Capacity-Insensitive Algorithms for Online Facility Assignment Problems on a Line

Tsubasa Harada†  Toshiya Itoh‡  Shuichi Miyazaki§

Abstract: In the online facility assignment problem OFA($k, \ell$), there exist $k$ servers with a capacity $\ell \geq 1$ on a metric space and a request arrives one-by-one. The task of an online algorithm is to irrevocably match a current request with one of the servers with vacancies before the next request arrives. As special cases for OFA($k, \ell$), we consider OFA($k, \ell$) on a line, which is denoted by OFAL($k, \ell$) and OFAL_{eq}($k, \ell$), where the latter is the case of OFAL($k, \ell$) with equidistant servers. In this paper, we perform the competitive analysis for the above problems. As a natural generalization of the greedy algorithm grdy, we introduce a class of algorithms called MPFS (Most Preferred Free Servers) and show that any MPFS algorithm has the capacity-insensitive property, i.e., for any $\ell \geq 1$, alg is $c$-competitive for OFA($k, 1$) iff alg is $c$-competitive for OFA($k, \ell$). By applying the capacity-insensitive property of the greedy algorithm grdy, we derive the matching upper and lower bounds $4k - 5$ on the competitive ratio of grdy for OFAL_{eq}($k, \ell$). To investigate the capability of MPFS algorithms, we show that the competitive ratio of any MPFS algorithm alg for OFAL_{eq}($k, \ell$) is at least $2k - 1$. Then we propose a new MPFS algorithm idas (Interior Division for Adjacent Servers) for OFAL($k, \ell$) and show that the competitive ratio of idas for OFAL_{eq}($k, \ell$) is at most $2k - 1$, i.e., idas for OFAL_{eq}($k, \ell$) is best possible in all the MPFS algorithms.

Key Words: Online algorithm, Competitive analysis, Online metric matching, Online facility assignment problem, Greedy algorithm.

1 Introduction

Online optimization (profit maximization or cost minimization) problems are real-time computation, in which a sequence of requests is an input, each request is given to an online algorithm one-by-one, and an online algorithm must decide how to deal with the current request before the next request arrives. Once the decision is fixed for the current request, the online algorithm is not allowed to change it later. In general, the efficiency of online algorithms is measured by competitive analysis which is initiated by Sleator
and Tarjan [33]. Informally, we say that an online algorithm \textsc{alg} is \textit{c-competitive} (or the \textit{competitive ratio} of \textsc{alg} is at most \(c\)) if the cost of output by \textsc{alg} is at most \(c\) times worse than the optimal cost (the formal definition will be given in Section 2.3).

The \textit{online metric matching problem} is initiated independently by Kalyanasundaram and Pruhs [17] and Khuller et al. [21] as an online variant of the minimum cost bipartite matching problem, and is formulated as follows: \(k\) servers are located on a given metric space and \(k\) requests (on the metric space) are given one-by-one in an online manner. The task of an online algorithm is to match each request immediately with one of the \(k\) servers. The cost of matching a request with a server is determined by the distance between them. The goal of the problem is to minimize the sum of the costs of matching \(k\) requests to \(k\) distinct servers. For this problem, Kalyanasundaram and Pruhs [17] and Khuller et al. [21] presented a deterministic online algorithm, which is called \textit{Permutation} [17], and showed that it is \((2k - 1)\)-competitive and best possible.

Later, Kalyanasundaram and Pruhs [19] restricted the underlying metric space to be a line and introduced a problem referred to as the online matching problem on a line. For this restricted problem, Kalyanasundaram and Pruhs [19] conjectured that (i) there exists a 9-competitive algorithm and (ii) the \textit{Work Function} algorithm [23] has a constant competitive ratio, but (i) and (ii) were disproved in [12] and [22], respectively. There have been extensive studies on this problem [3, 2, 13, 28, 30, 31] and the best upper bound on the competitive ratio [28, 31] is \(O(\log k)\), which is achieved by the \textit{ROBUST-MATCHING} algorithm [30]. While the best lower bound on the competitive ratio [12] has been 9.001 for a long time, Peserico and Scquizzato [29] drastically improved it to \(\Omega(\sqrt{\log k})\).

As a variant of the online metric matching problem, Ahmed et al. [1] formulated the \textit{online facility assignment} problem as follows: There exist \(k\) servers located equidistantly on a line and each request appears (one-by-one) on the line. Each server has a \textit{capacity}, which corresponds to the possible number of requests that can be matched to the server. Ahmed et al. [1] showed (with rough proofs) that the greedy algorithm \textsc{grdy} [18] is \(4k\)-competitive and the \textit{Optimal-fill} algorithm is \(k\)-competitive for any \(k > 2\). Itoh et al. [16] also analyzed the competitive ratio for small \(k \geq 2\), and showed that (i) for \(k = 2\), the competitive ratio of any algorithm is at least 3 and \textsc{grdy} is 3-competitive, i.e., \textsc{grdy} is best possible for \(k = 2\), and (ii) for \(k = 3, 4, \text{ and } 5\), the competitive ratio of any algorithm is at least \(1 + \sqrt{6} > 3.449\), \(\frac{4 + \sqrt{73}}{3} > 4.181\), and \(\frac{13}{3} > 4.333\), respectively. Further results on this problem were extensively obtained by Satake [32].

### 1.1 Our Contributions

In this paper, we deal with the online facility assignment problem, where each server has a capacity\(^1\), and consider the following cases: (case 1) a capacity of each server is 1; (case 2) a capacity of each server is not necessarily 1. In general, the competitive analysis for (case 2) may be harder than that for (case 1). Optimistically, we expect for an algorithm to have the \textit{capacity-insensitive} property, i.e., if it is \(c\)-competitive for (case 1), then it is also \(c\)-competitive for (case 2). This property makes the algorithm design much easier.

In Section 3, we introduce the class of MPFS (Most Preferred Free Servers) algorithms and show that any MPFS algorithm has the capacity-insensitive property (Corollary 3.1). In Section 4, we formulate the \textit{faithful} property crucial for the competitive analysis in

\(^1\) Note that if a capacity of each server is 1, then it is equivalent to the online metric matching problem.
the subsequent discussions. In Section 5, we analyze the competitive ratio of GRDY for OFAL_{eq}(k, \ell) and derive a lower bound $4k - 5$ (in Theorem 5.1) and an upper bound $4k - 5$ (in Corollary 5.2). In Section 6, we show that for any MPFS algorithm ALG for OFAL_{eq}(k, \ell), the competitive ratio of ALG is at least $2k - 1$, while in Section 7, we propose a new MPFS algorithm IDAS (Interior Division for Adjacent Servers) for OFAL(k, \ell) and show that the competitive ratio of IDAS for OFAL_{eq}(k, \ell) is at most $2k - 1$, i.e., IDAS for OFAL_{eq}(k, \ell) is best possible in all the MPFS algorithms.

1.2 Related Work

Another version of the online metric matching problem was initiated by Karp et al. [20]. Since it has application to ad auction, several variants of the problem have been extensively studied (see, e.g., [25] for a survey).

For the online metric matching problem with $k$ servers, a deterministic algorithm (called Permutation algorithm [17]) is known, which is $(2k - 1)$-competitive and best possible, but probabilistic algorithms with better competitive ratio [7, 26] are also shown.

Ahmed et al. [1] also considered the online facility assignment problem on an unweighted graph $G$, and showed the competitive ratios of GRDY and Optimal-fill algorithms are $2|E(G)|$ and $\frac{|E(G)|k}{r}$, respectively, where $r$ is the radius of $G$. Muttakee et al. [27] derived the competitive ratios of GRDY and Optimal-fill algorithms for grid graphs and the competitive ratio of the Optimal-fill algorithm for arbitrary graphs. There have been extensive studies for the online metric matching problem with delays [10], in which an online algorithm is allowed to deter a decision for the current request at the cost of waiting time as a "time cost." The goal of the problem is to minimize the sum of total matching cost and total time cost. There exist studies for deterministic algorithms [6, 8, 9, 11] and the best upper bound on the competitive ratio [6] is $O(m^\frac{\log(3/2+\varepsilon)}{2+\varepsilon}) \approx O(m^{0.59})$, where $m$ is the number of requests. There also exist studies for randomized algorithms [4, 5, 10, 24]. The best upper bound on the competitive ratio is $O(\log n)$ by Azar et al. [5] and the best lower bound for the competitive ratio is $\Omega(\frac{\log n}{\log \log n})$ by Ashlagi et al. [4].

2 Preliminaries

2.1 Online Facility Assignment Problem

Let $(X, d)$ be a metric space, where $X$ is a (possibly infinite) set of points and $d : X \times X \to \mathbb{R}$ is a distance function. We use $S = \{s_1, \ldots, s_k\}$ to denote the set of $k$ servers and use $\sigma = r_1 \cdots r_n$ to denote a request sequence. For each $1 \leq j \leq k$, a server $s_j$ is characterized by the position $p(s_j) \in X$ and $s_j$ has capacity $c_j \in \mathbb{N}$, i.e., $s_j$ can be matched with at most $c_j$ requests. We assume that $n \leq c_1 + \cdots + c_k$. For each $1 \leq i \leq n$, a request $r_i$ is also characterized by the position $p(r_i) \in X$.

The set $S$ is given to an online algorithm in advance, while requests are given one-by-one from $r_1$ to $r_n$. At any time of the execution of an algorithm, a server is called free if the number of requests matched with it is less than its capacity, and full otherwise. When a request $r_i$ is revealed, an online algorithm must match $r_i$ with one of free servers. If $r_i$ is matched with $s_j$, the pair $(r_i, s_j)$ is added to the current matching and the cost $\text{cost}(r_i, s_j) = d(p(r_i), p(s_j))$ is incurred for this pair. The cost of the matching is the sum of the costs of all the pairs contained in it. The goal of online algorithms is to
minimize the cost of the final matching. We refer to such a problem as the online facility assignment problem with \( k \) servers and denote it by \( \text{OFA}(k, \{c_j\}_{j=1}^k) \). For the case that \( c_1 = \ldots = c_k = \ell \geq 1 \), it is immediate that \( n \leq k\ell \) and we simply use \( \text{OFA}(k, \ell) \) to denote the online facility assignment problem with \( k \) servers (of uniform capacity \( \ell \)).

### 2.2 Online Facility Assignment Problem on a Line

By setting \( X = \mathbb{R} \), we can regard the online facility assignment problem with \( k \) servers as the online facility assignment problem on a line with \( k \) servers, and we denote such a problem by \( \text{OFAL}(k, \{c_j\}_{j=1}^k) \) for general capacities and \( \text{OFA}(k, \ell) \) for uniform capacities. In this case, it is immediate that \( p(s_j) \in \mathbb{R} \) for each \( 1 \leq j \leq k \) and \( p(r_i) \in \mathbb{R} \) for each \( 1 \leq i \leq n \). Without loss of generality, we assume that \( p(s_1) < \cdots < p(s_k) \) and let

\[
d_j = p(s_{j+1}) - p(s_j)
\]

for each \( 1 \leq j \leq k - 1 \). For the case that \( d_1 = \cdots = d_k = d \) with some constant \( d > 0 \), we use \( \text{OFAL}_{eq}(k, \{c_j\}_{j=1}^k) \) and \( \text{OFAL}_{eq}(k, \ell) \) to denote \( \text{OFAL}(k, \{c_j\}_{j=1}^k) \) and \( \text{OFA}(k, \ell) \) with equidistant \( k \) servers, respectively. For the subsequent discussion, we assume without loss of generality that \( d = 1 \) for both \( \text{OFAL}_{eq}(k, \{c_j\}_{j=1}^k) \) and \( \text{OFAL}_{eq}(k, \ell) \).

In the rest of the paper, we will abuse the notations \( r_i \in \mathbb{R} \) and \( s_j \in \mathbb{R} \) for \( \text{OFA}(k, \ell) \) instead of \( p(r_i) \in \mathbb{R} \) and \( p(s_j) \in \mathbb{R} \), respectively, when those are clear from the context.

### 2.3 Notations and Terminologies

For a request sequence \( \sigma \), let \( |\sigma| \) be the length of \( \sigma \), i.e., \( |\sigma| = n \) for \( \sigma = r_1 \cdots r_n \). For a request sequence \( \sigma = r_1 \cdots r_n \) and a request sequence \( \tau = \tilde{r}_1 \cdots \tilde{r}_m \), we use \( \sigma \circ \tau \) to denote the concatenation of \( \sigma \) and \( \tau \), i.e., \( \sigma \circ \tau = r_1 \cdots r_n \tilde{r}_1 \cdots \tilde{r}_m \).

For \( \text{OFA}(k, \{c_j\}_{j=1}^k) \), let \( S = \{s_1, \ldots, s_k\} \) be the set of \( k \) servers. For an (online/offline) algorithm \( DA \) for \( \text{OFA}(k, \{c_j\}_{j=1}^k) \) and a request sequence \( \sigma = r_1 \cdots r_n \), we use \( s_{da}(r_i; \sigma|S) \) to denote the server with which \( DA \) matches \( r_i \) for each \( 1 \leq i \leq n \) when \( DA \) processes \( \sigma \). Let \( DA(r_i; \sigma|S) \) be the cost incurred by \( DA \) to match \( r_i \) with \( s_{da}(r_i; \sigma|S) \), i.e., \( DA(r_i; \sigma|S) = \text{cost}(r_i, s_{da}(r_i; \sigma|S)) \). For a subsequence \( \tau = r_{i_1} \cdots r_{i_m} \) of \( \sigma \), we use \( DA(\tau; \sigma|S) \) to denote the total cost incurred by \( DA \) to match each \( r_{i_h} \) with the server \( s_{da}(r_{i_h}; \sigma|S) \), i.e.,

\[
DA(\tau; \sigma|S) = \sum_{h=1}^m DA(r_{i_h}; \sigma|S).
\]

When \( \tau = \sigma \), we simply write \( DA(\sigma|S) \) instead of \( DA(\sigma; \sigma|S) \). On defining \( s_{da}(r_i; \sigma|S) \), \( DA(r_i; \sigma|S) \), \( DA(\tau; \sigma|S) \), and \( DA(\sigma|S) \), it is crucial to indicate the set \( S \) of servers explicitly, whose role will become clear in Theorem 5.3 (especially in Claims 5.1 and 5.2). We use \( \text{OPT} \) to denote the optimal offline algorithm, i.e., \( \text{OPT} \) knows the entire sequence \( \sigma = r_1 \cdots r_n \) in advance and minimizes the total cost to match each request \( r_i \) with the server \( s_{opt}(r_i; \sigma|S) \). Let \( \text{ALG} \) be an online algorithm for \( \text{OFA}(k, \{c_j\}_{j=1}^k) \) and \( \sigma = r_1 \cdots r_n \) be a request sequence. For each \( 1 \leq i \leq n \), we define the type of a request \( r_i \) w.r.t. \( \text{ALG} \) by

\[
\text{type}_{\text{al}}(r_i) = (s_{al}(r_i; \sigma|S), s_{opt}(r_i; \sigma|S)).
\]

To evaluate the performance of an online algorithm \( \text{ALG} \), we use the (strict) competitive ratio. We say that \( \text{ALG} \) is \( c \)-competitive if \( \text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S) \) for any request sequence \( \sigma \). The competitive ratio \( R(\text{ALG}) \) of \( \text{ALG} \) is defined to be the infimum of \( c \geq 1 \) such that \( \text{ALG} \) is \( c \)-competitive, i.e.,

\[
R(\text{ALG}) = \inf \{c \geq 1 : \text{ALG} \text{ is } c \text{-competitive} \}.
\]
2.4 Technical Lemmas

As mentioned in Section 2.1, the online facility assignment problem OFA\((k, \{c_j\}_{j=1}^k)\) is defined by the set \(S = \{s_1, \ldots, s_k\}\) of \(k\) servers, where the server \(s_j\) has the capacity \(c_j\) for each \(1 \leq j \leq k\), and for any request sequence \(\sigma = r_1 \cdots r_n\) to OFA\((k, \{c_j\}_{j=1}^k)\), the condition that \(n \leq c_1 + \cdots + c_k\) must be met.

