An Input Sensitive Online Algorithm for the Metric Bipartite Matching Problem

Krati Nayyar
Department of Computer Science, Virginia Tech
Blacksburg, USA
krati14@vt.edu

Sharath Raghvendra
Department of Computer Science, Virginia Tech
Blacksburg, USA
sharathr@vt.edu

***Abstract*—We present a novel input sensitive analysis of a deterministic online algorithm [1] for the minimum metric bipartite matching problem. We show that, in the adversarial model, for any metric space $\mathbb{M}$ and a set of n servers S, the competitive ratio of this algorithm is $O(\mu_{\mathbb{M}}(S)\log^2 n)$; here $\mu_{\mathbb{M}}(S)$ is the maximum ratio of the traveling salesman tour and the diameter of any subset of S. It is straight-forward to show that any algorithm, even with complete knowledge of $\mathbb{M}$ and S, will have a competitive ratio of $\Omega(\mu_{\mathbb{M}}(S))$. So, the performance of this algorithm is sensitive to the input and near-optimal for any given S and $\mathbb{M}$. As consequences, we also achieve the following results:**

- **If S is a set of points on a line, then $\mu_{\mathbb{M}}(S) = \Theta(1)$ and the competitive ratio is $O(\log^2 n)$, and,**
- **If S is a set of points spanning a subspace with doubling dimension d, then $\mu_{\mathbb{M}}(S) = O(n^{1-1/d})$ and the competitive ratio is $O(n^{1-1/d}\log^2 n)$.**

Prior to this result, the previous best-known algorithm for the line metric has a competitive ratio of $O(n^{0.59})$ and requires both S and the request set R to be on a line. There is also an $O(\log n)$ competitive algorithm in the weaker oblivious adversary model.

To obtain our results, we partition the requests into well-separated clusters and replace each cluster with a small and a large weighted ball; the weight of a ball is the number of requests in the cluster. We show that the cost of the online matching can be expressed as the sum of the weight times radius of the smaller balls. We also show that the cost of edges of the optimal matching inside each larger ball can be shown to be proportional to the weight times the radius of the larger ball. We then use a simple variant of the well-known Vitali's covering lemma to relate the radii of these balls and obtain the competitive ratio.

***Keywords*-online algorithms; minimum metric matching; input sensitive analysis**

I. Introduction

Driven by consumers' demand for a quick access to their ordered products, business ventures schedule their delivery of goods and services in real-time, often without the complete knowledge of the future request locations or their order of arrival. Due to this lack of complete information, decisions made tend to be sub-optimal. Therefore, there is a need for robust and competitive *online algorithms* that immediately and irrevocably allocate resources to requests in real-time at minimal cost. These resources are servers placed in various locations S with $|S| = n$ of the city and each server has a capacity that restricts how many requests it can serve. When a new request $r \in R$ arrives, one of the servers which has a positive residual capacity is assigned to this request. After this request is served, the residual capacity of the server reduces by one. The cost associated with this assignment is a metric cost; for instance, it could be the minimum distance traveled by the server to reach the request.

The case where the capacity of every server is ∞ is the celebrated *k-server problem*. The case where every server has a capacity of 1 is the *metric bipartite matching problem*. In this case, the requests arrive one at a time and we have to immediately and irrevocably match it to some unmatched server. The resulting assignment is a matching and is referred to as an *online matching*. Finding a a minimum-cost online matching is impossible because an adversary can easily fill up the remaining locations of requests in R in a way that our current assignment becomes sub-optimal. Therefore, we want our algorithm to compute an online matching which is competitive. For any input S, R and any arrival order of requests in R, we say our algorithm is *α-competitive*, for $\alpha > 1$, when the cost of the online matching M is at most α times the minimum cost, i.e.,

$$w(M) \leq \alpha w(M_{\text{OPT}}).$$

Here M_{OPT} is the minimum-cost matching of the locations in S and R.

In the *adversarial model*, there is an adversary who knows the server locations and the assignments made by the algorithm and generates a sequence of requests to maximize α. Another well-studied model is the *random arrival model* [2] where the adversary chooses the set of request locations R before the algorithm executes but the arrival order is a permutation chosen uniformly at random from the set of all possible permutations of R. A popular model of theoretical interest is the *oblivious adversary* model. In this model, the adversary knows the algorithm and decides the request locations and their arrival order. However, the online algorithm is a randomized algorithm and the adversary does not know the random choices made by the algorithm. This model is weaker than the adversarial model but stronger than the random arrival model.

We would also like to note the role of the server locations and the metric space in the design of online algorithms for

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.53

these problems. Many algorithms are designed to work for any metric space and any set of server locations and they are analyzed based on a worst-case choice of metric space and server locations S. However, in practical scenarios, the subspace spanned by the server locations may be "nice" and may admit online algorithms that produce substantially better quality solutions. For example, if servers are restricted to be points on a line, one may expect better quality algorithms. It is, therefore, desirable to have algorithms that work optimally for any input metric space and any set of server locations. In this paper, we present such an algorithm for the minimum metric bipartite matching problem that is sensitive to the input and simultaneously achieves a near-optimal performance for every metric space.

Previous Work: Solutions for the k-server problem and the online bipartite matching problem use similar mathematical tools and methodologies. Both of these problems have been extensively studied in the adversarial model and the oblivious model. In the random arrival model, there is an online algorithm for the metric bipartite matching problem that was presented recently [1]. We do not know of any work for the k-server problem in the random arrival model.

The k-server problem is central to the theory of online algorithms. The problem was first posted by Manasse *et al.* [3]. In the adversarial model, the best-known deterministic algorithm for this problem is the $2k-1$-competitive work function algorithm [4]. In this problem, we assume there are k servers, each of which can serve arbitrary many of the n arriving requests. It is known that no deterministic algorithm can achieve a competitive ratio better than k in any metric space with at least $k+1$ points [3] and is conjectured that in fact there is a k-competitive algorithm for this problem. This conjecture is popularly called the *k-server conjecture*.

For the online metric bipartite matching problem, in the adversarial model, there is a $2n-1$-competitive deterministic algorithm by Khuller *et al.* [5] and Kalyanasundaram and Pruhs [6]. They also show that there is a metric space where no online algorithm can achieve a better competitive ratio in the adversarial model. For the line metric, i.e., when the requests and servers are points on a line, it is possible to achieve sub-linear competitive ratio of $O(n^{0.59})$ [7] . Khuller *et al.* [5] point to the possibility of better online algorithms in d-dimensional Euclidean spaces but leave this problem as open.

For the oblivious adversary, there are $\log^{O(1)} n$-competitive algorithms for both the k-server problem and the online metric bipartite matching problem. Bansal *et al.* [8] achieve a $O(\log^2 n)$-competitive algorithm for the metric bipartite matching problem. For the k-server problem, Bansal *et al.* [9] presented an $O(\text{polylog}\, n \log k)$-competitive algorithm. There is also an online algorithm [1] for metric bipartite matching problem that simultaneously achieves optimal competitive ratio of $2n-1$ and $2H_n - 1$ under the adversarial and the random arrival model respectively. Interestingly, the algorithm relies on a parameter $t > 1$ and the competitive ratio approaches optimal as t tends to ∞. In this paper, we will present an input sensitive analysis of this algorithm in the adversarial model. However, for our analysis, we need t to be a constant.

Also note that for metric with bounded doubling dimension, there is an a $O(d \log n)$-competitive algorithm in the oblivious model [10]; here d is the doubling dimension of the metric space. In the adversarial model, the question of finding a deterministic $O(1)$-competitive online algorithm for the line metric is an important open question; see [7], [11] for results on the special case of line metric.

Our Result: In this paper, we provide a new and fine-grained analysis of the deterministic online algorithm presented in [1] in the adversarial model. We refer to this deterministic algorithm as the robust matching or the RM-Algorithm. For a given metric space $\mathbb{M}$, we show that this algorithm achieves a competitive ratio of $O(\mu_{\mathbb{M}}(S)\log^2 n)$ in the adversarial model; here $\mu_{\mathbb{M}}(S)$ is the worst case ratio of the cost of the traveling salesman tour and the diameter of a subset of servers among all subsets S' of S that have a positive diameter. There is a straight-forward lower bound of $\Omega(\mu_{\mathbb{M}}(S))$ on the competitive ratio of any algorithm that knows the metric space $\mathbb{M}$ and the initial server configuration S and is designed to perform optimally for $\mathbb{M}$ and S. Therefore, while our algorithm is oblivious to the metric space $\mathbb{M}$ and also the server locations S, it nevertheless is near-optimal for $\mathbb{M}$ and S. As consequences, we obtain improved bounds on competitive ratio for the line metric and any metric space with low doubling dimension.

