is proved. ■

Now let $f(j) = \sum_{\ell=1}^j g(j)$. Then $f(k)$ is the reward value, i.e. the sum of objective values in all k iterations, returned by the local greedy algorithm above. According to Lemma 2, we have:

$$\begin{aligned}
f(1) &= g(1) \geq \frac{1}{n} f_{opt}; \qquad (23) \\
f(j) &= f(j-1) + [f(j) - f(j-1)] \\
&= f(j-1) + g(j) \\
&\geq f(j-1) + \frac{1}{n}(f_{opt} - \sum_{\ell=1}^{j-1} g(\ell-1)) \\
&= f(j-1) + \frac{1}{n}(f_{opt} - f(j-1)) \\
&= (1-\frac{1}{n}) f(j-1) + \frac{1}{n} f_{opt}, \\
&\quad \text{for } j = 2, \ldots, k. \qquad (24)
\end{aligned}$$

Combining the base case equation (23) and the recursive in equation (24), we can get:

$$f(j) \geq (1-(1-1/n)^j) f_{opt}. \qquad (25)$$

Thus:

$$f(k) \geq (1-(1-1/n)^k) f_{opt} \qquad (26)$$

and the approximation ratio of the local greedy algorithm is $(1-(1-1/n)^k)$.