In this subsection, we show that for the design of online algorithms for OFA\((k, \{c_j\}_{j=1}^k)\), it is sufficient to deal with the case that \(n = c_1 + \cdots + c_k\) (in Lemma 2.1) and it is sufficient to deal with the case that \(c_1 = \cdots c_k = \ell\) (in Lemma 2.2).

**Lemma 2.1.** For OFA\((k, \{c_j\}_{j=1}^k)\), let \(L = c_1 + \cdots + c_k\). For any \(c \geq 1\), \(\text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S)\) for any request sequence \(\sigma\) such that \(|\sigma| = L\) iff \(\text{ALG}(\sigma'|S) \leq c \cdot \text{OPT}(\sigma'|S)\) for any request sequence \(\sigma'\) such that \(|\sigma'| \leq L\).

**Proof:** If \(\text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma'|S)\) for any request sequence \(\sigma'\) such that \(|\sigma'| \leq L\), then it is obvious that \(\text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S)\) for any request sequence \(\sigma\) such that \(|\sigma| = L\).

We show that if \(\text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S)\) for any request sequence \(\sigma\) such that \(|\sigma| = L\), then \(\text{ALG}(\sigma'|S) \leq c \cdot \text{OPT}(\sigma'|S)\) for any request sequence \(\sigma'\) such that \(|\sigma'| < L\). For a request sequence \(\sigma'\) such that \(|\sigma'| < L\), define a request sequence \(\sigma\) as follows: Append \(L - |\sigma'|\) requests at the end of \(\sigma'\) to make free servers of \(\text{OPT}\) full with zero cost. Note that \(|\sigma| = L\), and we have that \(\text{OPT}(\sigma'|S) = \text{OPT}(\sigma|S)\) and \(\text{ALG}(\sigma'|S) \leq \text{ALG}(\sigma|S)\). Thus it follows that for any request sequence \(\sigma'\) such that \(|\sigma'| < L\),

\[
\text{ALG}(\sigma'|S) \leq \text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S) = c \cdot \text{OPT}(\sigma'|S),
\]

where the 2nd inequality follows from the assumption that \(\text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S)\) for any request sequence \(\sigma\) such that \(|\sigma| = L\).

**Lemma 2.2.** For any \(\ell \geq 1\), any \(c_1, \ldots, c_k\) such that \(1 \leq c_1, \ldots, c_k \leq \ell\), and any \(c \geq 1\), there exists a \(c\)-competitive algorithm for OFA\((k, \ell)\) iff there exists a \(c\)-competitive algorithm for OFA\((k, \{c_j\}_{j=1}^k)\).

**Proof:** If there exists a \(c\)-competitive algorithm \(\text{ALG}\) for OFA\((k, \{c_j\}_{j=1}^k)\), then by setting \(c_1 = \cdots = c_k = \ell\), it is obvious that \(\text{ALG}\) is \(c\)-competitive for OFA\((k, \ell)\).

We show that if \(\text{ALG}\) is \(c\)-competitive for OFA\((k, \ell)\), then there exists a \(c\)-competitive algorithm \(\text{ALG}'\) for OFA\((k, \{c_j\}_{j=1}^k)\). For each \(1 \leq j \leq k\), let \(m_j = \ell - c_j \geq 0\) and \(\sigma'_j\) be a sequence of \(m_j\) requests on \(s_j\). Let \(\sigma' = \sigma'_1 \circ \cdots \circ \sigma'_k\). Define an online algorithm \(\text{ALG}'\) for OFA\((k, \{c_j\}_{j=1}^k)\) as follows: From Lemma 2.1, it suffices to consider a request sequence \(\sigma\) such that \(|\sigma'| = c_1 + \cdots + c_k\), and \(\text{ALG}'\) simulates \(\text{ALG}\) on \(\rho = \sigma' \circ \sigma\). Note that \(|\rho| = |\sigma' \circ \sigma| = k\ell\), and it is immediate that

\[
\text{OPT}(\sigma|S) \geq \text{OPT}(\sigma' \circ \sigma|S) = \text{OPT}(\rho|S);
\]

\[
\text{ALG}'(\sigma|S) = \text{ALG}(\sigma' \circ \sigma|S) = \text{ALG}(\rho|S).
\]

Thus it follows that for any request sequence \(\sigma\) such that \(|\sigma'| = c_1 + \cdots + c_k\),

\[
\text{ALG}'(\sigma|S) = \text{ALG}((\sigma' \circ \sigma|S) = \text{ALG}(\rho|S) \leq c \cdot \text{OPT}(\rho|S) \leq c \cdot \text{OPT}(\sigma|S),
\]

where the 1st inequality follows from the assumption that \(\text{ALG}(\rho|S) \leq c \cdot \text{OPT}(\rho|S)\) for any request sequence \(\rho\) such that \(|\rho| = k\ell\).

Based on Lemmas 2.1 and 2.2, we assume that \(c_1 = \cdots = c_k = \ell\) and we consider only request sequences \(\sigma\) such that \(|\sigma| = k\ell\) in the rest of the paper (except for Section 6).
Remark 2.1. Let $\text{DA}$ be an (online/offline) algorithm for OFA$(k, \ell)$ and $\sigma = r_1 \cdots r_n$ be a request sequence. From Lemmas 2.1 and 2.2, we assume that $|\sigma| / |S| \in \mathbb{N}$ denotes the (uniform) capacity of servers in $S$. □

3 Capacity-Insensitive Algorithms

In this section, we introduce a novel notion of “capacity-insensitive algorithms.” We first define a class of MPFS (Most Preferred Free Servers) algorithms.

Definition 3.1. Let $\text{ALG}$ be an online algorithm for OFA$(k, \ell)$. We say that $\text{ALG}$ is an MPFS (Most Preferred Free Servers) algorithm if for any request sequence $\sigma = r_1 \cdots r_n$ such that $n = k\ell$, it behaves as follows: For each $1 \leq i \leq n$,

1. the priority of servers for $r_i$ is determined by only the position of $r_i$;
2. $\text{ALG}$ matches $r_i$ with a server with the highest priority among free servers.

Let $\text{MPFS}$ be the class of MPFS algorithms. In the subsequent discussion, we show that for any $\text{ALG} \in \text{MPFS}$ and any $\ell \geq 1$, $\text{ALG}$ is $c$-competitive for OFA$(k, 1)$ iff $\text{ALG}$ is $c$-competitive for OFA$(k, \ell)$. We begin by introducing several ingredients to analyze the properties of algorithms in $\text{MPFS}$.

Definition 3.2. For a request sequence $\sigma$, we say that a set $\{\sigma_i\}_{i=1}^m$ of request sequences is a partition of $\sigma$ if it satisfies the following conditions:

1. For each $1 \leq i \leq m$, $\sigma_i$ is a subsequence of $\sigma$;
2. For each $1 \leq i < j \leq m$, $\sigma_i$ and $\sigma_j$ have no common request in $\sigma$;
3. $|\sigma_1| + \cdots + |\sigma_m| = |\sigma|$.

Example 3.1. Let $\sigma = r_1 r_2 r_3 r_4 r_5 r_6$. For $\sigma_1 = r_1 r_2 r_3 r_6$ and $\sigma_2 = r_4 r_5$, $\{\sigma_1, \sigma_2\}$ satisfies the conditions (1), (2), and (3) of Definition 3.2. Thus $\{\sigma_1, \sigma_2\}$ is a partition of $\sigma$.

For $\tilde{\sigma}_1 = r_5 r_6$, $\tilde{\sigma}_2 = r_2 r_4$, and $\tilde{\sigma}_3 = r_1 r_3$, $\{\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3\}$ satisfies the conditions (1), (2), and (3) of Definition 3.2. Thus $\{\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3\}$ is a partition of $\sigma$.

For $\tilde{\sigma}_1 = r_2 r_3$ and $\tilde{\sigma}_2 = r_1 r_2 r_4 r_6$, $\{\tilde{\sigma}_1, \tilde{\sigma}_2\}$ does not satisfy the condition (2) of Definition 3.2. Thus $\{\tilde{\sigma}_1, \tilde{\sigma}_2\}$ is not a partition of $\sigma$. □

Definition 3.3. Let $\text{ALG}$ be an online algorithm for OFA$(k, \ell)$ and $\{\sigma_i\}_{i=1}^\ell$ be a partition of a request sequence $\sigma$ such that $|\sigma| = k\ell$, where $\sigma_i = r_1^i \cdots r_k^i$ for each $1 \leq i \leq \ell$. We say that the partition $\{\sigma_i\}_{i=1}^\ell$ of $\sigma$ is coprime w.r.t. $\text{ALG}$ if

$$s_{\text{alg}}(r_s^i; \sigma|S) \neq s_{\text{alg}}(r_t^i; \sigma|S) \land s_{\text{opt}}(r_s^i; \sigma|S) \neq s_{\text{opt}}(r_t^i; \sigma|S)$$

for each $1 \leq i \leq \ell$ and any pair $1 \leq s, t \leq k$ such that $s \neq t$.

For a bipartite graph $G = (X \cup Y; E)$, we say that $M \subseteq E$ is a matching if no vertex is incident to more than one edge in $M$. For $|X| = |Y|$, we say that a matching $M$ between $X$ and $Y$ is perfect if every vertex in $X$ is incident to an edge in $M$. The following theorem plays a crucial role to analyze the properties of algorithms in $\text{MPFS}$.

Theorem 3.1 ([14, Corollary 1.57]). For any bipartite graph $G = (X \cup Y; E)$, if $G$ is $\ell$-regular, then $G$ contains a perfect matching.
From Theorem 3.1, we have the following lemma.

**Lemma 3.1.** Let \( \text{alg} \) be an online algorithm for \( \text{OFA}(k, \ell) \) and \( \sigma \) be a request sequence such that \( |\sigma| = k\ell \). Then there exists a coprime partition \( \{\sigma_i\}_{i=1}^\ell \) of \( \sigma \) w.r.t. \( \text{alg} \).

**Proof:** Fix a request sequence \( \sigma \) with \( |\sigma| = k\ell \) arbitrarily and define a bipartite graph \( G = (X \cup Y; E) \) as follows: Let \( X = Y = \{s_1, \ldots, s_k\} \), and \( (x, y) \in E \) if there exists a request \( r \) in \( \sigma \) such that type\(_{\text{alg}}\)(\( r \)) = \( (x, y) \). Since \( G \) is \( \ell \)-regular by construction, we have that \( G \) contains a perfect matching by Theorem 3.1.

We show the lemma by induction on \( \ell \geq 1 \). For \( \ell = 1 \), the lemma obviously holds. For any \( \ell \geq 2 \), we assume that the lemma holds for \( \ell - 1 \) and we show that the lemma holds for \( \ell \). For the bipartite graph \( G = (X \cup Y; E) \) such that \( |X| = |Y| \), let \( M \subseteq E \) be a perfect matching of \( G \). Note that \( M \) can be represented by a permutation \( \pi \) on \( \{s_1, \ldots, s_k\} \), i.e., \( M = \{(s_i, s_{\pi(i)})\}_{i=1}^k \). From the definition of \( G \), it follows that for each \( 1 \leq i \leq k \), there exists a request \( r^\ell_i \) such that type\(_{\text{alg}}\)(\( r^\ell_i \)) = \( (s_i, s_{\pi(i)}) \). Let \( \sigma_\ell = r^\ell_1 \cdots r^\ell_k \) and define the request sequence \( \sigma' \) by deleting \( \sigma_\ell \) from \( \sigma \). Then \( \sigma' \) can be regarded as a request sequence for \( \text{OFA}(k, \ell - 1) \) and from the induction hypothesis, it follows that there exists a coprime partition \( \{\sigma'_i\}_{j=1}^{\ell - 1} \) of \( \sigma' \) w.r.t. \( \text{alg} \). Thus \( \{\sigma'_i\}_{j=1}^{\ell - 1} \cup \{\sigma_\ell\} \) is a coprime partition of \( \sigma \) w.r.t. \( \text{alg} \), and this completes the proof of the lemma. \( \blacksquare \)

Informally, an algorithm \( \text{alg} \) for \( \text{OFA}(k, \ell) \) is **separable** if there exists a coprime partition \( \{\sigma_i\}_{i=1}^\ell \) of \( \sigma \) with \( |\sigma| = k\ell \) such that the way of matching servers for \( \sigma_i \) by \( \text{alg} \) on \( \sigma \) is completely the same as the way of matching servers for \( \sigma_i \) by \( \text{alg} \) on \( \sigma_i \) for \( \text{OFA}(k, 1) \).

**Definition 3.4.** Let \( \text{alg} \) be an online algorithm for \( \text{OFA}(k, \ell) \). For any request sequence \( \sigma \) such that \( |\sigma| = k\ell \), we say that \( \text{alg} \) is **separable** if there exists a coprime partition \( \{\sigma_i\}_{i=1}^\ell \) of \( \sigma \) w.r.t. \( \text{alg} \) such that for each \( 1 \leq i \leq \ell \) and each \( 1 \leq j \leq k \),

\[
\begin{align*}
\text{alg}(r^\ell_j; |\sigma_i|S) &= \text{alg}(r^\ell_j; |\sigma_i|S), \quad (3.1) \\
\text{opt}(r^\ell_j; |\sigma_i|S) &= \text{opt}(r^\ell_j; |\sigma_i|S), \quad (3.2)
\end{align*}
\]

where \( \sigma_i = r^\ell_1 \cdots r^\ell_k \).