- Suppose the server locations S are points on a line, then $\mu_{\mathbb{M}}(S) = \Theta(1)$ and so our analysis leads to a competitive ratio of $O(\log^2 n)$ for the RM-Algorithm. Note that we do not restrict the request locations in the set R to be on a this line. The previous bound for the line metric is an $O(n^{0.59})$ competitive algorithm [7] and this result requires both the servers S as well as the requests R to be on a line.
- Suppose the subspace spanned by the server locations in S has a doubling dimension of d, then $\mu_{\mathbb{M}}(S)$ is $O(n^{1-1/d})$ and so we obtain a competitive ratio of $O(n^{1-1/d}\log^2 n)$ for the RM-Algorithm in this case. This is the first sub-linear and near-optimal competitive algorithm in the adversarial model for any metric with bounded doubling dimension.

Technical Contribution: Our analysis combines known properties of the RM-Algorithm with carefully crafted construction of certain inner and outer balls in the metric space which can be related to the competitive ratio through a variant of the well-known Vitali's covering Lemma. We present an overview of our analysis.

Using properties of the RM-Algorithm, we carefully partition requests into $O(\log n)$ groups. Within each group, we partition the requests into $O(\log n)$ outer groups. For a fixed group and outer group, we cluster the requests based on proximity and generate a set of well-separated clusters. We then replace each cluster with two concentric weighted balls; the weight of both these balls is set to the number of requests in the cluster. We refer to the ball with a smaller radius as the inner ball and the ball with a larger radius as outer ball. The radius of the inner ball and the outer ball is chosen carefully so that they relate to the online and offline matching costs respectively (see Section III-B2 for their construction and properties). In particular, we can express the cost of the online matching as the sum, over all groups, outer groups and clusters, the product of the radius and the weight of the inner balls (Lemma 7). We also show that the outer ball contains edges of the optimal matching whose cost is at least the weight of the outer ball times its radius (Lemma 8). To bound the competitive ratio, we relate the inner ball and its radius to the radius of the outer balls.

To understand how our analysis relates inner balls to outer balls, consider the special case where each cluster has exactly one request, i.e., the weight of inner and outer ball is 1. Suppose also that the set of inner balls of all the well-separated clusters are pairwise-disjoint and suppose there is a cluster whose outer ball contains all the requests. We can bound the cost of the online matching with sum of the radii of the inner balls which, due to the disjointness of inner balls, is bounded by the cost of the traveling salesman tour of the requests. By the definition of outer ball, the optimal matching is greater than or equal to the radius of any outer ball. Since there is an outer ball that covers all the requests, the cost of the optimal matching is at least the diameter of the entire request set. Therefore, in this special case, we can bound the competitive ratio by the ratio of the traveling salesman tour and the diameter of the request set. Note that this ratio is expressed with respect to the R. We use the optimal matching to map requests to servers and express this ratio as $\mu_{\mathbb{M}}(S)$. However, the radius of the inner ball and outer ball may not satisfy these special conditions, i.e., inner balls may not be pairwise disjoint and the outer balls that we construct may not enclose all the requests.

We overcome this difficulty by using a simple variant of the Vitali's covering Lemma (Section III-B1). Using this lemma, we express the competitive ratio as a weighted average of several smaller sub-problems (Lemma 10) where each sub-problem satisfies conditions similar to that of the special case.

In Section II, we will present the RM-Algorithm and some of its useful properties. In Section III, we will present the input sensitive analysis of this algorithm. We present the proofs for the useful properties of the RM-Algorithm in Section IV. In Section V, we present a lower bound on the competitive ratio of any algorithm for this problem. We conclude in Section VI.

II. Background and Algorithm Details

In this section, we will present the RM-Algorithm and some of its useful properties. We begin by introducing notations required to describe the RM-Algorithm.

Let S and R be the set of server and request locations. A *matching* $M \subseteq S \times R$ is any set of vertex-disjoint edges of the complete bipartite graph $G(S, R)$. We denote the cost of server s serving a request r by $d(s, r)$; we assume that the locations $S \cup R$ along with the cost function $d(\cdot, \cdot)$ form a metric space. For any subset $K \subseteq S \times R$, we define the sum of the cost of its edges as its *cost* and denote it by $w(K) = \sum_{(s,r)\in K} d(s, r)$. The cost of any matching M is $w(M)$. We extend this definition of cost to any path and cycle as well. In a *perfect matching* every server in S will serve exactly one request in R, and so, $|M| = n$. A *minimum-cost perfect matching* is a perfect matching with the minimum cost. We also refer to any minimum-cost perfect matching as an *optimal matching*. Throughout this paper, we fix M_{OPT} to be a fixed optimal matching of the servers to requests. For any request $r \in R$, we use $opt(r)$ to denote the match of r in M_{OPT}. For any subset $R' \subseteq R$, let $opt(R')$ denote the set of servers to which requests of R' match to in M_{OPT}. We also say that any edge (u, v) is *contained* inside a ball $\mathcal{B}$ if both its end points are contained inside $\mathcal{B}$. For the matching M_{OPT} and the ball $\mathcal{B}$, we denote the edges of M_{OPT} that are contained in $\mathcal{B}$ by $M_{\text{OPT}} \cap \mathcal{B}$.

Given a matching M^* in the complete graph $G(S, R)$, an *alternating path* (or cycle) is a simple path (resp. cycle) whose edges alternate between those in M^* and those not in M^*. We refer to any server that is not yet matched in M^* as a *free server* and denote the set of free servers by S_F. An *alternating tree* is a tree rooted at a free request in which every path is an alternating path. An *augmenting path* P is an alternating path between a free request and a free server. We can *augment* M^* by one edge along the path P if we remove the edges of $P \cap M^*$ from M^* and add the edges of $P \setminus M^*$ to M^*. The matching after augmentation is $M^* \oplus P$, where $\oplus$ is the symmetric difference operator. For a parameter $t \geq 1$, we define the *t-net-cost* of any augmenting path P to be:

$$\phi_t(P) = t \left(\sum_{(s,r)\in P\setminus M^*} d(s, r) \right) - \sum_{(s,r)\in P\cap M^*} d(s, r).$$

The parameter t is fixed at the beginning of the algorithm. In this paper, we set t to be any constant, say $t = 3$.

Using these notations, we will now describe our algorithm. Our algorithm maintains two matchings: an online matching M and an offline matching M^* both of which are initialized to $\emptyset$. After processing $i-1$ requests, the matching M and M^* will match each of the $i-1$ requests to servers in S such that the set of unmatched servers S_F is the same for both the online matching M and the offline matching

M^*. To process the i^{th} request r_i, the algorithm does the following:

1) Compute the minimum t-net-cost augmenting path P_i with respect to the offline matching M^*. Let P_i be this path that starts at r_i and ends at some free server; we denote this free server as $s_i \in S_F$.
2) Update offline matching M^* by augmenting it along P_i, i.e., $M^* \leftarrow M^* \oplus P_i$.
3) Match r_i to s_i in the online matching M, i.e., $M \leftarrow M \cup \{(s_i, r_i)\}$.

We refer to the steps taken by the RM-Algorithm to process request r_i as phase i of the algorithm. There is an $O(n^2)$-time algorithm to compute such a minimum t-net-cost path in Step 1 of any phase i. This algorithm maintains a dual weight $y(v)$ for each $v \in S \cup R$. This set of dual weights play an important role in the input-sensitive analysis presented in this paper. The offline matching M^* that is maintained by the algorithm is always a *t-feasible matching*: M^* along with the set of dual weights $y(\cdot)$ is a t-feasible matching if the following conditions hold for any $(s, r) \in S \times R$:

$$y(s) + y(r) \quad \leq \quad td(s, r), \tag{1}$$

$$y(s) + y(r) \quad = \quad d(s, r) \quad \text{for } (s, r) \in M^*. \tag{2}$$

Initially, at the start of phase 1, every request and server will have a dual weight of 0 and the empty matching M^* along with these dual weights together form a t-feasible matching. Also, we refer to an edge $(s, r) \in S \times R$ to be *eligible* if it satisfies the following conditions:

$$y(s) + y(r) \quad = \quad td(s, r), \quad \text{if } (s, r) \notin M^* \tag{3}$$

$$y(s) + y(r) \quad = \quad d(s, r) \quad \text{if } (s, r) \in M^*. \tag{4}$$

During phase i, we process the request r_i in two sub-phases. The first sub-phase (Step 1 of the RM-Algorithm) is similar to the Hungarian Search procedure where we compute the minimum t-net-cost path P_i with respect to M^* by growing an alternating consisting only of eligible edges. To grow this tree, we adjust the dual weights of every server and request until at least one more edge becomes eligible and a new vertex enters the tree. This search procedure ends when an augmenting path P_i consisting only of eligible edges is found. Let A_i (resp. B_i) be the set of requests (resp. servers) that participated in this alternating tree for request r_i. We would like to note that during this sub-phase, the dual weights of requests in A_i will only increase where as the dual weights of servers in B_i will only reduce. It can be shown that P_i is the minimum t-net-cost augmenting path with respect to the matching M^*.