Note that \( |\sigma|/|S| = \ell \) and \( |\sigma_i|/|S| = 1 \) in (3.1) and (3.2). Then from Remark 2.1, it is immediate that on the left hand side of (3.1) and (3.2), \( \text{alg} \) and \( \text{opt} \) can be regarded as an online algorithm and an offline algorithm for \( \text{OFA}(k, \ell) \), respectively, and on the right hand side of (3.1) and (3.2), \( \text{alg} \) and \( \text{opt} \) can be regarded as an online algorithm and an offline algorithm for \( \text{OFA}(k, 1) \), respectively. The following lemma plays a crucial role to discuss the properties of algorithms in \( \mathcal{MFS} \).

**Lemma 3.2.** Let \( \text{alg} \) be a separable online algorithm for \( \text{OFA}(k, \ell) \). For any \( \ell \geq 1 \), if \( \text{alg} \) is \( c \)-competitive for \( \text{OFA}(k, 1) \), then \( \text{alg} \) is \( c \)-competitive for \( \text{OFA}(k, \ell) \).

**Proof:** Fix a request sequence \( \sigma \) such that \( |\sigma| = k\ell \) arbitrarily. Then from the assumption that \( \text{alg} \) is separable for \( \text{OFA}(k, \ell) \), there exists a coprime partition \( \{\sigma_i\}_{i=1}^\ell \) of \( \sigma \) w.r.t. \( \text{alg} \).
that satisfies (3.1) and (3.2), where \( \sigma_i = r_1^i \cdots r_k^i \) for each \( 1 \leq i \leq \ell \). Then

\[
\text{ALG}(\sigma|S) = \sum_{i=1}^{\ell} \sum_{j=1}^{k} \text{ALG}(r_j^i; \sigma|S) = \sum_{i=1}^{\ell} \sum_{j=1}^{k} \text{ALG}(r_j^i; \sigma_i|S) = \sum_{i=1}^{\ell} \text{ALG}(\sigma_i|S);
\]

\[
\text{OPT}(\sigma|S) = \sum_{i=1}^{\ell} \sum_{j=1}^{k} \text{OPT}(r_j^i; \sigma|S) = \sum_{i=1}^{\ell} \sum_{j=1}^{k} \text{OPT}(r_j^i; \sigma_i|S) = \sum_{i=1}^{\ell} \text{OPT}(\sigma_i|S).
\]

Since ALG is \( c \)-competitive for \( \text{OFA}(k, 1) \), we have that

\[
\text{ALG}(\sigma|S) = \sum_{i=1}^{\ell} \text{ALG}(\sigma_i|S) \leq \sum_{i=1}^{\ell} c \cdot \text{OPT}(\sigma_i|S) = c \cdot \sum_{i=1}^{\ell} \text{OPT}(\sigma_i|S) = c \cdot \text{OPT}(\sigma|S),
\]

and this implies that ALG is \( c \)-competitive for \( \text{OFA}(k, \ell) \) for any \( \ell \geq 1 \).

The following theorem is one of the main results that captures the crucial property of algorithms in \( \text{MPFS} \) and plays an important role in the subsequent discussions.

**Theorem 3.2.** If \( \text{ALG} \) for \( \text{OFA}(k, \ell) \) is in \( \text{MPFS} \), then \( \text{ALG} \) is separable.

**Proof:** Fix an arbitrary algorithm \( \text{ALG} \in \text{MPFS} \) for \( \text{OFA}(k, \ell) \) and a request sequence \( \sigma \) such that \( |\sigma| = k\ell \). From Lemma 3.1, it follows that there exists a coprime partition \( \{\sigma_i\}_{i=1}^{k} \) of \( \sigma \) w.r.t. ALG, where \( \sigma_i = r_1^i \cdots r_k^i \) is a subsequence of \( \sigma \) for each \( 1 \leq i \leq \ell \). To complete the proof of the theorem, it suffices to show the following two facts:

1. there exists an optimal offline algorithm \( \text{OPT} \) such that \( s_{\text{opt}}(r_j^i; \sigma|S) = s_{\text{opt}}(r_j^i; \sigma_i|S) \) for each \( 1 \leq i \leq \ell \) and each \( 1 \leq j \leq k \);
2. \( s_{\text{alg}}(r_j^i; \sigma|S) = s_{\text{alg}}(r_j^i; \sigma_i|S) \) for each \( 1 \leq i \leq \ell \) and each \( 1 \leq j \leq k \).

For the fact (1), it is obvious that \( s_{\text{opt}}(\sigma_i|S) = \text{OPT}(\sigma_i; \sigma_i|S) \leq \text{OPT}(\sigma_i; \sigma|S) \) for each \( 1 \leq i \leq \ell \). Assume that there exists an \( 1 \leq h \leq \ell \) such that \( \text{OPT}(\sigma_h|S) = \text{OPT}(\sigma_h; \sigma_h|S) < \text{OPT}(\sigma_h; \sigma|S) \). Define a subsequence \( \sigma - \sigma_h \) of \( \sigma \) by deleting \( \sigma_h \) from \( \sigma \). Then

\[
\text{OPT}(\sigma|S) = \text{OPT}(\sigma_h; \sigma|S) + \text{OPT}(\sigma - \sigma_h; \sigma|S) > \text{OPT}(\sigma_h; \sigma_h|S) + \text{OPT}(\sigma - \sigma_h; \sigma|S),
\]

and this contradicts the optimality of \( \text{OPT} \). Thus we have that \( \text{OPT}(\sigma_i|S) = \text{OPT}(\sigma_i; \sigma|S) \) for each \( 1 \leq i \leq \ell \), which is achieved in such a way that \( \text{OPT} \) for \( \text{OFA}(k, \ell) \) matches \( r_j^i \) with \( s_{\text{opt}}(r_j^i; \sigma_i|S) \), i.e., \( s_{\text{opt}}(r_j^i; \sigma|S) = s_{\text{opt}}(r_j^i; \sigma_i|S) \) for each \( 1 \leq i \leq \ell \) and each \( 1 \leq j \leq k \).

We turn to show the fact (2). For simplicity, let \( s_{\text{alg}}(r_j^i; \sigma|S) = s_j^i \) for each \( 1 \leq i \leq \ell \) and \( 1 \leq j \leq k \). From the definition of coprime partition, it follows that \( \{s_1^i, \ldots, s_k^i\} = S \) for each \( 1 \leq i \leq \ell \). After ALG matches \( r_j^i \) with \( s_j^i \), ALG matches \( r_{j+1}^i, \ldots, r_k^i \) with \( s_{j+1}^i, \ldots, s_k^i \), respectively, and this implies that \( s_j^i, \ldots, s_k^i \) are free just before ALG matches \( r_j^i \) with \( s_j^i \). Since \( \text{ALG} \in \text{MPFS} \), we have that \( s_j^i \) has the highest priority for \( r_j^i \) among \( s_j^i, \ldots, s_k^i \). As mentioned in Remark 2.1, we regard ALG as an algorithm for \( \text{OFA}(k, 1) \) for the request sequence \( \sigma_i \). When processing \( r_j^i \), it is immediate that \( s_1^i, \ldots, s_{j-1}^i \) are full and \( s_j^i, \ldots, s_k^i \) are free. Thus from the fact that \( s_j^i \) has the highest priority for \( r_j^i \) among free servers \( s_j^i, \ldots, s_k^i \), it follows that ALG for \( \text{OFA}(k, 1) \) matches \( r_j^i \) with \( s_j^i \).

Then we have the following immediate corollary to Theorem 3.2.
Corollary 3.1. Let Alg ∈ MPFS. For any c ≥ 1 and any ℓ ≥ 1, Alg is c-competitive for OFA(k, 1) iff Alg is c-competitive for OFA(k, ℓ).

Proof: It is obvious that Alg for OFA(k, ℓ) is c-competitive for any ℓ ≥ 1, then Alg for OFA(k, 1) is c-competitive. The converse follows from Lemma 3.2 and Theorem 3.2.

4 Faithful Algorithms

In this section, we introduce a notion of faithful algorithms, which will play a crucial role to analyze upper bounds on the competitive ratio of algorithms in MPFS. Before discussing the faithful algorithms, we introduce tours for a set of fixed points on a line and we also observe the related notions and properties.

4.1 Tours and Their Properties

Let $V = \{v_1, \ldots, v_n\}$ be a set of distinct $n$ points (on a line), i.e., $v_1, \ldots, v_n \in \mathbb{R}$. We say that $T : v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ is a tour on $V$ and define the length of $T$ by

$$\ell(T) = |v_n - v_1| + \sum_{i=1}^{n-1} |v_i - v_{i+1}|.$$ 

For each $2 \leq i \leq n$, we identify $v_1 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ with $v_i \rightarrow \cdots \rightarrow v_n \rightarrow v_1 \rightarrow \cdots \rightarrow v_i$.

Definition 4.1. Let $T : v_1 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ be a tour on $V = \{v_1, \ldots, v_n\}$. We say that a pair $(v_i, v_j)$ is conflicting in $T$ if $v_i \leq v_{j+1} < v_{i+1} \leq v_j$.

Definition 4.2. Let $T : v_1 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ be a tour on $V = \{v_1, \ldots, v_n\}$. We say that $v_i$ is a relay point in $T$ if $v_{i-1} < v_i < v_{i+1}$ or $v_{i-1} > v_i > v_{i+1}$, where $v_0 = v_n$ for $i = 1$ and $v_{n+1} = v_1$ for $i = n$, and say that $v_i$ is a turning point in $T$ otherwise.

For a tour $T$ on $V$, we use $\text{cf}(T)$ to denote the set of all conflicting pairs in $T$ and use $\text{tp}(T)$ to denote the set of all turning points in $T$.

Remark 4.1. $|\text{tp}(T)|$ is even for any tour $T$ on $V = \{v_1, \ldots, v_n\}$. □

Definition 4.3. Let $T : v_1 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ be a tour on $V = \{v_1, \ldots, v_n\}$. We say that $\tilde{T} : t_1 \rightarrow \cdots \rightarrow t_{2m} \rightarrow t_1$ is a contracted tour of $T$ if $\tilde{T}$ consists of all the turning points in $T$ by skipping all the relay points in $T$.

For the contracted tour $\tilde{T}$ of $T$, it is immediate that $|\text{tp}(T)| = |\text{tp}(\tilde{T})|$. Note that conflicting pairs in $\tilde{T}$ can be defined in a way similar to the conflicting pairs in $T$. Let $\text{cf}(\tilde{T})$ be the set of all conflicting pairs in $\tilde{T}$.

Remark 4.2. For a tour $T : v_1 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$ on $V$, let $\tilde{T} : t_1 \rightarrow \cdots \rightarrow t_{2m} \rightarrow t_1$ be a contracted tour of $T$. Then for a conflicting pair $(t_i, t_j) \in \text{cf}(\tilde{T})$, $i$ is even iff $j$ odd. □

For a tour $T : v_1 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$, let $\tilde{T} : t_1 \rightarrow \cdots \rightarrow t_{2m} \rightarrow t_1$ be the contracted tour of $T$. For each $1 \leq p \leq 2m$, let $T^p : t_p = v_1^p \rightarrow \cdots \rightarrow v_{2m}^p = t_{p+1}$ be the path from $t_p$ to $t_{p+1}$ in $T$, where $t_{2m+1} = t_1$, and relay($t_p$) = $\{v_2^p, \ldots, v_{2m}^p\}$ be the set of relay points on $T^p$. 

Remark 4.3. \(\text{relay}(t_p) \cap \text{relay}(t_q) = \emptyset\) for each \(1 \leq p < q \leq 2m\). \(\square\)

Lemma 4.1. For a tour \(T\), let \(\hat{T}\) be the contracted tour of \(T\). Then there exists an injection \(f_{\text{inj}} : \text{cf}(\hat{T}) \to \text{cf}(T)\).

Proof: For a conflicting pair \((t_i, t_j) \in \text{cf}(\hat{T})\), it is immediate that \(t_i \leq t_{j+1} < t_{i+1} \leq t_j\) by definition. Let \(T^i : t_i = v^i_1 \to \cdots \to v^i_x = t_{i+1}\) be the path from \(t_i\) to \(t_{i+1}\) in \(T\) and \(T^j : t_j = v^j_1 \to \cdots \to v^j_y = t_{j+1}\) be the path from \(t_j\) to \(t_{j+1}\) in \(T\). Determine the maximum \(1 \leq \alpha < x\) such that \(v^i_\alpha \leq t_{j+1}\) and the maximum \(1 \leq \beta < y\) such that \(v^i_\alpha \leq v^j_\beta\). Then

\[v^i_\alpha \leq v^j_\beta \leq v^j_{\alpha+1} \leq v^i_\beta,\]

and this implies that \((v^i_\alpha, v^j_\beta) \in \text{cf}(T)\). Let \(f_{\text{inj}} : (t_i, t_j) \mapsto (v^i_\alpha, v^j_\beta)\). From Remark 4.3. it follows that \(f_{\text{inj}}(t_i, t_j) \neq f_{\text{inj}}(t_p, t_q)\) for \((t_i, t_j) \neq (t_p, t_q)\). \(\blacksquare\)

Definition 4.4. For the contracted tour \(\hat{T} : t_1 \to \cdots \to t_{2m} \to t_1\) of a tour \(T\) with \(m \geq 2\), we say that a path \(D_i : t_i \to t_{i+1} \to t_{i+2} \to t_{i+3}\) is a detour in \(\hat{T}\) if

\[t_i \leq t_{i+2} < t_{i+1} \leq t_{i+3} \quad \text{or} \quad t_i \geq t_{i+2} > t_{i+1} \geq t_{i+3},\]

where \(i + j = i + j - 2m\) if \(i + j > 2m\) for each \(1 \leq j \leq 3\).

The following guarantees that a contracted tour \(\hat{T}\) of any tour \(T\) has a detour in \(\hat{T}\).

Lemma 4.2. For a tour \(T\), let \(\hat{T} : t_1 \to \cdots \to t_{2m} \to t_1\) be the contracted tour of \(T\). If \(m \geq 2\), then there exists a detour in \(\hat{T}\).

Proof: Without loss of generality, we assume that \(t_1 = \min\{t_1, \ldots, t_{2m}\}\) by the definition of contracted tours. Since \(t_1, \ldots, t_{2m}\) are turning points in \(\hat{T}\), we have that \(t_1 < t_3 < t_2\) and \(t_3 < t_4\). If \(t_2 \leq t_4\), then the path \(t_1 \to t_2 \to t_3 \to t_4\) is a detour \(D_1\) in \(\hat{T}\) and the lemma holds. If \(t_2 > t_4\), then \(t_1 < t_3 < t_4 < t_2\). By continuing this process, assume that we reach to the setting that

\[t_1 < \cdots < t_{2i-1} < t_{2i} < \cdots < t_2.\]

From the definition of turning points in \(\hat{T}\), it follows that \(t_{2i+1} < t_{2i}\) and \(t_{2i+1} < t_{2i+2}\). If \(t_{2i+1} \leq t_{2i-1}\), then \(t_{2i-2} \to t_{2i-1} \to t_{2i} \to t_{2i+1}\) is a detour \(D_{2i-2}\) in \(\hat{T}\) and the lemma holds. If \(t_{2i+1} > t_{2i-1}\), then it is immediate that

\[t_1 < \cdots < t_{2i-1} < t_{2i+1} < t_{2i} < \cdots < t_2.\]

If \(t_{2i} \leq t_{2i+2}\), then \(t_{2i-1} \to t_{2i} \to t_{2i+1} \to t_{2i+2}\) is a detour \(D_{2i-1}\) in \(\hat{T}\) and the lemma holds. If \(t_{2i} > t_{2i+2}\), then we reach to the setting that

\[t_1 < \cdots < t_{2i-1} < t_{2i+1} < t_{2i+2} < t_{2i} < \cdots < t_2.\]

As a result, we reach to the final setting that \(t_1 < \cdots < t_{2m-1} < t_{2m} < \cdots < t_{2}\), however, \(t_{2m-2} \to t_{2m-1} \to t_{2m} \to t_1\) is a detour \(D_{2m-2}\) in \(\hat{T}\) and the lemma holds. Thus there always exists a detour in \(\hat{T}\), and this completes the proof of the lemma. \(\blacksquare\)

We have the following result on conflicting pairs and turning points in a tour \(T\).
**Lemma 4.3.** Let $T : v_1 \to \cdots \to v_n \to v_1$ be a tour on $V = \{v_1, \ldots, v_n\}$. Then

$$|\text{cf}(T)| \geq \frac{|\text{tp}(T)|}{2}.$$  

**Proof:** Let $\tilde{T} : t_1 \to \cdots \to t_{2m} \to t_1$ be the contracted tour of $T$. Since $|\text{cf}(T)| \geq |\text{cf}(\tilde{T})|$ by Lemma 4.1, it suffices to show that $|\text{cf}(\tilde{T})| \geq m$. We show this by induction on $m \geq 1$. For the case that $m = 1$, it follows that $(t_1, t_2)$ is a conflicting pair in $\tilde{T}$.

Assume that the lemma holds for $m \geq 1$, i.e., for any contracted tour $\tilde{T}$ with $2m$ turning points, there exist at least $m$ conflicting pairs in $\tilde{T}$. Let $\tilde{T}' : \tau_1 \to \cdots \to \tau_{2(m+1)} \to \tau_1$ be the contracted tour of a tour $\tilde{T}'$. From Lemma 4.2, there exists a detour $\tau_i \to \tau_{i+1} \to \tau_{i+2} \to \tau_{i+3}$ in $\tilde{T}'$. For the contracted tour $\tilde{T}'$, define $\tilde{T}^{*}$ by replacing the detour $\tau_i \to \tau_{i+1} \to \tau_{i+2} \to \tau_{i+3}$ in $\tilde{T}'$ with the arrow $\tau_i \to \tau_{i+3}$, i.e., $\tilde{T}^*: \tau_1 \to \cdots \to \tau_i \to \tau_{i+3} \to \cdots \to \tau_{2(m+1)} \to \tau_1$. Note that $\tilde{T}^*$ consists of $2m$ turning points. Then by the induction hypothesis, there exist at least $m$ conflicting pairs in $\tilde{T}^*$. Since $\tilde{T}^*$ loses a conflicting pair $(\tau_i, \tau_{i+1})$ in $\tilde{T}'$, we have that there exist at least $m + 1$ conflicting pairs in $\tilde{T}'$. $lacksquare$

### 4.2 Faithful Algorithms and Opposite Request Sequences

In this subsection, we introduce a notion of faithful algorithms and a notion of opposite sequences, which will make the competitive analysis easier.

**Definition 4.5.** Let $\text{alg}$ be an online/offline algorithm for OFAL($k, \ell$) and $\sigma = r_1 \cdots r_{k\ell}$ and $\tau = q_1 \cdots q_{k\ell}$ be request sequences. We say that $\tau$ is closer to $S$ than $\sigma$ w.r.t. alg if

1. for each $1 \leq i \leq k\ell$, $q_i$ is not farther than $r_i$ to $s_{\text{alg}}(r_i; \sigma | S)$ with which alg matches $r_i$, i.e., $r_i \geq q_i \geq s_{\text{alg}}(r_i; \sigma | S)$ or $r_i \leq q_i \leq s_{\text{alg}}(r_i; \sigma | S)$.

2. there exists $1 \leq i \leq k\ell$ such that $q_i$ is closer than $r_i$ to $s_{\text{alg}}(r_i; \sigma | S)$ with which alg matches $r_i$, i.e., $r_i > q_i \geq s_{\text{alg}}(r_i; \sigma | S)$ or $r_i < q_i \leq s_{\text{alg}}(r_i; \sigma | S)$.

**Definition 4.6.** Let $\text{alg}$ be an online/offline algorithm for OFAL($k, \ell$). We say that alg is faithful if for any request sequence $\sigma = r_1 \cdots r_{k\ell}$ and any request sequence $\tau = q_1 \cdots q_{k\ell}$ that is closer to $S$ than $\sigma$ w.r.t. alg, $s_{\text{alg}}(r_i; \sigma | S) = s_{\text{alg}}(q_i; \tau | S)$ for each $1 \leq i \leq k\ell$.

Then we have the following lemma on $\text{opt}$ for OFAL($k, \ell$).

**Lemma 4.4.** opt is faithful for OFAL($k, \ell$).

**Proof:** We consider opt for OFAL($k, \ell$). For a request sequence $\sigma = r_1 \cdots r_{k\ell}$, let $r_i$ be a request such that $|r_i - s_{\text{opt}}(r_i; \sigma | S)| > 0$. For the request $r_i$, let $r'_i$ be a request such that $r_i > r'_i \geq s_{\text{opt}}(r_i; \sigma | S)$ or $r_i < r'_i \leq s_{\text{opt}}(r_i; \sigma | S)$. Define a request sequence $\tau = q_1 \cdots q_{k\ell}$ by replacing $r_i$ with $r'_i$ in $\sigma$, i.e., for each $1 \leq h \leq k\ell$,

$$q_h = \begin{cases} r_h & h \neq i; \\ r'_h & h = i \end{cases}$$

Note that the request sequence $\tau$ is closer to $S$ than $\sigma$ w.r.t. opt. Let

$$s_{\text{opt}}(\sigma | S) = (s_{\text{opt}}(r_1; \sigma | S), \ldots, s_{\text{opt}}(r_{k\ell}; \sigma | S));$$

$$s_{\text{opt}}(\tau | S) = (s_{\text{opt}}(q_1; \tau | S), \ldots, s_{\text{opt}}(q_{k\ell}; \tau | S)),$$
Lemma 4.5. Let \( \text{alg} \) be an online algorithm for \( \text{OFAL}(k, \ell) \) and \( \sigma = r_1 \cdots r_{k\ell} \) be a request sequence. We say that \( \sigma \) is opposite w.r.t. \( \text{alg} \) if for each \( 1 \leq i \leq k\ell \),

\[
\frac{|r_h - s_{\text{opt}}(r_h; |S|)|}{\sum_{h=1}^{k\ell} |r_h - s_{\text{opt}}(r_h; |S|)| + |q_i - r_i|} \geq \frac{|q_h - s_{\text{opt}}(q_h; |S|)| + |q_i - r_i|}{\sum_{h=1}^{k\ell} |q_h - s_{\text{opt}}(q_h; |S|)| + \sum_{h=1}^{k\ell} |r_h - s_{\text{opt}}(r_h; |S|)|}.
\]

This implies that \( \sum_{h=1}^{k\ell} |r_h - s_{\text{opt}}(r_h; |S|)| = \sum_{h=1}^{k\ell} |r_h - s_{\text{opt}}(q_h; |S|)| \). Then it follows that \( s_{\text{opt}}(\tau; |S|) \) is a common optimal matching for \( \sigma \) and \( \tau \). By iterating this process, we can conclude that \( \text{OPT} \) is faithful for \( \text{OFAL}(k, \ell) \).

To analyze the competitive ratio for faithful algorithms, the following notion is useful.

Definition 4.7. Let \( \text{alg} \) be an online algorithm for \( \text{OFAL}(k, \ell) \) and \( \sigma = r_1 \cdots r_{k\ell} \) be a request sequence. We say that \( \sigma \) is opposite w.r.t. \( \text{alg} \) if for each \( 1 \leq i \leq k\ell \),

\[
r_i \in [s_{\text{alg}}(r_i; |S|), s_{\text{opt}}(r_i; |S|)] \lor r_i \in [s_{\text{opt}}(r_i; |S|), s_{\text{alg}}(r_i; |S|)].
\]

The following lemma holds for request sequences w.r.t. a faithful \( \text{alg} \) for \( \text{OFAL}(k, \ell) \).

Lemma 4.5. Let \( \text{alg} \) be a faithful online algorithm for \( \text{OFAL}(k, \ell) \). Then for any request sequence \( \sigma \), there exists an opposite \( \tau \) w.r.t. \( \text{alg} \) such that \( \text{rate}(\sigma) \leq \text{rate}(\tau) \), where

\[
\text{rate}(\sigma) = \begin{cases} \frac{\text{alg}(\sigma|S|)}{\text{opt}(\sigma|S|)} & \text{if } \text{opt}(\sigma|S|) > 0; \\ \infty & \text{if } \text{opt}(\sigma|S|) = 0, \text{alg}(\sigma|S|) > 0; \\ 1 & \text{if } \text{opt}(\sigma|S|) = \text{alg}(\sigma|S|) = 0. \end{cases}
\]

Proof: If \( \sigma \) is opposite w.r.t. \( \text{alg} \), then it suffices to set \( \tau = \sigma \). Then we assume that \( \sigma = r_1 \cdots r_{k\ell} \) is not opposite w.r.t. \( \text{alg} \). In this case, there exists \( 1 \leq i \leq k\ell \) such that

1. \( r_i < \min\{s_{\text{alg}}(r_i; |S|), s_{\text{opt}}(r_i; |S|)\} \);
2. \( r_i > \max\{s_{\text{alg}}(r_i; |S|), s_{\text{opt}}(r_i; |S|)\} \).

For the case (1), let \( s \in S \) be the server closest to \( r_i \) among \( s_{\text{alg}}(r_i; |S|) \) and \( s_{\text{opt}}(r_i; |S|) \), i.e., \( s = \min\{s_{\text{alg}}(r_i; |S|), s_{\text{opt}}(r_i; |S|)\} \). Let \( r'_i \) be a request that is located on \( s \) and we define a request sequence \( \tau = q_1 \cdots q_{k\ell} \) as follows: for each \( 1 \leq h \leq k\ell \),

\[
q_h = \begin{cases} r_h & h \neq i; \\ r'_i & h = i. \end{cases}
\]

Note that \( \tau \) is closer to \( S \) than \( \sigma \) w.r.t. \( \text{alg} \) and \( \text{opt} \). Since \( \text{opt} \) is faithful for \( \text{OFAL}(k, \ell) \) by Lemma 4.4 and \( \text{alg} \) is faithful for \( \text{OFAL}(k, \ell) \), we have that \( s_{\text{alg}}(r_h; |S|) = s_{\text{alg}}(q_h; |S|) \) and \( s_{\text{opt}}(r_h; |S|) = s_{\text{opt}}(q_h; |S|) \) for each \( 1 \leq h \leq k\ell \). Thus it follows that

\[
\text{rate}(\sigma) = \frac{\text{alg}(\sigma|S|)}{\text{opt}(\sigma|S|)} = \frac{\text{alg}(\tau|S|) + |r_i - s|}{\text{opt}(\tau|S|) + |r_i - s|} \leq \frac{\text{alg}(\tau|S|)}{\text{opt}(\tau|S|)} = \text{rate}(\tau).
\]

For the case (2), the argument similar to that of the case (1) holds. Iterate this process until \( \tau \) gets opposite w.r.t. \( \text{alg} \), and this completes the proof of the lemma.
4.3 Faithful MPFS Algorithms

In this subsection, we introduce crucial notions of a characteristic permutation, a single tour, and multiple tours. These notions provide a general framework for the analysis of the competitive ratio for faithful algorithms in MPFS.

Definition 4.8. Let \( S = \{s_1, \ldots, s_k\} \) be the set of \( k \) servers on a line. For an online algorithm \( \text{ALG} \) and a request sequence \( \sigma = r_1 \cdots r_k \), we say that a bijection \( \pi^\text{alg}_\sigma : S \to S \) is a characteristic permutation for \( \sigma \) w.r.t. \( \text{ALG} \) if \( \pi^\text{alg}_\sigma : s_{\text{opt}(r_i; \sigma|S)} \mapsto s_{\text{alg}(r_i; \sigma|S)} \) for each \( 1 \leq i \leq k \). We say that a request sequence \( \sigma \) has a single tour \( T^\text{alg}_\sigma \) on \( S \) w.r.t. \( \text{ALG} \) if \( \pi^\text{alg}_\sigma \) is cyclic on \( S \), and \( \sigma \) has multiple tours \( \{T^\text{alg,i}_\sigma\}_{i=1} \) if \( \pi^\text{alg}_\sigma \) is not cyclic on \( S \).

For a request sequence \( \sigma \), assume that \( \sigma \) has a single tour \( T^\text{alg}_\sigma : s_i \to \cdots \to s_k \to s_i \) on \( S \) w.r.t. \( \text{ALG} \). Then we define the length of \( T^\text{alg}_\sigma \) by

\[
\ell(T^\text{alg}_\sigma) = |s_{i_k} - s_i| + \sum_{j=1}^{k-1} |s_{i_{j+1}} - s_{i_j}|.
\]

For an opposite request sequence \( \sigma \) w.r.t. faithful \( \text{ALG} \), the following properties hold:

Property 4.1. Let \( \sigma = r_1 \cdots r_k \) be an opposite request sequence w.r.t. faithful \( \text{ALG} \) and assume that \( \sigma \) has a single tour \( T^\text{alg}_\sigma : s_i \to \cdots \to s_k \to s_i \) on \( S \). Then

(1) for each \( 1 \leq j \leq k \), there exists a request \( r \) (in \( \sigma \)) that is located between \( s_{i_j} \) and \( s_{i_{j+1}} \), where we regard \( s_{i_{k+1}} \) as \( s_i \), i.e.,

- (a) if \( s_{i_j} < s_{i_{j+1}} \), then \( r \in [s_{i_j}, s_{i_{j+1}}] \)
- (b) if \( s_{i_{j+1}} < s_{i_j} \), then \( r \in [s_{i_{j+1}}, s_{i_j}] \)

and has type \( \text{alg}(r) = \langle s_{i_{j+1}}, s_{i_j} \rangle \).