The second sub-phase (Steps 2 and 3 of the RM-Algorithm) begins once the augmenting path P_i is found. We augment the matching M^* along this path. Note that, any edge (s, r) that newly entered the offline matching M^* will satisfy (3). However, in order to ensure that the matching remains t-feasible after the augmentation, this edge (s, r) must satisfy (2). We will reduce the dual weight of r, $y(r) \leftarrow y(r) - (t-1)d(s, r)$ which will guarantee that the edge (s, r) satisfies (2). Details and correctness proof for the RM-Algorithm can be found in [1]. In addition, it is also shown that the algorithm maintains the following three invariants:

(I1) M^* and dual weights $y(\cdot)$ form a t-feasible matching,
(I2) For every server $s \in S$, $y(s) \leq 0$ and if $s \in S_F$, $y(s) = 0$. For every request $r \in R$, $y(r) \geq 0$ and if r has not yet arrived, $y(r) = 0$.
(I3) At the end of the first sub-phase, i.e., Step 1 of the algorithm is executed and the augmenting path P_i is found, the dual weight of r_i, $y(r_i)$, is equal to the t-net-cost $\phi_t(P_i)$.

Note that (I3) is not explicitly presented in [1], however, it can be inferred from Lemma 4 of [1]. Throughout the rest of this paper, we will use the following notations. We will index the requests in the order of their arrival, i.e., let r_i be the ith request to arrive. For any subset of requests $R' \subseteq R$, let $\sigma(R') = \langle r'_1, r'_2 ..., r'_i, ..., r'_{|R'|}\rangle$ be the sequence of all the requests in R' sorted in the order in which they are processed by the algorithm, i.e., r'_i was processed before r'_j if $i < j$. For any request r, let $h(r)$ be the index of this request in $\sigma(R)$. While processing a request r_i from $\sigma(R)$, our algorithm will compute a minimum t-net-cost augmenting path P_i. While processing a request r, let P be the augmenting path computed by the algorithm. For notational convenience, we denote the t-net-cost $\phi_t(P)$ by $\phi_t(r)$. For the ith request processed by the algorithm, i.e., r_i, we simplify the notation further and denote the t-net-cost of P_i by $\phi_i (= \phi_t(P_i))$. We denote the free server at the other end of the P_i as s_i. Let M_i^* be the offline matching after the ith request has been processed; i.e., the matching obtained after augmenting the matching M_{i-1}^* along P_i. Note that M_0^* is an empty matching and $M_n^* = M^*$ is the final offline matching after all the n requests have been processed. Let M_i be the online matching produced by the algorithm for the first i requests. M_i consists of edges $\bigcup_{j=1}^{i}(s_j, r_j)$. Let S_i^F be the free servers with respect to matchings M and M^* after processing i requests. The following two properties, whose proof is provided in Section IV, will be useful in the analysis:

(P1) The cost of offline matching after phase i, $w(M_i^*)$, is at most $tw(M_{\text{OPT}})$,
(P2) The t-net-cost ϕ_i of the augmenting path P_i satisfies:

$$0 \leq \phi_i \leq tw(M_{\text{OPT}}).$$

In [1], is has been shown that, when $t > 1$, the cost of the online matching can be bounded by the sum of the t-net-cost of all augmenting paths generated by the algorithm.

Lemma 1 ([1]). *For $t \geq 1$ and when processing request r_i, let P_i be the augmenting path computed by the* RM-

Algorithm and ϕ_i be the t-net-cost of P_i. Then, the following holds:

$$\sum_{i=1}^{n} \phi_i \geq \frac{t-1}{2} w(M) + \frac{t+1}{2} w(M_{\text{OPT}}). \tag{5}$$

III. Analysis of the Algorithm

For a set of points W, let $\text{DIAM}(W)$ denote the cost between the farthest pair of points in W, i.e., $\text{DIAM}(W) = \max_{a,b \in W} d(a,b)$. When $\text{DIAM}(S) = 0$, it is easy to see that any matching of servers to requests, including the one produced by the RM-Algorithm, will be an optimal matching. Therefore, we work under the assumption that the $\text{DIAM}(S) > 0$.

Let $\text{TSP}(W)$ denote the smallest cost simple cycle that visits every vertex in W exactly once. Given the server locations S, we define $\mu_{\mathbb{M}}(S)$ to be

$$\mu_{\mathbb{M}}(S) = \max_{W \subseteq S, \text{DIAM}(W) > 0} \frac{\text{TSP}(W)}{\text{DIAM}(W)}.$$

We can bound the competitive ratio of the RM-algorithm by using Lemma 1 and relating the sum of the t-net-cost of the augmenting paths to the cost of the optimal matching. Recollect that we fixed M_{OPT} to be an optimal matching and for any request r_i, $opt(r_i)$ represents the server to which r_i is matched to in M_{OPT}. We divide the request set R into R' and R'' where R' contains every request r_i of R that satisfies (a) $\phi_i \leq \frac{w(M_{\text{OPT}})}{n}$ or (b) $\phi_i \leq 16td(r_i, opt(r_i))$. The requests that do not satisfy both (a) and (b) are added to R'', i.e., $R'' = R \setminus R'$.

There are at most n requests and so the total contribution of requests that satisfy (a) to the LHS of (5) is at most $w(M_{\text{OPT}})$. Any request r_i that satisfies (b) contributes at most $16t$ times $d(r_i, opt(r_i))$ to the LHS of (5). Thus the total contribution of all such requests satisfying (b) to the LHS of (5) is at most $16tw(M_{\text{OPT}})$. Every request in R' satisfies (a) or (b) and therefore, we have

$$\sum_{r_i \in R'} \phi_i \leq (16t+1) w(M_{\text{OPT}}). \tag{6}$$

We partition the requests of R'' into groups $\{\mathcal{R}_1, \ldots, \mathcal{R}_m\}$ where any request r_i is added to a group $\mathcal{R}_j$ if $\frac{2^{j-1} w(M_{\text{OPT}})}{n} \leq \phi_i < \frac{2^j w(M_{\text{OPT}})}{n}$. For any request $r_i \in R''$, by construction $\phi_i > \frac{w(M_{\text{OPT}})}{n}$ and from (P2) $\phi_i \leq tw(M_{\text{OPT}})$ and so $m = O(\log nt)$. Combining (6) with (5), we get

$$\sum_{r_i \in R'} \phi_i + \sum_{r_i \in R''} \phi_i \geq \frac{t-1}{2} w(M) + \frac{t+1}{2} w(M_{\text{OPT}}),$$

$$\sum_{j=1}^{m} \sum_{r_i \in \mathcal{R}_j} \phi_i \geq \frac{t-1}{2} w(M) - \frac{31t+1}{2} w(M_{\text{OPT}}),$$

$$\frac{2}{t-1} \left(\frac{\sum_{j=1}^{m} \sum_{r_i \in \mathcal{R}_j} \phi_i}{w(M_{\text{OPT}})} + \frac{31t+1}{2} \right) \geq \frac{w(M)}{w(M_{\text{OPT}})}.$$

For each $1 \leq j \leq m$, if we bound $\frac{\sum_{r_i \in \mathcal{R}_j} \phi_i}{w(M_{\text{OPT}})}$ by $O(\mu_{\mathbb{M}}(S) \log n)$, then along with the fact that t is a constant greater than or equal to 3 we obtain the following:

Theorem 1. *Given any metric space $\mathbb{M}$ and any initial server configuration S with $\text{DIAM}(S) > 0$, the competitive ratio of the RM-Algorithm is $O(\mu_{\mathbb{M}}(S) \log^2 n)$ where $\mu_{\mathbb{M}}(S)$ is the largest ratio of the traveling salesman tour and the diameter of a subset among all subsets of S with a positive diameter.*

A. Analysis for each Group

In the rest of this section, we will provide our analysis for some group $\mathcal{R}_j$. As noted earlier, the t-net-cost of any request r_i in group $\mathcal{R}_j$ satisfies $\phi_i > 16td(r_i, opt(r_i))$ and $\frac{2^{j-1} w(M_{\text{OPT}})}{n} \leq \phi_i < \frac{2^j w(M_{\text{OPT}})}{n}$, where $(r_i, opt(r_i))$ is an edge in the fixed optimal matching M_{OPT}. For simplicity, we will drop j from the notation. We will denote the set $\mathcal{R}_j$ by $\mathcal{R}$ and use N to denote the number of requests in $\mathcal{R}$, i.e., $|\mathcal{R}| = N$. Consider the sequence $\sigma(\mathcal{R}) = \langle r^1, \ldots, r^N \rangle$. Recollect that $\sigma(\mathcal{R})$ sorts requests of $\mathcal{R}$ based on their arrival order.

We will partition the requests in $\mathcal{R}$ into clusters. For each *cluster* $C \subseteq \mathcal{R}$, we will designate one of the requests as its *center*. We construct these clusters by processing the requests of $\sigma(\mathcal{R}) = \langle r^1, ..., r^N \rangle$ in the order in which they appear in this sequence. While processing any request r^i, suppose r^i is "close" (defined below) to the center of an already existing cluster, we assign r^i to this cluster. Otherwise, we create a new cluster that contains r^i and we will make r^i the center of this cluster. More specifically, suppose we have already partitioned requests $\langle r^1, ..., r^{i-1} \rangle$ into clusters $\langle C_1, ..., C_p \rangle$. For any cluster $C_{k'}$, let $c^{(k')}$ be its center. For the next request r^i, let C_k be the cluster such that its center $c^{(k)}$ is the closest cluster center to r^i, i.e., $k = \arg\min_{k'=1,\ldots,p} d(c^{(k')}, r^i)$.