(2) \( \text{ALG}(\sigma|S) + \text{OPT}(\sigma|S) = \ell(T^\text{alg}_\sigma) \).

To derive an upper bound on the competitive ratio of faithful \( \text{ALG} \) for \( \text{OFAL}(k, \ell) \), we deal with the case that a request sequence \( \sigma \) has a single tour in Theorems 5.2 and 7.2, while we deal with the case that \( \sigma \) has multiple tours in Theorems 5.3 and 7.3.

To show that any faithful algorithm is \( c \)-competitive, the following lemma is crucial, especially for the proofs of Theorems 5.2 and 7.2.

Lemma 4.6. Let \( \text{ALG} \) be faithful for \( \text{OFAL}(k, \ell) \) and assume that an opposite request sequence \( \sigma \) has a single tour \( T^\text{alg}_\sigma \) on \( S \) w.r.t. \( \text{ALG} \). If there exists a function \( H(T^\text{alg}_\sigma) \in \mathbb{R} \) such that \( \text{OPT}(\sigma|S) \geq \frac{H(T^\text{alg}_\sigma)}{c+1} \) and \( \ell(T^\text{alg}_\sigma) \leq H(T^\text{alg}_\sigma) \), then \( \text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S) \).

Proof: From Property 4.1(2), we have that \( \text{ALG}(\sigma|S) + \text{OPT}(\sigma|S) = \ell(T^\text{alg}_\sigma) \). Thus

\[
\text{ALG}(\sigma|S) + \text{OPT}(\sigma|S) = \ell(T^\text{alg}_\sigma) \leq H(T^\text{alg}_\sigma) \leq (c+1) \cdot \text{OPT}(\sigma|S),
\]

and this implies that \( \text{ALG}(\sigma|S) \leq c \cdot \text{OPT}(\sigma|S) \). 

In the remainder of the paper, we will simply use \( T_\sigma \), \( \{T^i_\sigma\}_{i=1} \), and \( \pi_\sigma \) instead of \( T^\text{alg}_\sigma \), \( \{T^\text{alg,i}_\sigma\}_{i=1} \), and \( \pi^\text{alg}_\sigma \), respectively, when \( \text{ALG} \) is clear from the context.
5 Competitive Ratio of Greedy Algorithm

In this section, we define one of the most natural algorithms for OFA($k, \ell$) that is referred to as a greedy algorithm [18], and discuss the basic properties of the greedy algorithm.

Before introducing the greedy algorithm for OFA($k, \ell$), we begin with presenting a notion of consuming pairs of a request sequence $\sigma$.

**Definition 5.1.** Let $\text{ALG}$ be a faithful algorithm for OFA($k, \ell$) and $\sigma$ be an opposite request sequence w.r.t. $\text{ALG}$ with a single tour $T_\sigma : s_{h_1} \rightarrow \cdots \rightarrow s_{h_k} \rightarrow s_{h_1}$ on $S$. We say that a pair $(r_i, r_j)$ of requests is consuming in $\sigma$ if $(s_{h_p}, s_{h_q})$ is conflicting in $T_\sigma$, where $s_{h_p} = s_{\text{opt}}(r_i; \sigma|S)$ and $s_{h_q} = s_{\text{opt}}(r_j; \sigma|S)$. Let $c_{\text{alg}}(\sigma)$ be the set of all consuming pairs in $\sigma$, i.e., $c_{\text{alg}}(\sigma) = \{(r_i, r_j) : (r_i, r_j) \text{ is consuming in } \sigma\}$, and $f_{bij} : c(T_\sigma) \rightarrow c_{\text{alg}}(\sigma)$ be a bijection that maps $(s_{h_p}, s_{h_q})$ to $(r_i, r_j)$ as above.

**Remark 5.1.** From Definition 5.1, it is immediate that $|c(T_\sigma)| = |c_{\text{alg}}(\sigma)|$.

Informally, we say that an algorithm for OFA($k, \ell$) is greedy if the current request is matched with the nearest free server. More formally, we have the following definition.

**Definition 5.2.** Let $\text{ALG}$ be an online algorithm for OFA($k, \ell$) and $\sigma = r_1 \cdots r_n$ be a request sequence such that $n = kl$. We say that $\text{ALG}$ is a greedy algorithm (denoted by GRDY), if $\text{ALG}$ matches a request $r_i$ with the nearest\(^2\) free server $s \in S$ for each $1 \leq i \leq n$.

Kalyanasundaram and Pruhs [18] showed that GRDY is $(2^k - 1)$-competitive for OFA($k, 1$) and Kalyanasundaram and Pruhs [19] mentioned that GRDY is also $(2^k - 1)$-competitive for OFA($k, \ell$) without proof. Since GRDY $\in \text{MPFS}$ (see Definitions 3.1 and 5.2), Corollary 3.1 immediately provides a formal proof of the following result:

**Corollary 5.1.** For any $\ell \geq 1$, GRDY is $(2^k - 1)$-competitive for OFA($k, \ell$).

The following lemma is essential for the subsequent discussions on GRDY.

**Lemma 5.1.** GRDY is faithful for OFA($k, \ell$).

**Proof:** Consider GRDY for OFA($k, \ell$). Let $\sigma = r_1 \cdots r_{kl}$ and $\tau = q_1 \cdots q_{kl}$ be request sequences, where $\tau$ is closer to $S$ than $\sigma$ w.r.t. GRDY. For each $1 \leq i \leq kl$, GRDY matches a request $r_i$ with the server $s_{\text{grdy}}(r_i; \sigma|S)$. Since $s_{\text{grdy}}(r_i; \sigma|S)$ is the closest free server to both $r_i$ and $q_i$, for each $1 \leq i \leq kl$, we have that $s_{\text{grdy}}(r_i; \sigma|S) = s_{\text{grdy}}(q_i; \tau|S)$ for each $1 \leq i \leq kl$, i.e., GRDY is faithful for OFA($k, \ell$). \(\square\)

Ahmed et al. [1] showed that GRDY is $4k$-competitive for OFA$_{eq}(k, \ell)$ with an informal proof. In this section, we show that $R(\text{GRDY}) = 4k - 5$ for OFA$_{eq}(k, \ell)$. In fact, we show that $R(\text{GRDY}) \geq 4k - 5$ in Theorem 5.1 and $R(\text{GRDY}) \leq 4k - 5$ in Corollary 5.2, which generalizes the result by Itoh et al. [16], i.e., $R(\text{GRDY}) = 3 = 4 \cdot 2 - 5$ for OFA$_{eq}(2, \ell)$.

**Remark 5.2.** Since ROBUST-MATCHING [30] matches a request $r_i$ with a server depending on the positions of requests $r_1, \ldots, r_i$ observed so far, we have that ROBUST-MATCHING $\notin \text{MPFS}$. Thus, although ROBUST-MATCHING for OFA($k, 1$) is $O(\log k)$-competitive [30, 31], this cannot be applied to OFA($k, \ell$) for $\ell > 1$. \(\square\)

\(^2\) If there exist at least two nearest free servers for the request $r_i$, then GRDY chooses the one with the largest index as the matching server for $r_i$. 

14
In the following subsections, we analyze the competitive ratio of GRDY for OFAL(k, ℓ). More precisely, we derive a lower bound on the competitive ratio of GRDY for OFAL\(_{eq}(k, ℓ)\) in Section 5.1 and upper bounds on the competitive ratio of GRDY for OFAL(k, ℓ) and OFAL\(_{eq}(k, ℓ)\) in Section 5.2.

### 5.1 A Lower Bound for the Competitive Ratio

In this subsection, we construct an adversarial request sequence \(σ\) to derive a lower bound on the competitive ratio of GRDY for OFAL\(_{eq}(k, ℓ)\) with \(k \geq 2\).

**Theorem 5.1.** For OFAL\(_{eq}(k, ℓ)\) with \(k \geq 2\), \(R(\text{GRDY}) \geq 4k - 5\).

**Proof:** For simplicity, assume that \(s_j = j - 1\) for each \(1 \leq j \leq k\). We construct a request sequence \(σ = σ_1 \cdots σ_ℓ\) such that GRDY(\(σ|S\)) = \((4k - 5) \cdot \text{OPT}(σ|S)\), where \(σ_i = r_i^1 \cdots r_i^k\).

For each \(1 \leq i \leq ℓ\), let \(r_i^1 = \frac{1}{2}\) and \(r_i^j = s_j = j - 1\) for each \(2 \leq j \leq k\). By Definition 5.2,

\[
\text{GRDY}(σ|S) = \left\{\frac{1}{2} + (k - 2) + (k - 1)\right\} \cdot ℓ = \frac{4k - 5}{2} \cdot ℓ;
\]

\[
\text{OPT}(σ|S) = \frac{1}{2} \cdot ℓ.
\]

Thus for the request sequence \(σ\) defined above, it follows that

\[
\frac{\text{GRDY}(σ|S)}{\text{OPT}(σ|S)} = \frac{4k - 5}{\frac{1}{2} \cdot ℓ} = 4k - 5
\]

and this implies that \(R(\text{GRDY}) \geq 4k - 5\). 

### 5.2 An Upper Bound for the Competitive Ratio

In this subsection, we investigate the properties of GRDY and derive an upper bound on the competitive ratio of GRDY for OFAL(k, ℓ), which leads to the matching upper bound for OFAL\(_{eq}(k, ℓ)\). From Corollary 3.1 and the fact that GRDY \(∈ \text{MPFS}\) for OFAL(k, ℓ), it suffices to analyze the competitive ratio of GRDY for OFAL(k, 1). In this subsection, we consider only request sequences \(σ\) of length \(k\). For \(k = 1\), let \(U(S) = 1\) and for \(k \geq 2\), let

\[
U(S) = \frac{s_k - s_1}{d_{\text{min}}} = \frac{\sum_{j=1}^{k-1} d_j}{\min_{1 \leq j \leq k-1} d_j}.
\]

As shown in Lemma 5.1, we already know that GRDY is faithful for OFAL(k, ℓ). Then from Lemma 4.5, it suffices to consider opposite request sequences w.r.t. GRDY to derive an upper bound on the competitive ratio of GRDY. In the remainder of this subsection, we assume that a request sequence \(σ\) is opposite w.r.t. GRDY.

#### 5.2.1 Single Tour for GRDY

We first consider the case that a request sequence \(σ\) has a single tour w.r.t. GRDY (and in Section 5.2.2, we also consider the case that \(σ\) has multiple tours w.r.t. GRDY).

**Lemma 5.2.** \(\text{OPT}(σ|S) \geq \frac{d_{\text{min}}}{2} \cdot |c_{\text{GRDY}}(σ)|\) for any opposite request sequence \(σ\).
**Proof:** Fix an opposite request sequence $\sigma$ w.r.t. GRDY arbitrarily and we partition $c_{\text{grdy}}^+(\sigma)$ into $c_{\text{grdy}}^+(\sigma)$ and $c_{\text{grdy}}^-(\sigma)$ as follows:

$$c_{\text{grdy}}^+(\sigma) = \{(r_i; r_j) : (r_i; r_j) \in c_{\text{grdy}}^+(\sigma) \land i < j\};$$

$$c_{\text{grdy}}^-(\sigma) = \{(r_i; r_j) : (r_i; r_j) \in c_{\text{grdy}}^-(\sigma) \land i > j\}.$$

For each request $a$ in $\sigma$, let $c_{\text{grdy}}^+(a; \sigma)$ be the set of consuming pairs (in $\sigma$) of the form $(a, \ast) \in c_{\text{grdy}}^+(\sigma)$, i.e., $c_{\text{grdy}}^+(a; \sigma) = \{(a, r) : (a, r) \in c_{\text{grdy}}^+(\sigma)\}$, and enumerate nonempty $c_{\text{grdy}}^+(a; \sigma)$'s by $c_{\text{grdy}}^+(a_1; \sigma), \ldots, c_{\text{grdy}}^+(a_\mu; \sigma)$. Note that $c_{\text{grdy}}^+(a_1; \sigma), \ldots, c_{\text{grdy}}^+(a_\mu; \sigma)$ is a partition of $c_{\text{grdy}}^+(\sigma)$. For each request $b$ in $\sigma$, let $c_{\text{grdy}}^-(b; \sigma)$ be the set of consuming pairs (in $\sigma$) of the form $(\ast, b) \in c_{\text{grdy}}^-(\sigma)$, i.e., $c_{\text{grdy}}^-(b; \sigma) = \{(r, b) : (r, b) \in c_{\text{grdy}}^-(\sigma)\}$, and in a way similar to the definition of $c_{\text{grdy}}^+(a; \sigma)$'s, we use $c_{\text{grdy}}^-(b_1; \sigma), \ldots, c_{\text{grdy}}^-(b_\nu; \sigma)$ to denote a partition of $c_{\text{grdy}}^-(\sigma)$. It is immediate that

$$|c_{\text{grdy}}^+(\sigma)| = |c_{\text{grdy}}^+(\sigma)| + |c_{\text{grdy}}^-(\sigma)| = \sum_{i=1}^\mu |c_{\text{grdy}}^+(a_i; \sigma)| + \sum_{j=1}^\nu |c_{\text{grdy}}^-(b_j; \sigma)|.$$

Note that $\{a_1, \ldots, a_\mu\} \cap \{b_1, \ldots, b_\nu\} = \emptyset$. If $\text{OPT}(a_i; \sigma|S) \geq \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^+(a_i; \sigma)|$ for each $1 \leq i \leq \mu$ and $\text{OPT}(b_j; \sigma|S) \geq \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^-(b_j; \sigma)|$ for each $1 \leq j \leq \nu$, then it follows that

$$\text{OPT}(\sigma|S) \geq \sum_{i=1}^\mu \text{OPT}(a_i; \sigma|S) + \sum_{j=1}^\nu \text{OPT}(b_j; \sigma|S)$$

$$\geq \sum_{i=1}^\mu \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^+(a_i; \sigma)| + \sum_{j=1}^\nu \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^-(b_j; \sigma)| = \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^+(\sigma)|.$$

Thus it suffices to show that (1) $\text{OPT}(a_i; \sigma|S) \geq \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^+(a_i; \sigma)|$ for each $1 \leq i \leq \mu$ and (2) $\text{OPT}(b_j; \sigma|S) \geq \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^-(b_j; \sigma)|$ for each $1 \leq j \leq \nu$.

For the case (1), fix $1 \leq i \leq \mu$ arbitrarily and consider $\text{OPT}(a_i; \sigma|S)$. We assume that $c_{\text{grdy}}^+(a_i; \sigma) = \{(a_i, r_{i_1}), \ldots, (a_i, r_{i_u})\}$, where $i_1, \ldots, i_u$ are ordered in such a way that

$$s_{\text{opt}}(a_i; \sigma|S) \leq s_{\text{grdy}}(r_{i_1}; \sigma|S) < \cdots < s_{\text{grdy}}(r_{i_u}; \sigma|S) < s_{\text{grdy}}(a_i; \sigma|S).$$

Since $a_i$ is the earliest request among $a_i, r_{i_1}, \ldots, r_{i_u}$ by the definition of $c_{\text{grdy}}^+(\sigma)$, we have that $s_{\text{grdy}}(a_i; \sigma|S), s_{\text{grdy}}(r_{i_1}; \sigma|S), \ldots, s_{\text{grdy}}(r_{i_u}; \sigma|S)$ are free just before GRDY matches $a_i$ with $s_{\text{grdy}}(a_i; \sigma|S)$. This implies that $a_i \geq s_{\text{grdy}}(a_i; \sigma|S) + s_{\text{grdy}}(r_{i_1}; \sigma|S) + \cdots + s_{\text{grdy}}(r_{i_u}; \sigma|S)$.

$$\text{OPT}(a_i; \sigma|S) = |a_i - s_{\text{opt}}(a_i; \sigma|S)| = a_i - s_{\text{opt}}(a_i; \sigma|S) \geq a_i - s_{\text{grdy}}(r_{i_1}; \sigma|S)$$

$$= a_i - s_{\text{grdy}}(r_{i_1}; \sigma|S) + s_{\text{grdy}}(r_{i_1}; \sigma|S) - s_{\text{grdy}}(r_{i_1}; \sigma|S)$$

$$\geq \frac{s_{\text{grdy}}(a_i; \sigma|S) + s_{\text{grdy}}(r_{i_1}; \sigma|S)}{2} - s_{\text{grdy}}(r_{i_1}; \sigma|S) + s_{\text{grdy}}(r_{i_1}; \sigma|S) - s_{\text{grdy}}(r_{i_1}; \sigma|S)$$

$$= \frac{s_{\text{grdy}}(a_i; \sigma|S) - s_{\text{grdy}}(r_{i_1}; \sigma|S)}{2} + s_{\text{grdy}}(r_{i_1}; \sigma|S) - s_{\text{grdy}}(r_{i_1}; \sigma|S)$$

$$\geq \frac{d_{\text{min}}}{2} + \sum_{j=1}^{u-1} \{s_{\text{grdy}}(r_{i_j}; \sigma|S) - s_{\text{grdy}}(r_{i_{j+1}}; \sigma|S)\}$$

$$\geq \frac{d_{\text{min}}}{2} + \sum_{j=1}^{u-1} d_{\text{min}} = d_{\text{min}} \cdot \frac{d_{\text{min}}}{2} = \frac{d_{\text{min}}}{2} \cdot |c_{\text{grdy}}^+(a_i; \sigma)|,$$
where the 1st inequality is due to the assumption that $s_{\text{grdy}}(r_{i_u}; \sigma|S) \geq s_{\text{opt}}(a_i; \sigma|S)$.

For the case (2), fix $1 \leq j \leq \nu$ arbitrarily and consider $\text{OPT}(b_j; \sigma|S)$. We assume that $cs_{\text{grdy}}^{-}(b_j; \sigma) = \{(r_{j_1}, b_j), \ldots, (r_{j_u}, b_j)\}$, where $j_1, \ldots, j_u$ are ordered in such a way that

$$s_{\text{grdy}}(b_i; \sigma|S) < s_{\text{grdy}}(r_{j_1}; \sigma|S) < \cdots < s_{\text{grdy}}(r_{j_u}; \sigma|S) \leq s_{\text{opt}}(b_j; \sigma|S).$$

In a way similar to the case (1), we can show that $\text{OPT}(b_j; \sigma|S) \geq \frac{d_{\text{min}}}{2} \cdot |cs_{\text{grdy}}^{-}(b_j; \sigma)|$ for each $1 \leq j \leq \nu$, and this complete the proof the lemma.

By applying Lemma 4.6 to $GRDY$ for $\text{OFAL}(k, \ell)$, we can show the following theorem.

**Theorem 5.2.** For a request sequence $\sigma$, if $\sigma$ has a single tour w.r.t. $GRDY$ on $S$, then $GRDY(\sigma|S) \leq (4 \cdot U(S) - 1) \cdot \text{OPT}(\sigma|S)$.

**Proof:** For $k = 1$, it is immediate that $GRDY(\sigma|S) = \text{OPT}(\sigma|S) \leq (4U(S) - 1) \cdot \text{OPT}(\sigma|S)$ with $U(S) = 1$. For any $k \geq 2$, let $T_\sigma : s_{h_1} \rightarrow \cdots \rightarrow s_{h_k} \rightarrow s_{h_1}$ be a single tour on $S$ and $H(T_\sigma) = |\text{tp}(T_\sigma)| \cdot (s_k - s_1)$. Then it follows that

$$\text{OPT}(\sigma|S) \geq \frac{d_{\text{min}}}{2} \cdot |cs_{\text{grdy}}(\sigma)| = \frac{d_{\text{min}}}{2} \cdot |\text{cf}(T_\sigma)| \geq \frac{d_{\text{min}}}{4} \cdot |\text{tp}(T_\sigma)| = \frac{|\text{tp}(T_\sigma)| \cdot (s_k - s_1)}{4 \cdot U(S)} = H(T_\sigma),$$

where the 1st inequality follows from Lemma 5.2, the 1st equality follows from Remark 5.1, and the 2nd inequality follows from Lemma 4.3. It is easy to see that $\ell(T_\sigma) \leq H(T_\sigma)$. Thus from Lemma 4.6, we have that $GRDY(\sigma|S) \leq (4 \cdot U(S) - 1) \cdot \text{OPT}(\sigma|S)$.

### 5.2.2 Multiple Tours for $GRDY$

In general, all the opposite request sequences do not necessarily have a single tour w.r.t. $GRDY$. For the case that an opposite request sequence $\sigma$ has multiple tours $\{T^i_{\sigma}\}_{i=1}^t$ w.r.t. $GRDY$, we regard each $T^i_{\sigma}$ as a single tour for a subsequence $\sigma_i$ of $\sigma$ and derive an upper bound of the competitive ratio by combining each of them for the request sequence $\sigma$.

**Theorem 5.3.** $R(\text{GRDY}) \leq 4 \cdot U(S) - 1$ for $\text{OFAL}(k, \ell)$.

**Proof:** In Theorem 5.2, we already showed that $GRDY(\sigma|S) \leq (4 \cdot U(S) - 1) \cdot \text{OPT}(\sigma|S)$ for any request sequence $\sigma = r_1 \cdots r_k$ with a single tour $T_\sigma$ on $S$, i.e., a bijection $\pi_\sigma : s_{\text{opt}}(r_i; \sigma|S) \mapsto s_{\text{grdy}}(r_i; \sigma|S)$ is cyclic on $S$. In the remainder of the proof, we show that $GRDY(\sigma|S) \leq (4 \cdot U(S) - 1) \cdot \text{OPT}(\sigma|S)$ for any request sequence $\sigma$ with multiple tours $T^1_\sigma, \ldots, T^t_\sigma$ on $S$, i.e., the bijection $\pi_\sigma$ is not cyclic on $S$. Assume that $\pi_\sigma = \pi^1_\sigma \cdot \cdots \cdot \pi^t_\sigma$ for some $t \geq 2$, where $\pi^h_\sigma$ is a cyclic permutation on $S_h$ and can be regarded as a directed cycle on $S_h$ for each $1 \leq h \leq t$. Note that $S_1, \ldots, S_t$ is a partition of $S$. For each $1 \leq h \leq t$, we define a subsequence $\sigma_h = r^h_1 \cdots r^h_{k_h}$ of a request sequence $\sigma$ such that $s_{\text{grdy}}(r^h_j; \sigma_h|S) \in S_h$ and $s_{\text{opt}}(r^h_j; \sigma_h|S) \in S_h$ for each $1 \leq j \leq k_h$. Then we have that $|\sigma_h| = |S_h|$ for each $1 \leq h \leq t$. The following claims hold.

**Claim 5.1.** For each $1 \leq h \leq t$, $GRDY(\sigma_h; \sigma|S) = GRDY(\sigma_h; \sigma_h|S_h)$.

**Claim 5.2.** For each $1 \leq h \leq t$, $\text{OPT}(\sigma_h; \sigma|S) = \text{OPT}(\sigma_h; \sigma_h|S_h)$. 

17
The proofs of Claims 5.1 and 5.2 are given in Sections A.1 and A.2, respectively. Recall that $\pi_h$ is a cyclic permutation on $S_h$ for each $1 \leq h \leq t$. Then from Theorem 5.2, we have that $GRDy(\sigma_h; \sigma_h|S_h) \leq (4 \cdot U(S_h) - 1) \cdot OPT(\sigma_h; \sigma_h|S_h)$ for each $1 \leq h \leq t$. Thus

$$GRDy(\sigma|S) = GRDy(\sigma; \sigma|S) = \sum_{h=1}^{t} GRDy(\sigma_h; \sigma|S)$$

$$= \sum_{h=1}^{t} GRDy(\sigma_h; \sigma_h|S_h) \leq \sum_{h=1}^{t} (4 \cdot U(S_h) - 1) \cdot OPT(\sigma_h; \sigma_h|S_h)$$

$$\leq (4 \cdot U(S) - 1) \cdot \sum_{h=1}^{t} OPT(\sigma_h; \sigma_h|S_h) = (4 \cdot U(S) - 1) \cdot \sum_{h=1}^{t} OPT(\sigma; \sigma|S)$$

$$= (4 \cdot U(S) - 1) \cdot OPT(\sigma; \sigma|S) = (4 \cdot U(S) - 1) \cdot OPT(\sigma|S),$$

where the 3rd equality is due to Claim 5.1, the 1st inequality is due to Theorem 5.2, and the 4th equality is due to Claim 5.2, and this completes the proof of the theorem.  

5.3 Competitive Ratio for Greedy Algorithm for OFAL$_{eq}(k, \ell)$

As an immediate consequence, we have the following corollary to Theorems 5.1 and 5.3.

**Corollary 5.2.** For OFAL$_{eq}(k, \ell)$ such that $k \geq 2$, $R(\text{GRDy}) = 4k - 5$.

**Proof:** Since the distance between adjacent servers $s_j$ and $s_{j+1}$ is the same, i.e., $d_j = s_{j+1} - s_j = d$ for each $1 \leq j \leq k - 1$, it is immediate that

$$U(S) = \frac{s_k - s_1}{d_{\text{min}}} = \frac{\sum_{j=1}^{k-1} (s_{j+1} - s_j)}{\min_{1 \leq j \leq k-1} \{s_{j+1} - s_j\}} = \frac{\sum_{j=1}^{k-1} d}{\min_{1 \leq j \leq k-1} d} = k - 1.$$

From Theorem 5.3, it follows that $R(\text{GRDy}) \leq 4(k - 1) - 1 = 4k - 5$. Since $R(\text{GRDy}) \geq 4k - 5$ for OFAL$_{eq}(k, \ell)$ by Theorem 5.1, we have that $R(\text{GRDy}) = 4k - 5$.  

6 A Lower Bound on the Competitive Ratio of MPFS

In this section, we derive a lower bound on the competitive ratio of algorithms in $\mathcal{MPFS}$.

**Theorem 6.1.** Let $\text{ALG} \in \mathcal{MPFS}$ for OFAL$(k, \ell)$. Then $R(\text{ALG}) \geq 2L(S) + 1$, where $L(S) = 0$ for $k = 1$, and for any $k \geq 2$,

$$L(S) = \frac{s_k - s_1}{d_{\text{max}}} = \frac{s_k - s_1}{\max_{1 \leq j \leq k-1} d_j}.$$

Note that $d_j = s_{j+1} - s_j$ for each $1 \leq j \leq k - 1$ as defined in (2.1).

Before presenting the proof of Theorem 6.1, we introduce several notions, e.g., surrounding servers [22, 3], surrounding-oriented algorithms [22, 3], specification of algorithms (in Definition 6.3), and feature points (in Definition 6.4) and we also provide several technical lemmas related to those notions.
**Definition 6.1.** Given a request \( r \) for OFAL\((k, \ell)\), the surrounding servers for \( r \) are \( s^L \) and \( s^R \), where \( s^L \) is the closest free server to the left of \( r \) (if any) and \( s^R \) is the closest free server to the right of \( r \) (if any). If \( r = s \) for some \( s \in S \) and \( s \) is free, then the surrounding server of \( r \) is only the server \( s \).

**Definition 6.2.** Let \( \text{alg} \) be an online algorithm for OFAL\((k, \ell)\). We say that \( \text{alg} \) is surrounding-oriented for a request sequence \( \sigma \) if it matches every request \( r \) of \( \sigma \) with one of the surrounding servers of \( r \). We say that \( \text{alg} \) is surrounding-oriented if it is surrounding-oriented for every request sequence \( \sigma \).

For surrounding-oriented algorithms, the following useful lemma [3, 16] is known.

**Lemma 6.1.** Let \( \text{alg} \) be an online algorithm for OFAL\((k, \ell)\). Then there exists a surrounding-oriented algorithm \( \text{alg}' \) for OFAL\((k, \ell)\) such that \( \text{alg}'(\sigma) \leq \text{alg}(\sigma) \) for any \( \sigma \).

According to Lemma 6.1, it suffices to consider only \( \text{alg} \in MPFS \) that is surrounding-oriented. To complete the proof of Theorem 6.1, the following notions are necessary.

**Definition 6.3.** Let \( \text{alg} \in MPFS \) for OFAL\((k, \ell)\). For any pair of \( 1 \leq i < j \leq k \), we say that \( \text{alg} \) follows the specification \( \text{spec}(i, j) \) if \( \text{alg} \) matches a request \( r \) with the server \( s_i \) when \( s_i \) is free, \( s_{i+1}, \ldots, s_j \) are full, and \( r \) occurs on \( s_j \).

**Definition 6.4.** For any \( k \geq 3 \), let \( \text{alg} \in MPFS \) for OFAL\((k, \ell)\). We say that \( P_{\text{alg}} = \{p_1, \ldots, p_m\} \subseteq \{2, \ldots, k-1\} \) is a set of feature points of \( \text{alg} \) if it satisfies the following conditions: Let \( p_0 = 1 \) and for each \( 0 \leq i \leq m-1 \),

\[
p_{i+1} = \max\{j \in \{p_i + 1, \ldots, k - 1\} : \text{alg follows spec}(p_i, j)\}.
\]

For \( \text{alg} \in MPFS \), if \( p_1 \) cannot be defined, then \( P_{\text{alg}} = \emptyset \).

By classifying algorithms in \( MPFS \) due to Definition 6.4, we show Theorem 6.1.