- If $d(c^{(k)}, r^i) < \frac{2^{j-2} w(M_{\text{OPT}})}{tn}$, then we assign the request r^i to cluster C_k,
- Otherwise, we create a new cluster $C_{p+1} = \{r^i\}$ and set its center to r^i, i.e., $c^{(p+1)} \leftarrow r^i$.

Let $\tilde{k}$ be the number of clusters formed after all the requests in $\sigma(\mathcal{R})$ are processed. Suppose $c^{(k)}$ was the hth request to be processed by the RM-Algorithm. We associate a matching with cluster C_k, denoted by $M^{(k)}$ and set it to the offline matching maintained by the RM-Algorithm right before the request $c^{(k)}$ is processed by it, i.e., $M^{(k)} = M^*_{h-1}$. From the construction of the clusters and the observation that $c^{(k)}$ is the first request in the sequence $\sigma(C_k)$, we obtain (C1) and (C2):

(C1) For any cluster C_k, the cost of any request $r \in C_k$ to its center is $d(r, c^{(k)}) < \frac{2^{j-2} w(M_{\text{OPT}})}{tn}$ and for any two cluster C_k and $C_{k'}$, the cost between their centers, $d(c^{(k)}, c^{(k')}) \geq \frac{2^{j-1} w(M_{\text{OPT}})}{tn}$.

(C2) Every request in C_k is free with respect to the matching $M^{(k)}$.

Recollect that we need to construct outer balls that contain a high density of edges of the optimal matching M_{OPT}. To assist in defining outer balls, for every request $r^g \in \mathcal{R}$, we associate a path Q_g containing many edges from the optimal matching. We refer to this path as the *optimal path*. Suppose r^g is in the cluster C_k. We define the optimal path Q_g for r_g as follows: Consider the graph $\tilde{G} = \tilde{G}(S \cup R, M_{\text{OPT}} \oplus M^{(k)})$. M_{OPT} is a perfect matching and so $\tilde{G}$ will contain only alternating cycles and one augmenting path for each free request with respect to the matching $M^{(k)}$. From (C2), r^g is a free vertex with respect to $M^{(k)}$. We set Q_g to be the augmenting path corresponding to r^g in $\tilde{G}$. Let δ_g be the maximum cost between the center $c^{(k)}$ of cluster C_k, and any request or server that participates on the optimal path Q_g, i.e.,

$$\delta_g = \max_{v \in Q_g} d(c^{(k)}, v).$$

In other words, a ball with $c^{(k)}$ as its center and δ_g as its radius will contain all the edges of the path Q_g, including the edges of M_{OPT} that participate in the path Q_g. Using Lemma 2, in Lemma 3 we provide a lower bound and an upper bound on δ_g.

Lemma 2. *Let $\mathcal{R}$ be requests of group j and for a request $r^g \in \mathcal{R}$, let r^g belong to the cluster C_k with center $c^{(k)}$. Let S_F be the free servers with respect to the matching $M^{(k)}$. Among all servers in S_F, let $s \in S_F$ be the server that is closest to $c^{(k)}$. Then, $d(s, c^{(k)}) \geq \frac{2^{j-1} w(M_{\text{OPT}})}{nt}$.*

Proof: For the sake of contradiction, let us assume that $s \in S_F$ satisfies $d(s, c^{(k)}) < \frac{2^{j-1} w(M_{\text{OPT}})}{nt}$. Since $c^{(k)}$ is in group $\mathcal{R}_j$ (note, $\mathcal{R}_j = \mathcal{R}$), $\frac{2^{j-1} w(M_{\text{OPT}})}{nt} \leq \frac{\phi_t(c^{(k)})}{t}$ and so $d(s, c^{(k)}) < \frac{\phi_t(c^{(k)})}{t}$. When processing $c^{(k)}$, let $y(c^{(k)})$ and $y(s)$ be the dual weights at the end of the first sub-phase of the algorithm. From (I3), we know that $\frac{\phi_t(c^{(k)})}{t} = \frac{y(c^{(k)})}{t}$. Combining these inequalities, we get

$$y(c^{(k)}) > td(s, c^{(k)}).$$

The algorithm maintains a t-feasible matching and so, from feasibility condition (1), we have $y(c^{(k)}) + y(s) \leq td(s, c^{(k)})$ implying that the free server s has a dual weight $y(s) < 0$ contradicting invariant (I2). ■

Next, we present an upper bound and a lower bound on δ_g.

Lemma 3. *For any request $r^g \in \mathcal{R}$, let r^g belong to cluster C_k. Let Q_g be the optimal path of r^g and let let δ_g be the cost of $c^{(k)}$ to the vertex of Q_g that is furthest from it. Then,*

$$\frac{2^{j-1} w(M_{\text{OPT}})}{tn} \leq \delta_g \leq (t + \frac{3}{2}) w(M_{\text{OPT}}).$$

Proof: Let S_F be the free servers with respect to $M^{(k)}$. Since Q_g is an augmenting path with respect to $M^{(k)}$, it contains a free server $s \in S_F$. From Lemma 2, we know that the cost of $c^{(k)}$ to s is at least $\frac{2^{j-1} w(M_{\text{OPT}})}{tn}$ implying that $\delta_g \geq \frac{2^{j-1} w(M_{\text{OPT}})}{tn}$.

We know that Q_g is an augmenting path with respect to $M^{(k)}$ containing only edges of $M^{(k)}$ and M_{OPT}. From (P1), $M^{(k)} \leq tw(M_{\text{OPT}})$ and so $w(Q_g) \leq (t+1) w(M_{\text{OPT}})$. Since r_g in cluster C_k of group $\mathcal{R}_j$, from property (C1), the cost of any request r^g to its representative request $c^{(k)}$ can be bounded by

$$d(r^g, c^{(k)}) < \frac{2^{j-2} w(M_{\text{OPT}})}{tn} \leq \frac{\phi_t(r^g)}{2t} \leq \frac{w(M_{\text{OPT}})}{2}.$$

The last inequality follows from (P2). Let δ_g be the distance from the representative request $c^{(k)}$ to the furthest vertex of the path Q_g. By triangle inequality,

$$\delta_g \leq d(r^g, c^{(k)}) + w(Q_g) \leq w(M_{\text{OPT}})/2 + (t+1) w(M_{\text{OPT}}) \leq (t + \frac{3}{2}) w(M_{\text{OPT}}).$$

■

Lemma 4. *For requests of group j given by $\mathcal{R}$, let $C_k \subseteq \mathcal{R}$ be any of its clusters. Let $\mathbb{Q}_k$ be the set containing optimal paths for every request in C_k. Then, the following properties hold:*

(a) *$\mathbb{Q}_k$ is a set of vertex-disjoint augmenting paths with respect to $M^{(k)}$,*
(b) *For $r^g \in C_k$, let Q_g be the optimal path in $\mathbb{Q}_k$. The total cost of the edges in Q, given by $w(Q)$, is at least $\delta_g/2$*
(c) *For any optimal path $Q \in \mathbb{Q}_k$, $\frac{w(M_{\text{OPT}} \cap Q)}{w(Q)} \geq \frac{1}{t+1}$.*

Proof: For (a), from (C2), we know that every request of C_k is free with respect to the matching $M^{(k)}$. By construction, the graph $\tilde{G} = \tilde{G}(S \cup R, M_{\text{OPT}} \oplus M^{(k)})$ has vertex-disjoint augmenting paths with respect to $M^{(k)}$ for each of the free request of C_k. This set of optimal paths is $\mathbb{Q}_k$.

For (b), we can reach any vertex on the path Q_g from $c^{(k)}$ by first moving from $c^{(k)}$ to r^g and then following the path Q. From triangle inequality,

$$\delta_g \leq w(Q_g) + d(c^{(k)}, r^g). \tag{7}$$

From Lemma 3, $\frac{2^{j-1} w(M_{\text{OPT}})}{nt} \leq \delta_g$ and from (C1) $d(c^{(k)}, r^g) \leq \frac{2^{j-2} w(M_{\text{OPT}})}{nt} \leq \delta_g/2$. Plugging in these bounds to (7) we get $w(Q_g) \geq \delta_g/2$.

To prove (c), let Q be the optimal path for the request $r \in C_k$. By construction, Q is an augmenting path with respect to $M^{(k)}$ and contains only edges of $M^{(k)}$ and M_{OPT}. Let s be the free server at the other end of Q. For the sake of contradiction, let us assume

$$\frac{w(Q \cap M_{\text{OPT}})}{w(Q)} < \frac{1}{t+1},$$

$$(t+1)w(Q \cap M_{\text{OPT}}) < w(Q),$$
$$tw(Q \setminus M^{(k)}) < w(Q) - w(Q \cap M_{\text{OPT}}) = w(Q \cap M^{(k)}),$$
$$tw(Q \setminus M^{(k)}) - w(Q \cap M^{(k)}) < 0,$$

implying that $\phi_t(Q) < 0$. The RM-Algorithm computes the minimum t-net-cost augmenting path for the request $c^{(k)}$. Since Q is an augmenting path with respect to $M^{(k)}$, it follows that the t-net-cost of the augmenting path computed by the RM-Algorithm is negative contradicting (P2). ■

Lemma 4 establishes some key properties of the optimal paths. In particular, it provides a lower bound on the length of any optimal path (Lemma 4(b)) and also shows that at least $\frac{1}{t+1}$ of this length belongs to edges of M_{OPT}(Lemma 4(c)). These observations will help us in the construction of an outer ball.

Outer Groups: Next, we will use the optimal paths to partition the requests of group $\mathcal{R}_j$ into outer groups. For any request $r^g \in \mathcal{R}$, we assign r^g to the *outer group* l denoted by the set $\mathcal{R}^l$ if:

$$\frac{2^{l-1}w(M_{\text{OPT}})}{nt} \leq \delta_g < \frac{2^l w(M_{\text{OPT}})}{nt}. \quad (8)$$

Doing so will partition requests in $\mathcal{R}$ into several outer groups. From Lemma 3, $\delta_g \geq \frac{2^{j-1}w(M_{\text{OPT}})}{tn}$ and so the outer group with the smallest index is $\mathcal{R}^j$. Since $\delta_g \leq (t+\frac{3}{2})w(M_{\text{OPT}})$, the outer group with the largest index $\mathcal{R}^{m'}$ is such that $\frac{2^{m'-1}w(M_{\text{OPT}})}{tn} \leq (t+\frac{3}{2})w(M_{\text{OPT}}) < \frac{2^{m'}w(M_{\text{OPT}})}{tn}$ implying $m' = O(\log(nt))$. Therefore, the total number of outer groups is $O(\log(nt))$. Recollect that we had set out to show that for any group j, $\frac{\sum_{r_i \in \mathcal{R}} \phi_i}{w(M_{\text{OPT}})}$ is $O(\mu_{\mathbb{M}}(S)\log n)$. The requests in $\mathcal{R}$ are partitioned into $O(\log(nt))$ outer groups. Therefore, for any outer group $\mathcal{R}^l$, if we show that $\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})}$ is $O(\mu_{\mathbb{M}}(S))$ then we can bound

$$\frac{\sum_{r_i \in \mathcal{R}_j} \phi_i}{w(M_{\text{OPT}})} = \sum_{l=j}^{m'} \left(\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})} \right) = O(\mu_{\mathbb{M}}(S)\log n).$$

This leads to the following lemma.

Lemma 5. *For any j, let $\mathcal{R}$ be the requests in group j which are partitioned into outer groups $\{\mathcal{R}^j, \dots, \mathcal{R}^{m'}\}$. Suppose, for each outer group l, $j \leq l \leq m'$, $\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})} = O(\mu_{\mathbb{M}}(S))$, then $\frac{\sum_{r_i \in \mathcal{R}} \phi_i}{w(M_{\text{OPT}})} = O(\mu_{\mathbb{M}}(S)\log n)$.*

B. Analysis for an Outer Group

For any outer group l of group j, if we show that $\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})} = O(\mu_{\mathbb{M}}(S))$, from Lemma 5 we will immediately prove Theorem 1. So, in this section, for any outer group l of group j, we will bound $\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})}$ by $O(\mu_{\mathbb{M}}(S))$. Let N^l represent the number of requests in outer group l. So, $\sum_{l=j}^{m'} N^l = N$. To assist with the analysis for each outer group, we will introduce a weighted variant of Vitali's covering Lemma. We will then use this variant to present our input sensitive analysis for each outer group.

1) A variant of the Vitali's covering lemma: We present a variant of *Vitali's covering lemma*. For any given ball $\mathcal{B}$ with a center c and radius $\mathbf{r}$, let $3\mathcal{B}$ denote the 3-*expansion* of ball $\mathcal{B}$ which is a ball with center c and radius $3\mathbf{r}$. Given a set of balls B, Vitali's covering lemma states that it is possible to select a subset $B' \subseteq B$ such that B' is a set of mutually disjoint balls and the union of their 3-expansion contains every ball in B.

For our variant of this lemma, we consider a set $B = \{\mathcal{B}_1, \dots, \mathcal{B}_p\}$ of weighted balls, all with the same radius $\mathbf{r}$. The weight of any ball $\mathcal{B}_i \in B$ is given by $\beta_i > 0$. We present a greedy procedure that selects a set $B' = \{\mathcal{B}_{s_1}, \mathcal{B}_{s_2}, \dots, \mathcal{B}_{s_q}\}$ where $B' \subseteq B$ and for each $\mathcal{B}_{s_i}$in B', it also selects an *intersecting set* $\mathcal{IS}_{s_i} \subseteq B$ satisfying the following properties (Lemma 6):

(a) For any ball $\mathcal{B} \in B$ there is exactly one $i, 1 \leq i \leq q$ such that $\mathcal{B} \in \mathcal{IS}_{s_i}$,
(b) For any $1 \leq i < j \leq q$, $\mathcal{B}_{s_i} \cap \mathcal{B}_{s_j} = \emptyset$,
(c) For any $1 \leq i \leq q$, $\left(\bigcup_{\mathcal{B} \in \mathcal{IS}_{s_i}} \mathcal{B}\right) \subseteq 3\mathcal{B}_{s_i}$, and,
(d) For any ball $\mathcal{B}_j \in B$ let the set $\mathcal{IS}_{s_i}$ contain $\mathcal{B}_j$. Then, $\beta_{s_i} \geq \beta_j$.

Selection Procedure: Initialize $\hat{B}$ to B and $B' = \emptyset$. We select the subset of balls $B' = \{\mathcal{B}_{s_1}, \dots, \mathcal{B}_{s_q}\}$ in an iterative fashion as follows: At the start of iteration i, let $B' = \{\mathcal{B}_{s_1}, \dots, \mathcal{B}_{s_{i-1}}\}$ be the set of disjoint balls selected in the first $i-1$ iterations. Let $\mathcal{B}_{s_i}$ be the ball with the largest weight in $\hat{B}$. Consider the largest subset of balls of $\hat{B}$ that intersect with $\mathcal{B}_{s_i}$. We refer to this set as the *intersecting set* of $\mathcal{B}_{s_i}$and denote it by $\mathcal{IS}_{s_i}$. Note that $\mathcal{B}_{s_i}$ is also included in the intersecting set $\mathcal{IS}_{s_i}$. We add $\mathcal{B}_{s_i}$ to B' and remove all the balls in the intersecting set from $\hat{B}$, i.e., $\hat{B} \leftarrow \hat{B} \setminus \mathcal{IS}_{s_i}$. This procedure ends when $\hat{B} = \emptyset$, i.e., every ball in B has been covered. The next lemma will establish property (a)–(d).

Lemma 6. *Given a set of weighted balls B, the set $B' = \{\mathcal{B}_{s_1}, \mathcal{B}_{s_2}, \dots, \mathcal{B}_{s_q}\}$ and the set $\{\mathcal{IS}_{s_1}, \mathcal{IS}_{s_2}, \dots, \mathcal{IS}_{s_q}\}$ computed by the selection procedure described above will satisfy (a),(b),(c) and (d).*

Proof: Since $\hat{B} = B$ at the start of the selection procedure and $\hat{B} = \emptyset$ at the end, there is some iteration i where $\mathcal{B}$ is removed from $\hat{B}$. By construction $\mathcal{B}$ belongs to the intersecting set $\mathcal{IS}_{s_i}$ of this iteration leading to (a).

Let $1 \leq i < j \leq q$. In iteration i, the selection procedure adds $\mathcal{B}_{s_i}$ to B' and removes from $\hat{B}$ all balls that intersect with $\mathcal{B}_{s_i}$. Since $j > i$, so $\mathcal{B}_{s_j}$ is selected from $\hat{B}$ which does not contain any balls that intersect $\mathcal{B}_{s_i}$ and $\mathcal{B}_{s_i} \cap \mathcal{B}_{s_j} = \emptyset$ leading to (b).

By construction, every ball in the intersecting set $\mathcal{B} \in \mathcal{IS}_{s_i}$ intersects $\mathcal{B}_{s_i}$. Since $\mathcal{B}$ and $\mathcal{B}_{s_i}$ have the same radius $\mathbf{r}$ and since they mutually intersect, from triangle inequality we have $\mathcal{B} \subset 3\mathcal{B}_{s_i}$ leading to (c).

In iteration j, $\mathcal{B}_{s_i}$ was chosen as its weight was no less than the weight of any ball in $\hat{B}$ including $\mathcal{B}_j$. Therefore, $\beta_i \leq \beta_{s_j}$ and (d) holds. ■

2) Rest of the Analysis: Recollect that every cluster C_k is a subset of requests in group j, i.e., $C_k \subseteq \mathcal{R}$. Let C_k^l denote those requests of C_k that were assigned an outer group l and let $\beta_k^l = |C_k^l|$. Therefore, the cluster C_k is partitioned based on outer groups into $\{C_k^j, \ldots, C_k^{m'}\}$. Every request of outer group l belongs to some cluster, and so the requests of $\mathcal{R}^l$ are partitioned by the clusters $\{C_1^l, \ldots, C_{\tilde{k}}^l\}$ and $\sum_{k=1}^{\tilde{k}} \beta_k^l = N^l$. Next, we define inner and outer balls for each cluster C_k^l and then prove useful properties of these balls. Using this variant of Vitali's covering lemma, we will then bound, for any outer group l of group j, $\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})}$ by $O(\mu_{\mathbb{M}}(S))$.