**Proof of Theorem 6.1:** For simplicity, assume that \( s_1 = 0 \) and then \( L(S) = \frac{s_k}{d_{\text{max}}} \). Let 

\[
a_j = \frac{s_{j+1} + s_k}{s_k + d_j}
\]

for each \( 1 \leq j \leq k - 1 \), and it is immediate that \( s_j < a_j < s_{j+1} \).

Let \( k = 1 \) and \( \text{alg}' \in MPFS \) for OFAL\((1, \ell)\). Since \( R(\text{alg}') = 1 \) and \( L(S) = 0 \) for \( k = 1 \), we have that \( R(\text{alg}') = 1 = 2L(S) + 1 \). Let \( k = 2 \) and \( \text{alg}'' \in MPFS \) for OFAL\((2, \ell)\). Since \( R(\text{alg}'') \geq 3 \) [15, Theorem 3.7] and \( L(S) = 1 \) for \( k = 2 \), we have that \( R(\text{alg}'') \geq 3 = 2L(S) + 1 \). Thus it suffices to consider the case that \( k \geq 3 \).

For \( k \geq 3 \), fix \( \text{alg} \in MPFS \) for OFAL\((k, \ell)\) arbitrarily and let \( P_{\text{alg}} = \{p_1, \ldots, p_m\} \subseteq \{2, \ldots, k-1\} \) be the set of feature points of \( \text{alg} \). To derive a lower bound on the competitive ratio of \( \text{alg} \in MPFS \), we construct a request sequence \( \sigma \) as follows:

1. For each \( 1 \leq j \leq k \), generate \( \ell - 1 \) requests on \( s_j \), which leads to the state that the remaining capacity of \( s_j \) is 1 for each \( 1 \leq j \leq k \).
2. For each \( 2 \leq j \leq p_m \) such that \( j \not\in P_{\text{alg}} \), generate a request on \( s_j \), which leads to the state that \( s_j \) is full for each \( 2 \leq j \leq p_m \) such that \( j \not\in P_{\text{alg}} \), and the remaining capacity of \( s_h \) is 1 for \( h = 1, h \in P_{\text{alg}} \), or \( p_m < h \leq k \).
3. Generate a request \( r^{(1)} \) on \( a_{p_m} \) and let \( s^{(1)} \) be the server with which \( \text{alg} \) matches \( r^{(1)} \). Note that \( s_{p_m} < a_{p_m} < s_{p_m+1} \).
4. Generate a request \( r^{(i+1)} \) on \( s^{(i)} \) for each \( i \geq 1 \), and continue the process until a request is generated on \( s_1 \) or \( s_k \).

Since \( \text{ALG} \) is surrounding-oriented, it suffices to consider the following two cases: (Case 1) \( \text{ALG} \) matches \( r^{(1)} \) with \( s_{p_m+1} \); (Case 2) \( \text{ALG} \) matches \( r^{(1)} \) with \( s_{p_m} \).

(Case 1) Since \( \text{ALG} \) matches \( r^{(1)} \) with \( s_{p_m+1} \), we have two surrounding servers \( s_{p_m} \) and \( s_{p_m+2} \) for \( r^{(2)} \) appearing on \( s_{p_m+1} \). If \( \text{ALG} \) matches \( r^{(2)} \) with \( s_{p_m} \), then this implies that \( \text{ALG} \) follows \( \text{SPEC}(p_m, p_m+1) \), but \( p_m \) is the last element of \( P_{\text{alg}} \). Thus \( \text{ALG} \) must match \( r^{(2)} \) with \( s_{p_m+2} \). From this observation, it is obvious that \( \text{ALG} \) matches \( r^{(i+1)} \) with \( s_{p_m+i+1} \) for each \( 1 \leq i \leq k-p_m-1 \) and matches \( r^{(k-p_m+1)} \) with \( s_{p_m} \), where \( r^{(k-p_m+1)} \) is the last request of the request sequence \( \sigma \). Then we have that

\[
\text{ALG}(\sigma | S) = \sum_{i=1}^{k-p_m-1} (s_{p_m+i+1} - s_{p_m+i}) + (s_{p_m+1} - a_{p_m}) + (s_k - s_{p_m})
\]

and it is immediate that \( \text{OPT}(\sigma | S) \leq a_{p_m} - s_{p_m} \). Thus it follows that

\[
\frac{\text{ALG}(\sigma | S)}{\text{OPT}(\sigma | S)} = \frac{2 \cdot (s_k - s_{p_m}) - (a_{p_m} - s_{p_m})}{a_{p_m} - s_{p_m}} = 2 \cdot \frac{s_k - s_{p_m}}{a_{p_m} - s_{p_m}} - 1
\]

\[
= 2 \cdot \frac{s_k - s_{p_m}}{s_{p_m+1} - s_{p_m}} - 1 = 2 \cdot \frac{(s_k - s_{p_m}) \cdot (s_k + d_{p_m})}{d_{p_m} \cdot (s_k - s_{p_m})} - 1
\]

\[
= 2 \cdot \frac{s_k}{d_{p_m}} + 1 \geq 2 \cdot \frac{s_k}{d_{\max}} + 1 = 2 \cdot L(S) + 1.
\]

(Case 2) Recalling that \( \text{ALG} \) matches \( r^{(1)} \) with \( s_{p_m} \), we have two surrounding servers \( s_{p_m-1} \) and \( s_{p_m+1} \) for \( r^{(2)} \) appearing on \( s_{p_m} \). Since \( \text{ALG} \) follows \( \text{SPEC}(p_{m-1}, p_m) \), \( \text{ALG} \) must match \( r^{(2)} \) with \( s_{p_m-1} \). From this observation, it is obvious that \( \text{ALG} \) matches \( r^{(i+1)} \) with \( s_{p_m-i} \) for each \( 1 \leq i \leq m \), where \( p_0 = 1 \), and matches \( r^{(m+2)} \) with \( s_{p_m+1} \), where \( r^{(m+2)} \) is the last request of the request sequence \( \sigma \). Then we have that

\[
\text{ALG}(\sigma | S) = (a_{p_m} - s_{p_m}) + (s_{p_m+1} - s_1) + \sum_{i=1}^{m} (s_{p_m-i+1} - s_{p_m-i})
\]

\[
= (s_{p_m+1} - s_1) + (a_{p_m} - s_1) = s_{p_m+1} + a_{p_m},
\]

where the last equality follows from the assumption that \( s_1 = 0 \), and it is immediate that \( \text{OPT}(\sigma | S) \leq s_{p_m+1} - a_{p_m} \). Then it follows that

\[
\frac{\text{ALG}(\sigma | S)}{\text{OPT}(\sigma | S)} \geq \frac{s_{p_m+1} + a_{p_m}}{s_{p_m+1} - a_{p_m}} = \frac{s_{p_m+1} + \frac{s_{p_m+1} \cdot s_k}{s_k + d_{p_m}}}{s_{p_m+1} - \frac{s_{p_m+1} \cdot s_k}{s_k + d_{p_m}}}
\]

\[
= 2 \cdot \frac{s_k}{d_{p_m}} + 1 \geq 2 \cdot \frac{s_k}{d_{\max}} + 1 = 2 \cdot L(S) + 1,
\]

and this completes the proof of the theorem.

As an immediate corollary to Theorem 6.1, we have the following lower bound on the competitive ratio of any \( \text{ALG} \in \mathcal{MPFS} \) for \( \text{OFAL}_{eq}(k, \ell) \).
Corollary 6.1. Let $\text{alg} \in \mathcal{MPFS}$ for $\text{OFAL}_{eq}(k, \ell)$. Then $\mathcal{R}(\text{alg}) \geq 2k - 1$.

Proof: Since $d_j = s_{j+1} - s_j = d$ for each $1 \leq j \leq k - 1$, we have that

$$L(S) = \frac{s_k - s_1}{\max_{1 \leq j \leq k-1} d_j} = \frac{\sum_{j=1}^{k-1} (s_{j+1} - s_j)}{\max_{1 \leq j \leq k-1} (s_{j+1} - s_j)} = \frac{(k - 1) \cdot d}{d} = k - 1.$$  

Thus the corollary follows from Theorem 6.1. \hfill \blacksquare

7 An Optimal MPFS Algorithm

In this section, we propose a new algorithm $\text{IDAS} \in \mathcal{MPFS}$ (Interior Division for Adjacent Servers), and we show that $\text{IDAS}$ is $(2k - 1)$-competitive for $\text{OFAL}_{eq}(k, \ell)$. From Corollary 6.1, we can conclude that $\text{IDAS}$ is best possible in the class $\mathcal{MPFS}$ for $\text{OFAL}_{eq}(k, \ell)$.

7.1 A New Algorithm: Interior Division for Adjacency Servers

Before presenting the algorithm $\text{IDAS}$, we provide several notations. Fix $a, b \in \mathbb{R}$ with $a < b$ arbitrarily. For any $x, y \in \mathbb{R}$ such that $a \leq x < y \leq b$, let $B(x, y)$ be the point that internally divides the line segment $[x, y]$ into $b - x$ to $y - a$, i.e.,

$$B(x, y) = \frac{(b - x)y + (y - a)x}{(b - x) + (y - a)} = \frac{by - ax}{b - a + y - x}.$$  

Note that $x < B(x, y) < y$ and $B(x, y)$ implies a boundary between $x$ and $y$.

Given the set $S = \{s_1, \ldots, s_k\}$ of $k$ servers with $s_1 < \cdots < s_k$, we fix parameters $a, b \in \mathbb{R}$ such that $a \leq s_1 < s_k \leq b$. Then the algorithm $\text{IDAS}_{[a,b]}$ can be described in Algorithm 1.

Algorithm 1: $\text{IDAS}_{[a,b]}$ (Interior Division for Adjacent Servers)

For a request $r$, let $\text{ss}(r)$ be the set of surrounding servers for $r$.

1. If $|\text{ss}(r)| = 1$, then let $\text{ss}(r) = \{s_\ast\}$, where $s_\ast \in S$ is the unique surrounding server for $r$, and match $r$ with $s_\ast$.
2. If $|\text{ss}(r)| = 2$, then let $s^L$ be the left surrounding server for $r$ and $s^R$ be the right surrounding server for $r$.
   (a) If $r \leq B(s^L, s^R)$, then match $r$ with $s^L$;
   (b) If $B(s^L, s^R) < r$, then match $r$ with $s^R$.

It is immediate that $\text{IDAS}_{[a,b]}$ is surrounding-oriented (see Definition 6.2). In a way similar to Lemma 5.1, we can show the following lemma:

Lemma 7.1. $\text{IDAS}_{[a,b]}$ is faithful for $\text{OFAL}(k, \ell)$.

Hence, to derive an upper bound on the competitive ratio of $\text{IDAS}_{[a,b]}$, it suffices to consider only opposite request sequences w.r.t. $\text{IDAS}_{[a,b]}$ (see Lemma 4.5).

To observe that $\text{IDAS}_{[a,b]} \in \mathcal{MPFS}$, the following property of $B(\ast, \ast)$ is crucial and the boundary $B(\ast, \ast)$ naturally induces a total order $\preceq_{\rho}$. 

21
Property 7.1. Fix $a, b \in \mathbb{R}$ with $a < b$ arbitrarily. For any $x, y, z \in \mathbb{R}$ such that $a \leq x < y < z \leq b$, $B(x, y) < B(x, z) < B(y, z)$.

Proof: This follows from the straightforward calculations:

$$B(x, z) - B(x, y) = \frac{(b-a)(z-y)(b-x)}{(b-a+z-x)(b-a+y-x)} > 0;$$

$$B(y, z) - B(x, z) = \frac{(b-a)(y-x)(z-a)}{(b-z+y-a)(b-a+z-x)} > 0,$$

where the inequalities follow from the assumption that $a \leq x < y < z \leq b$.

We define the following binary relation $\leq_{\rho}$ on $[a, b]$ with a parameter $\rho \in \mathbb{R}$.

Definition 7.1. For any $a, b \in \mathbb{R}$, let $[a, b]$ be the closed interval and fix $\rho \in \mathbb{R}$ arbitrarily. For any $x, y \in [a, b]$, we write $x \leq_{\rho} y$ if one of the following conditions holds: (1) $x = y$; (2) $x < y$ and $B(x, y) < \rho$; (3) $y < x$ and $\rho \leq B(y, x)$.

For the binary relation $\leq_{\rho}$ on $[a, b]$, the following result holds.

Theorem 7.1. For any $\rho \in \mathbb{R}$, $\leq_{\rho}$ is a total order on the closed interval $[a, b]$.

The proof of the theorem is straightforward and is given in Appendix B. We summarize the properties of the total order $\leq_{\rho}$ in the following remark.

Remark 7.1. For any $a, b \in \mathbb{R}$ such that $a < b$, let $[a, b]$ be the closed interval. Then for any $x, y \in [a, b]$ and any $\rho \in \mathbb{R}$, the following properties hold:

1. if $\rho < x < y$, then $y \leq_{\rho} x \leq_{\rho} \rho$;
2. if $x < y < \rho$, then $x \leq_{\rho} y \leq_{\rho} \rho$;
3. if $\rho < a$, then $x \leq_{\rho} y$ iff $x \geq y$;
4. if $b < \rho$, then $x \leq_{\rho} y$ iff $x \leq y$;
5. if $\rho \in [a, b]$, then $x \leq_{\rho} \rho$.

The property (5) implies that $\rho$ is the maximum in $[a, b]$ w.r.t. the total order $\leq_{\rho}$.

From Remark 7.1, the following alternative definition is equivalent to that of IDAS$_{[a,b]}$.

Definition 7.2. For a request sequence $\sigma = r_1 \cdots r_i \cdots r_{k\ell}$, the algorithm IDAS$_{[a,b]}$ (Interior Division for Adjacent Servers) for OFAL($k, \ell$) works as follows: For each $1 \leq i \leq k\ell$, it matches a request $r_i$ with the highest free server$^3$ $s \in S$ w.r.t. the total order $\leq_{r_i}$.

From Definition 7.2, it is immediate that IDAS$_{[a,b]} \in \mathcal{MPFS}$.

7.2 An Upper Bound on the Competitive Ratio

In this subsection, we derive an upper bound on the competitive ratio of IDAS$_{[a,b]}$ for OFAL($k, \ell$), which leads to show that IDAS$_{[a,b]}$ is best possible in $\mathcal{MPFS}$ for OFAL($k, \ell$).

---

$^3$ For a request $r$, we say that $s \in S$ is the highest free server w.r.t. $\leq_r$ if $s$ is free and $s' \leq_r s$ for all free servers $s' \in S$ just before matching $r$ to a server.
7.2.1 Single Tour for IDAS

Similarly to the discussion on GRDY in Section 5.2, we first consider a request sequence $\sigma$ with a single tour w.r.t. IDAS$_{[a,b]}$ (and consider a request sequence $\sigma$ with multiple tours w.r.t. IDAS$_{[a,b]}$ in Section 7.2.2).