Inner and Outer Ball: For each cluster C_k^l for group j and outer group l, we define a ball $\mathcal{IB}_k^l$ centered at $c^{(k)}$ and radius $\frac{2^{j-1} w(M_{\text{OPT}})}{tn}$ to be its *inner ball*. We denote the radius of the inner ball as the *inner radius* and represented it by $\mathbf{r}_{\mathcal{IB}}^j$. We define the outer ball of C_k^l, $\mathcal{OB}_k^l$, to be a ball centered at $c^{(k)}$ with a radius $\frac{2^l w(M_{\text{OPT}})}{tn}$. The radius of the outer ball, which we refer to as the *outer radius*, is denoted by $\mathbf{r}_{\mathcal{OB}}^l$. We make a few straight-forward observations about inner ball and the outer ball.

1) The inner ball $\mathcal{IB}_k^l$ and $\mathcal{OB}_k^l$ are concentric balls with the same weight of β_k^l.
2) Since l is greater than or equal to j, $\frac{2^{j-1} w(M_{\text{OPT}})}{tn} < \frac{2^l w(M_{\text{OPT}})}{tn}$ and so the inner radius is strictly smaller than the outer radius,
3) For any request $r^g \in \mathcal{R}^l$, $\delta_g \leq \frac{2^l w(M_{\text{OPT}})}{tn}$ and so if $r^g \in C_k^l$, the optimal path Q_g is contained inside its outer ball $\mathcal{OB}_k^l$.

Next, we derive useful properties of the inner ball and outer ball. The following lemma relates the inner ball radius to $\sum_{r_i \in \mathcal{R}^l} \phi_i$ and also the traveling salesman tour of a subset of servers.

Lemma 7. *For a set of requests of group j and outer group l denoted by $\mathcal{R}^l$, let for any cluster C_k^l, for $1 \leq k \leq \tilde{k}$, the inner ball $\mathcal{IB}_k^l$ have a radius $\mathbf{r}_{\mathcal{IB}}^j$ and a weight β_k^l. Then, the following properties will hold true:*

(a) $\sum_{k=1}^{\tilde{k}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j > \frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{2t}$,
(b) *Let $\tilde{R}$ be any subset of cluster centers where $|\tilde{R}| > 1$ and let $\tilde{S} = opt(\tilde{R})$ be the set of servers to which the requests in $\tilde{R}$ match in M_{OPT}, then $|\tilde{S}| \frac{\mathbf{r}_{\mathcal{IB}}^j}{4} \leq \text{TSP}(\tilde{S})$.*

Proof: For any request r_i in group j, by definition, $\phi_i < \frac{2^j w(M_{\text{OPT}})}{n}$. Dividing both the sides of the inequality by $2t$ we get

$$\frac{\phi_g}{2t} < \frac{2^{j-1} w(M_{\text{OPT}})}{tn} = \mathbf{r}_{\mathcal{IB}}^j.$$

Summing over all requests r_i belonging to any cluster C_k^l and over all clusters $1 \leq k \leq \tilde{k}$ we get,

$$\sum_{k=1}^{\tilde{k}} \sum_{r_i \in C_k^l} \frac{\phi_i}{2t} < \sum_{k=1}^{\tilde{k}} \sum_{r_i \in C_k^l} \mathbf{r}_{\mathcal{IB}}^j = \sum_{k=1}^{\tilde{k}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j \tag{9}$$

Since any request r_i belongs to exactly one cluster in $\{C_1^l, \ldots, C_{\tilde{k}}^l\}$, we immediately get (a).

From (C1), the cost between any two cluster centers $c^{(k)}, c^{(k')}$ is at least $\frac{2^{j-2} w(M_{\text{OPT}})}{nt}$, i.e.,

$$d(c^{(k)}, c^{(k')}) \geq \frac{2^{j-2} w(M_{\text{OPT}})}{nt}. \tag{10}$$

Let $s^{(k)} = opt(c^{(k)})$ and $s^{(k')} = opt(c^{(k')})$. Recollect that both $c^{(k)}$ and $c^{(k')}$ are requests of $\mathcal{R} \subset R''$ and so $\frac{2^j w(M_{\text{OPT}})}{n} > \phi_t(c^{(k)}) > 16td(c^{(k)}, s^{(k)})$ and $\frac{2^j w(M_{\text{OPT}})}{n} > \phi_t(c^{(k')}) > 16td(c^{(k')}, s^{(k')})$. From this, we have that both $d(c^{(k)}, s^{(k)})$ and $d(c^{(k')}, s^{(k')})$ are at most $\frac{2^{j-1} w(M_{\text{OPT}})}{8nt}$. From triangle inequality, we get

$$\begin{aligned} d(c^{(k)}, c^{(k')}) &\leq d(c^{(k)}, s^{(k)}) + d(s^{(k)}, s^{(k')}) + d(s^{(k')}, c^{(k')}) \\ &\leq \frac{2^{j-1} w(M_{\text{OPT}})}{4nt} + d(s^{(k)}, s^{(k')}). \end{aligned}$$

Combining (10) and (11), we get

$$d(s^{(k)}, s^{(k')}) > \frac{2^{j-1} w(M_{\text{OPT}})}{4nt} = \mathbf{r}_{\mathcal{IB}}^j / 4. \tag{11}$$

The minimum cost between any two servers in $\tilde{S}$ is at least $\mathbf{r}_{\mathcal{IB}}^j/4$. Let $T = \langle \tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{|\tilde{S}|}, \tilde{s}_1 \rangle$ be the smallest cost simple cycle to visit every vertex in $\tilde{S}$. The cost of T is

$$\text{TSP}(\tilde{S}) = d(\tilde{s}_1, \tilde{s}_2) + \ldots + d(\tilde{s}_{|\tilde{S}|}, \tilde{s}_1) > |\tilde{S}| \frac{\mathbf{r}_{\mathcal{IB}}^j}{4}.$$

as desired. ■

The following lemma will show that the total cost of the edges of M_{OPT} that are contained in any outer ball is proportional to the product of its weight and its radius.

Lemma 8. *For any cluster C_k^l, let $\mathcal{OB}_k^l$ be the outer ball with a radius $\mathbf{r}_{\mathcal{OB}}^l$ and weight β_k^l. Then $w(\mathcal{OB}_k^l \cap M_{\text{OPT}}) \geq \frac{4}{(t+1)} \beta_k^l \mathbf{r}_{\mathcal{OB}}^l$.*

Proof: Let $\mathbb{Q}_k^l$ be the set of optimal paths for requests in C_k^l. From Lemma 4(a) and the fact that $C_k^l \subseteq C_k$, it follows that $\mathbb{Q}_k^l$ is a set of vertex-disjoint augmenting paths. Each path $Q_g \in \mathbb{Q}_k^l$ is an optimal path for some request $r^g \in C_k^l$. By definition, $\mathbf{r}_{\mathcal{OB}}^l = \frac{2^l w(M_{\text{OPT}})}{nt}$ and from Lemma 4(b), we know that $w(Q_g) \geq \delta_g/2 \geq \frac{2^{l-2} w(M_{\text{OPT}})}{tn} = 4\mathbf{r}_{\mathcal{OB}}^l$. From Lemma 4(c) it follows that

$$(t+1) w(M_{\text{OPT}} \cap Q_g) \geq w(Q_g) \geq 4\mathbf{r}_{\mathcal{OB}}^l$$

or

$$w(M_{\text{OPT}} \cap Q) \geq \frac{4\mathbf{r}_{\mathcal{OB}}^l}{t+1}.$$

The optimal paths for the β_k^l requests of C_k^l are vertex-disjoint and contained inside the ball $\mathcal{OB}_k^l$. So, $|C_k^l| = |\mathbb{Q}_k^l| = \beta_k^l$ and we can bound the edges of the optimal matching M_{OPT} contained inside $\mathcal{OB}_k^l$ by

$$w(M_{\text{OPT}} \cap \mathcal{OB}_k^l) \geq \sum_{Q \in \mathbb{Q}_k^l} w(M_{\text{OPT}} \cap Q)$$
$$\geq \sum_{Q \in \mathbb{Q}_k^l} \frac{4\mathbf{r}_{\mathcal{OB}}^l}{t+1} = \frac{4\beta_k^l \mathbf{r}_{\mathcal{OB}}^l}{t+1}$$

as desired. ■

Next, by using Lemma 9 and combining properties of inner and outer balls with our variant of Vitali's covering lemma, we obtain our analysis for each outer group (Lemma 10).

Lemma 9. *For group $\mathcal{R}_j$ and any cluster C_k. Let $s^{(k)} = opt(c^{(k)})$. Then $s^{(k)} \in \mathcal{OB}_k^l$.*

Proof: $c^{(k)} \in R''$ and so, $\phi_t(c^{(k)}) \geq 16td(c^{(k)}, s^{(k)})$. Since $c^{(k)}$ in also in group $\mathcal{R}_j$, $\phi_t(c^{(k)})/2t \leq \frac{2^{j-1}w(M_{\text{OPT}})}{nt} = \mathbf{r}_{\mathcal{IB}}^j$. Combining these inequalities, we get $d(c^{(k)}, s^{(k)}) \leq \mathbf{r}_{\mathcal{IB}}^j/8$ implying that $s^{(k)} \in \mathcal{IB}_k^l$. Since the inner ball $\mathcal{IB}_k^l$ is contained inside the outer ball $\mathcal{OB}_k^l$, $s^{(k)} \in \mathcal{OB}_k^l$. ■

Lemma 10. *Let $\mathcal{R}^l$ be requests group j and outer group l. Then,*

$$\frac{\sum_{r_i \in \mathcal{R}^l} \phi_i}{w(M_{\text{OPT}})} = O(\mu_{\mathbb{M}}(S)).$$

Proof: From Lemma 7(a), $2t \sum_{k=1}^{\tilde{k}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j \geq \sum_{r_i \in \mathcal{R}^l} \phi_i$. We set t to be a constant and so it suffices to show that

$$\frac{\sum_{k=1}^{\tilde{k}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j}{w(M_{\text{OPT}})} = O(\mu_{\mathbb{M}}(S)). \tag{12}$$

Let $B = \{\mathcal{OB}_1^l, \mathcal{OB}_2^l, \ldots, \mathcal{OB}_{\tilde{k}}^l\}$ be the set of $\tilde{k}$ outer balls, one for each cluster with β_k^l as the weight of $\mathcal{OB}_k^l$. If $C_k^l = \emptyset$, i.e., there is no request in C_k that is assigned an outer group l, then the ball $\mathcal{OB}_k^l$ will have a weight of 0. So, B may contain balls with a weight of 0. We apply the selection procedure for our variant of Vitali's covering Lemma. This procedure will return a subset $B' = \{\mathcal{OB}_{s_1}, \mathcal{OB}_{s_2}, \ldots, \mathcal{OB}_{s_q}\}$ of pairwise-disjoint outer balls and for each ball $\mathcal{OB}_{s_i}$ in this set it will also return an intersecting set $\mathcal{IS}_{s_i}$. The selected outer balls of B' and their intersection sets will satisfy Lemma 6 (a)–(d).

For any outer ball $\mathcal{OB}_{s_i}$ with weight β_{s_i}, from Lemma 8, $w(M_{\text{OPT}} \cap \mathcal{OB}_{s_i}) \geq \frac{4}{(t+1)} \beta_{s_i} \mathbf{r}_{\mathcal{OB}}^l$. From Lemma 6(b), outer balls in B' are pairwise disjoint, and so we can relate the radius of the outer ball to the total cost of the optimal matching

$$w(M_{\text{OPT}}) \geq \sum_{\mathcal{OB}_{s_i} \in B'} w(M_{\text{OPT}} \cap \mathcal{OB}_{s_i}) \geq$$
$$\sum_{\mathcal{B}_{s_i} \in B'} \frac{4\beta_{s_i} \mathbf{r}_{\mathcal{OB}}^l}{(t+1)} = \sum_{\mathcal{B}_{s_i} \in B'} \kappa_{s_i}. \tag{13}$$

Here $\kappa_{s_i} = \frac{4\beta_{s_i}\mathbf{r}_{\mathcal{OB}}^l}{(t+1)}$.

There is an intersecting set $\mathcal{IS}_{s_i}$ for every selected ball $\mathcal{B}_{s_i} \in B'$. Let K_{s_i} be the index of all those clusters whose outer ball participates in $\mathcal{IS}_{s_i}$, i.e., $k \in K_{s_i}$ if and only if $\mathcal{OB}_k^l \in \mathcal{IS}_{s_i}$. Let $\theta_{s_i} = |K_{s_i}| \beta_{s_i} \mathbf{r}_{\mathcal{IB}}^j$. If $\beta_{s_i} > 0$, then we define $\gamma_{s_i} = \frac{\theta_{s_i}}{\kappa_{s_i}}$. Otherwise, if β_{s_i} is 0, then we define $\gamma_{s_i} = 0$. From Lemma 6(a), we know that every outer ball in the set B belongs to exactly one intersecting set and from Lemma 6(d), we know that $\beta_{s_i} = \max_{k \in K_{s_i}} \beta_k^l$. Therefore, we can express

$$\sum_{k=1}^{\tilde{k}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j = \sum_{\mathcal{B}_{s_i} \in B'} \sum_{k \in K_{s_i}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j \leq \sum_{\mathcal{B}_{s_i} \in B'} \theta_{s_i}. \tag{14}$$

Combining (14) and (13), we can rewrite (12) as

$$\frac{\sum_{k=1}^{\tilde{k}} \beta_k^l \mathbf{r}_{\mathcal{IB}}^j}{w(M_{\text{OPT}})} \leq \frac{\sum_{\mathcal{B}_{s_i} \in B'} \theta_{s_i}}{\sum_{\mathcal{B}_{s_i} \in B'} \kappa_{s_i}} \tag{15}$$

$$= \sum_{\mathcal{B}_{s_i} \in B'} \left(\frac{\kappa_{s_i}}{\sum_{\mathcal{B}_{s_h} \in B'} \kappa_{s_h}} \right) \gamma_{s_i}. \tag{16}$$

Note that for any ball $\mathcal{B}_{s_i}$ if $\beta_{s_i} = 0$, then both θ_{s_i} and κ_{s_i} will be 0 and so, we can ignore them in the summation of (15). Therefore, $\frac{\sum_{\mathcal{B}_{s_i} \in B'} \theta_{s_i}}{\sum_{\mathcal{B}_{s_i} \in B'} \kappa_{s_i}}$ is simply a weighted average of γ_{s_i} values. To complete the proof, we will bound γ_{s_i} by $O(\mu_{\mathbb{M}}(S))$.

We consider two cases: (i) $|K_{s_i}| = 1$ and (ii) $|K_{s_i}| > 1$. In case (i), since $|K_{s_i}| = 1$, there is only outer ball $\mathcal{OB}_{s_i}$ in the intersecting set $\mathcal{IS}_{s_i}$. So, $\kappa_{s_i} = \frac{4}{(t+1)} \beta_{s_i} \mathbf{r}_{\mathcal{OB}}^l$ and $\theta_{s_i} = \beta_{s_i} \mathbf{r}_{\mathcal{IB}}^j$ The inner radius is smaller than the outer radius and so, $\gamma_{s_i} = O(1) = O(\mu_{\mathbb{M}}(S))$.

For case (ii), fix a ball $\mathcal{OB}_{s_i} \in B'$. Let $\tilde{R}_{s_i}$ be the centers of all the outer balls of $\mathcal{IS}_{s_i}$, i.e., $\tilde{R}_{s_i} = \bigcup_{k \in K_{s_i}} c^{(k)}$. $\tilde{R}_{s_i}$ is also a set of cluster centers. Let $\tilde{S}_{s_i} = opt(\tilde{R}_{s_i})$, i.e., the set of servers to which the requests of $\tilde{R}_{s_i}$ are matched to in M_{OPT}. From Lemma 7(b), since $|K_{s_i}| = |\tilde{S}_{s_i}| \geq 2$, we get $|\tilde{S}_{s_i}| \mathbf{r}_{\mathcal{IB}}^j \leq 4\text{TSP}(\tilde{S}_{s_i})$ and $\text{DIAM}(\tilde{S}_{s_i}) > 0$

For any ball $\mathcal{OB}_k^l$ in the intersecting set $\mathcal{IS}_{s_i}$, let $s^{(k)} = opt(c^{(k)})$. From Lemma 9, $s^{(k)} \in \mathcal{OB}_k^l$. From Lemma 6(c), every ball $\mathcal{OB}_k^l$ is inside $3\mathcal{OB}_{s_i}$. So, $\tilde{S}_{s_i} \subset 3\mathcal{OB}_{s_i}$ implying $6\mathbf{r}_{\mathcal{OB}}^l \geq \text{DIAM}(\tilde{S}_{s_i})$. Therefore,

$$\frac{\theta_{s_i}}{\kappa_{s_i}} = \frac{|\tilde{S}_{s_i}| \mathbf{r}_{\mathcal{IB}}^j}{\frac{4}{(t+1)} \mathbf{r}_{\mathcal{OB}}^l} \leq \frac{O(1)\text{TSP}(\tilde{S}_{s_i})}{\text{DIAM}(\tilde{S}_{s_i})} = O(\mu_{\mathbb{M}}(S)).$$

This concludes our analysis for each outer group. Combining this with Lemma 5, we get Theorem 1.

IV. Properties of the Algorithm

Now, we prove properties (P1) and (P2).

Lemma 11. *For any augmenting path P_i computed by our algorithm, the t-net-cost of the augmenting path P_i is at least 0 and at most $tw(M_{\text{OPT}})$.*

Proof: Let M_{OPT} be an optimal matching of S and R. For the ith request r_i, let M^*_{i-1} be the offline matching maintained by the algorithm just before request r_i is processed. Consider the graph $G(S \cup R, M_{\text{OPT}} \oplus M^*_{i-1})$. Since M_{OPT} is the perfect matching, graph G contains a set of $n-i+1$ vertex disjoint augmenting paths with respect to M^*_{i-1}, each having exactly one of the $n-i+1$ remaining requests as its end vertex. Let P' be the augmenting path in G that has r_i as one of its end vertex and let the t-net-cost of P' be ϕ'. From the definition of t-net-cost,

$$\phi' = t\left(\sum_{(s,r)\in M_{\text{OPT}}\cap P'} d(s,r)\right) - \sum_{(s,r)\in M^*_{i-1}\cap P'} d(s,r),$$

or,

$$\phi' \le t \sum_{(s,r)\in M_{\text{OPT}}\cap P'} d(s,r) \le tw(M_{\text{OPT}}).$$

Note that the augmenting path P_i computed by our algorithm while processing request r_i is a minimum t-net-cost augmenting path with a t-net-cost of ϕ_i. Therefore,

$$\phi_i \le \phi' \le tw(M_{\text{OPT}})$$

as desired.

Next, we prove that $\phi_i \ge 0$. For the sake of contradiction, suppose $\phi_i < 0$. Let $y(\cdot)$ be the dual weights right before processing r_i. Using feasibility conditions (1) and (2), we can express ϕ_i as

$$\phi_i = t \sum_{(s,r)\in P_i\setminus M^*_{i-1}} d(s,r) - \sum_{(s,r)\in P_i\cap M^*_{i-1}} d(s,r) \ge$$
$$\sum_{(s,r)\in P_i\setminus M^*_{i-1}} (y(s)+y(r)) - \sum_{(s,r)\in P_i\cap M^*_{i-1}} (y(s)+y(r)).$$

The dual weight of every vertex of P_i except the first vertex r_i and the last vertex s_i cancel each other in the above inequality and we get $\phi_i \ge y(s_i) + y(r_i)$. Since $y(s_i) = 0$ and $\phi_i < 0$, we get $y(r_i) < 0$ contradicting (I2). ■

Lemma 12. *The cost of the offline matching M^* maintained by the algorithm after the end of phase i is at most $tw(M_{\text{OPT}})$.*

Proof: Consider M^*_i, which is the offline matching computed by the algorithm after all the first i requests have been processed. Let M_{OPT} be the optimal matching for the set S and R. Let $y(\cdot)$ denote the dual weights at the end of phase i. From (I1), (I2) and the fact that all requests that have not yet arrived have a dual weight of 0, we get $\sum_{v\in S\cup R} y(v) = w(M^*_i)$. For any edge (u,v) in M_{OPT}, from (1), $y(u)+y(v) \le td(u,v)$. Adding over all the edges of M_{OPT}, we get $w(M^*_i) = \sum_{v\in S\cup R} y(v) \le tw(M_{\text{OPT}})$. ■

V. Lower Bound

In this section, we will provide a sketch of the lower bound construction that bounds from below the competitive ratio of any online algorithm $\mathcal{A}$ in a metric space $\mathbb{M}$ and for server locations S is $\Omega(\mu_{\mathbb{M}}(S))$. Let S' be the subset such that $\textsc{Tsp}(S')/\textsc{Diam}(S') = \mu_{\mathbb{M}}(S)$. The adversary can remove any server s' that is not in S' by simply placing a request at the location of s'; any other allocation by the algorithm $\mathcal{A}$ will only result in a higher cost assignment. After this, we are left with servers in the set S'. Without loss of generality, we assume that all the servers of S' are in distinct locations. The adversary will generate the remaining requests as follows. Let s_1 be some server in S'. The adversary will place two requests r_1 and r_2 at the location of s_1. The first request is matched by s_1 at a cost 0 and let s_2 be the server assigned by $\mathcal{A}$ to request r_2 with a cost $d(s_1, s_2)$. Next, the adversary will place a request r_3 at location of server s_2. For the ith request, let s_{i-1} be the server assigned to r_{i-1} at a cost $d(s_{i-2}, s_{i-1})$, then the adversary will place request r_i at the location of server s_{i-1}; let s_i be the server assigned by $\mathcal{A}$ to request r_i; the cost of this assignment is $d(s_{i-1}, s_i)$. In this fashion, the total cost incurred by $\mathcal{A}$ is $\sum_{i=1}^{|S'|-1} d(s_i, s_{i+1})$ which is the cost of a path that visits every vertex in S'. Therefore, the cost incurred by $\mathcal{A}$ is at least $\textsc{Tsp}(S')/2$. For $i \ge 2$, the optimal matching will match r_i to s_{i-1} at zero cost. It will also match the request r_1 to $s_{|S'|}$ with a cost $d(r_1, s_{|S'|}) \le \textsc{Diam}(S')$. Therefore, the ratio of the cost of the online solution and the offline solution produced by $\mathcal{A}$ is at least $\textsc{Tsp}(S')/2\textsc{Diam}(S') = \mu_{\mathbb{M}}(S)/2$ leading to the following theorem.

Theorem 2. *In the adversarial model, given a set of servers S in a metric space $\mathbb{M}$, any online algorithm $\mathcal{A}$ will competitive ratio of $\Omega(\mu_{\mathbb{M}}(S))$.*

VI. Conclusion

In this paper, we present a novel input sensitive analysis of a deterministic online algorithm that was introduced in [1]. In particular, we show that for any metric space $\mathbb{M}$ and a set of server locations S, the competitive ratio of the algorithm is $O(\mu_{\mathbb{M}}(S)\log^2 n)$ where $\mu_{\mathbb{M}}(S)$ is the maximum ratio of the traveling salesman tour and the diameter of a subset of servers among all subsets of S with a positive diameter. We also show that any algorithm, optimized for the metric space $\mathbb{M}$ and the server locations S has a competitive ratio of $\Omega(\mu_{\mathbb{M}}(S))$.

It is possible to construct a 1-dimensional example for which the competitive ratio of this RM-Algorithm has a competitive ratio of $\Theta(\log n)$. Therefore, for the line metric, i.e., the server locations are points on a line, the RM-Algorithm cannot achieve a competitive ratio of $\Theta(\mu_{\mathbb{M}}(S))$. We conclude by stating the following open problem.

- Is there an online algorithm that achieves a competitive ratio of $\Theta(\mu_{\mathbb{M}}(S))$ for any metric space $\mathbb{M}$ and server configuration S?

ACKNOWLEDGMENT

This work is supported by an NSF CRII grant CCF-1464276.

REFERENCES

[1] S. Raghvendra, "A Robust and Optimal Online Algorithm for Minimum Metric Bipartite Matching," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, vol. 60, 2016, pp. 18:1–18:16.

[2] M. Mahdian and Q. Yan, "Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps," in *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, ser. STOC '11, 2011, pp. 597–606.

[3] M. S. Manasse, L. A. McGeoch, and D. D. Sleator, "Competitive algorithms for server problems," *J. Algorithms*, vol. 11, no. 2, pp. 208–230, May 1990.

[4] E. Koutsoupias and C. H. Papadimitriou, "On the k-server conjecture," *J. ACM*, vol. 42, no. 5, pp. 971–983, Sep. 1995.

[5] S. Khuller, S. G. Mitchell, and V. V. Vazirani, "On-line algorithms for weighted bipartite matching and stable marriages," *Theor. Comput. Sci.*, vol. 127, no. 2, pp. 255–267, 1994.

[6] B. Kalyanasundaram and K. Pruhs, "Online weighted matching," *J. Algorithms*, vol. 14, no. 3, pp. 478–488, 1993.

[7] A. Antoniadis, N. Barcelo, M. Nugent, K. Pruhs, and M. Scquizzato, "A $o(n)$-competitive deterministic algorithm for online matching on a line," in *Approximation and Online Algorithms: 12th International Workshop, WAOA 2014*, 2015, pp. 11–22.

[8] N. Bansal, N. Buchbinder, A. Gupta, and J. Naor, "An $O(log^2 k)$-competitive algorithm for metric bipartite matching," in *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, 2007, pp. 522–533.

[9] N. Bansal, N. Buchbinder, A. Madry, and J. Naor, "A polylogarithmic-competitive algorithm for the k-server problem," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, Oct 2011, pp. 267–276.

[10] A. Gupta and K. Lewi, "The online metric matching problem for doubling metrics," in *Automata, Languages, and Programming*, 2012, vol. 7391, pp. 424–435.

[11] E. Koutsoupias and A. Nanavati, *Approximation and Online Algorithms: First International Workshop, WAOA 2003, Budapest, Hungary, September 16-18, 2003. Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, ch. The Online Matching Problem on a Line, pp. 179–191.