For a conflicting pair $(v_i, v_j)$ in a tour $T : v_1 \to \cdots \to v_n \to v_1$, we have that $v_i \leq v_{j+1} < v_{j+1} \leq v_j$ by definition. Then the following cases are possible: (1) $v_i = v_{j+1}$ and $v_{i+1} < v_j$; (2) $v_i < v_{j+1}$ and $v_{i+1} = v_j$; (3) $v_i < v_{j+1}$ and $v_{i+1} < v_j$; (4) $v_i = v_{j+1}$ and $v_{i+1} = v_j$. For $n > 2$, the case (4) never occurs, because the case (4) implies that $v_i \rightarrow v_{i+1} \rightarrow v_i$ is a tour of length 2, but $T$ is a tour of length $n > 2$. For $n > 2$, let

$$c(v_i, v_j | T) = \begin{cases} b - v_i & \text{if } v_i = v_{j+1}; \\ v_j - a & \text{if } v_{i+1} = v_j; \\ b - a & \text{if } v_i < v_{j+1} \text{ and } v_{i+1} < v_j, \end{cases}$$

and $c(v_1, v_2 | T) = |v_2 - v_1|$ for $n = 2$. Define the cost of $T$ by

$$C(T) = \sum_{(u,v) \in cf(T)} c(v_i, v_j | T).$$

In a way similar to $C(T)$, we can define $C(\overline{T})$ for the contracted tour $\overline{T}$ of $T$.

**Lemma 7.2.** For a tour $T$, let $\overline{T}$ be the contracted tour of $T$. Then $\ell(\overline{T}) \leq 2 \cdot C(\overline{T})$.

**Proof:** For a tour $T$, let $\overline{T}$ be the contracted tour with $2m$ turning points. We show the lemma by induction on $m \geq 1$. For $m = 1$, it is immediate that $\overline{T}_1 : t_1 \to t_2 \to t_1$ with $t_1 < t_2$ has a single conflicting pair $(t_1, t_2)$. Then $\ell(\overline{T}_1) = 2 \cdot (t_2 - t_1) = 2 \cdot C(\overline{T}_1)$.

For any $m \geq 2$, assume that $\ell(\overline{T}_{m-1}) \leq 2 \cdot C(\overline{T}_{m-1})$ for any $\overline{T}_{m-1}$ with $2(m-1)$ turning points. We show that $\ell(\overline{T}_m) \leq 2 \cdot C(\overline{T}_m)$ for any $\overline{T}_m : t_1 \to \cdots \to t_{2m} \to t_1$. Since $m \geq 2$, there must exist a detour $D$ in $\overline{T}_m$ by Lemma 4.2. Without loss of generality, assume that $t_1 \to t_2 \to t_3 \to t_4$ is a detour $D$ in $\overline{T}_m$, where $t_1 < t_3 < t_2 < t_4$ or $t_1 > t_2 > t_3 > t_4$. Consider the case that $t_1 < t_3 < t_2 < t_4$ (the other case can be discussed analogously). For $\overline{T}_{m-1} : t_1 \to t_4 \to \cdots \to t_{2m} \to t_1$ defined by contracting $D$ in $\overline{T}_m$, we have that $\ell(\overline{T}_{m-1}) \leq 2 \cdot C(\overline{T}_{m-1})$ by the induction hypothesis. Note that $t_2$ and $t_3$ are removed in $\overline{T}_{m-1}$. By Definition 4.1, $(t_1, t_j)$ is a conflicting pair in $\overline{T}_{m-1}$ for each $4 \leq j \leq 2m$ if $t_1 \leq t_{j+1} < t_4 \leq t_j$.

**Claim 7.1.** For some $4 \leq j \leq 2m$, if $(t_i, t_j) \in cf(\overline{T}_{m-1})$, then either $(t_i, t_j) \in cf(\overline{T}_m)$ or $(t_3, t_j) \in cf(\overline{T}_m)$ holds.

The proof of Claim 7.1 is given in Section C.1. According to Claim 7.1, partition $cf(\overline{T}_{m-1})$ into $cf^{(1)}(\overline{T}_{m-1})$, $cf^{(3)}(\overline{T}_{m-1})$, and $cf^*(\overline{T}_{m-1})$ as follows:

$$cf^{(1)}(\overline{T}_{m-1}) = \{ (t_i, t_j) \in cf(\overline{T}_{m-1}) : (t_i, t_j) \in cf(\overline{T}_m) \};$$
$$cf^{(3)}(\overline{T}_{m-1}) = \{ (t_i, t_j) \in cf(\overline{T}_{m-1}) : (t_3, t_j) \in cf(\overline{T}_m) \} \setminus cf^{(1)}(\overline{T}_{m-1});$$
$$cf^*(\overline{T}_{m-1}) = \{ (t_i, t_j) \in cf(\overline{T}_{m-1}) : i \neq 1 \};$$

and we also partition $cf(\overline{T}_m)$ into $cf^{(2)}(\overline{T}_m)$, $cf^{(3)}(\overline{T}_m)$, $cf^{(1)}(\overline{T}_m)$, and $cf^*(\overline{T}_m)$ as follows:

$$cf^{(2)}(\overline{T}_m) = \{ (t_i, t_j) \in cf(\overline{T}_m) : j = 2 \};$$
$$cf^{(3)}(\overline{T}_m) = \{ (t_i, t_j) \in cf(\overline{T}_m) : j \neq 2 \land i = 3 \};$$
$$cf^{(1)}(\overline{T}_m) = \{ (t_i, t_j) \in cf(\overline{T}_m) : j \neq 2 \land i = 1 \};$$
$$cf^*(\overline{T}_m) = \{ (t_i, t_j) \in cf(\overline{T}_m) : j \neq 2 \land i \notin \{1, 3\} \}.$$

23
For these partitions, we have the following claims:

Claim 7.2. $c(t_i, t_j | \bar{T}_{m-1}^*) = c(t_i, t_j | T_m)$ for each $(t_i, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)$.

Claim 7.3. $c(t_1, t_j | \bar{T}_{m-1}^*) = c(t_3, t_j | T_m)$ for each $(t_1, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)$.

Claim 7.4. $c(t_1, t_j | \bar{T}_{m-1}^*) \leq c(t_1, t_j | T_m)$ for each $(t_1, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)$.

The proofs of Claims 7.2, 7.3, and 7.4 are given in Sections C.2, C.3, and C.4, respectively. Then from these claims, it follows that

$$C(\bar{T}_{m-1}^*) = \sum_{(t_i, t_j) \in \text{cf}(\bar{T}_{m-1}^*)} c(t_i, t_j | \bar{T}_{m-1}^*)$$

$$= \sum_{(t_1, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)} c(t_1, t_j | \bar{T}_{m-1}^*)$$

$$+ \sum_{(t_1, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)} c(t_1, t_j | T_m)$$

$$\leq \sum_{(t_1, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)} c(t_1, t_j | \bar{T}_m) + \sum_{(t_1, t_j) \in \text{cf}^*(\bar{T}_{m-1}^*)} c(t_1, t_j | T_m)$$

$$= C(\bar{T}_m) - \sum_{(t_i, t_j) \in \text{cf}^*(\bar{T}_m)} c(t_i, t_j | \bar{T}_m) + \sum_{(t_i, t_j) \in \text{cf}^*(\bar{T}_m)} c(t_i, t_j | \bar{T}_m)$$

$$\leq C(\bar{T}_m) - c(t_1, t_2 | \bar{T}_m) \leq C(\bar{T}_m) - (t_2 - t_1). \quad (7.1)$$

Since $\ell(\bar{T}_m) = \ell(T_{m-1}^*) + 2 \cdot (t_2 - t_3)$, we have that

$$\ell(\bar{T}_m) = \ell(\bar{T}_{m-1}^*) + 2 \cdot (t_2 - t_3) \leq 2 \cdot C(\bar{T}_{m-1}^*) + 2 \cdot (t_2 - t_3)$$

$$\leq 2 \cdot \{C(\bar{T}_m) - (t_2 - t_1)\} + 2 \cdot (t_2 - t_3)$$

$$\leq 2 \cdot C(\bar{T}_m) + 2 \cdot (t_1 - t_3) \leq 2 \cdot C(\bar{T}_m),$$

where the 1st inequality is due to the induction hypothesis, the 2nd inequality is due to Eq. (7.1), and the last inequality is due to the assumption that $t_1 < t_3 < t_2 < t_4$.

Lemma 7.3. For a tour $T$, let $\bar{T}$ be the contracted tour of $T$. Then $C(\bar{T}) \leq C(T)$.

Proof: For a tour $T : v_1 \to \cdots \to v_n \to v_1$, let $\bar{T} : t_1 \to \cdots \to t_{2m} \to t_1$ be the contracted tour of $T$. Recall that the injection $f_{\text{inj}} : \text{cf}(\bar{T}) \to \text{cf}(T)$ was defined in Lemma 4.1 as
follows: For each \((t_i, t_j) \in \text{cf}(\bar{T})\), we have that \(t_i \leq t_{i+1} < t_{i+1} \leq t_j\) in \(\bar{T}\). Let \(T^i : t_i = v^i_1 \rightarrow \cdots \rightarrow v^i_x = t_{i+1}\) (resp. \(T^j : t_j = v^j_1 \rightarrow \cdots \rightarrow v^j_y = t_{j+1}\)) be the path from \(t_i\) to \(t_{i+1}\) (resp. from \(t_j\) to \(t_{j+1}\)) in \(T\). Let \(1 \leq \alpha < x\) be the maximum with \(v^i_\alpha \leq t_{j+1}\) and \(1 \leq \beta < y\) be the maximum with \(v^j_\beta \leq t_{j+1}\). Since \(1 \leq \alpha < x\) and \(1 \leq \beta < y\), we have that \(\alpha = 1\) and \(\beta + 1 = y\), i.e., \(t_i = v^i_1 = v^i_\alpha = v^j_{\beta+1} = v^j_y = t_{j+1}\). Then

\[c(t_i, t_j|\bar{T}) = \bar{c} - t_i - t_j = v^j_{\beta+1} - v^i_\alpha = c(v^i_\alpha, v^j_{\beta+1}|T).\]

For case (3), we have that \(v^j_{\beta+1} = v^i_\alpha\). If \(v^j_{\beta+1} \in \text{relay}(t_j)\) or \(v^i_\alpha \in \text{relay}(t_i)\), then it follows that \(T\) visits \(v^j_{\beta+1}\) or \(v^i_\alpha\) more than once, but this is impossible by the definition of \(T\). Thus \(v^j_{\beta+1} \not\in \text{relay}(t_j)\) and \(v^i_\alpha \not\in \text{relay}(t_i)\). Since \(1 \leq \alpha < x\) and \(1 \leq \beta < y\), we have that \(\alpha = 1\) and \(\beta + 1 = y\), i.e., \(t_i = v^i_1 = v^i_\alpha = v^j_{\beta+1} = v^j_y = t_{j+1}\). Then

\[c(t_i, t_j|\bar{T}) = b - t_i - b - v^i_\alpha = c(v^i_\alpha, v^j_{\beta+1}|T).\]

For case (4), it is immediate that \(v^j_{\beta+1} = v^i_\alpha\). In a way similar to the case (3), we have that \(\alpha + 1 = x\) and \(\beta = 1\), i.e., \(t_i = v^i_1 = v^i_\alpha = v^j_{\beta+1} = v^j_1 = t_j\). Then

\[c(t_i, t_j|\bar{T}) = t_j - a - v^j_\beta = a = c(v^i_\alpha, v^j_{\beta+1}|T).\]

Thus we can conclude that \(c(t_i, t_j|\bar{T}) \leq c(v^i_\alpha, v^j_{\beta+1}|T)\) for each \((t_i, t_j) \in \text{cf}(\bar{T})\). Then

\[C(\bar{T}) = \sum_{(t_i, t_j) \in \text{cf}(\bar{T})} c(t_i, t_j|\bar{T}) \leq \sum_{(t_i, t_j) \in \text{cf}(\bar{T})} c(f_{\text{inj}}(t_i, t_j)|T) \leq \sum_{(v_i, v_j) \in \text{cf}(T)} c(v_i, v_j|T) = C(T),\]

where the 2nd inequality is due to the fact that \(\{f_{\text{inj}}(t_i, t_j) : (t_i, t_j) \in \text{cf}(\bar{T})\} \subseteq \text{cf}(T)\). □

**Lemma 7.4.** For any opposite request sequence \(\sigma\), if \(\sigma\) has a single tour \(T_\sigma\) on \(S\) w.r.t. IDAS\(_{[a,b]}\), then \(\text{OPT}(\sigma|S) \geq \frac{d_{\min}}{b-a+2d_{\min}} \cdot C(T_\sigma)\).

**Proof:** Fix an opposite request sequence \(\sigma = r_1 \cdots r_k\) w.r.t. IDAS\(_{[a,b]}\) arbitrarily, and let \(T_\sigma : s_{h_1} \rightarrow \cdots \rightarrow s_{h_k} \rightarrow s_{h_1}\) be the single tour on \(S\) w.r.t. IDAS\(_{[a,b]}\). We partition the set \(c_{\text{idas}}(\sigma)\) of consuming pairs in \(\sigma\) (see Definition 5.1) into \(c_{\text{idas}}^+(\sigma)\) and \(c_{\text{idas}}^-(\sigma)\) as follows:

\[c_{\text{idas}}^+(\sigma) = \{(r_i, r_j) \in c_{\text{idas}}(\sigma) : i < j\};\]

\[c_{\text{idas}}^-(\sigma) = \{(r_i, r_j) \in c_{\text{idas}}(\sigma) : i > j\}.\]

For each request \(a\) in \(\sigma\), let \(c_{\text{idas}}^+(a; \sigma)\) be the set of consuming pairs \((a, r)\) in \(c_{\text{idas}}^+(\sigma)\), i.e., \(c_{\text{idas}}^+(a; \sigma) = \{(a, r) : (a, r) \in c_{\text{idas}}^+(\sigma)\}\), and enumerate nonempty
where the inequality is due to Lemma 7.4. Thus it follows that $S$ has a single tour on

Claim 7.6. Let $T_0$. Theorem 7.2.

Claim 7.7. Let $\sigma$ by letting $\sigma \in C$. Thus the lemma immediately follows from Claims 7.5 and 7.6. Then we have the following claims:

**Claim 7.5.** For each $1 \leq i \leq \mu$, the following inequality holds:

$$\text{OPT}(a_i; \sigma|S) \geq \frac{d_{\text{min}}}{b-a} \sum_{(a_i, r) \in \text{cs}^+(a_i; \sigma)} c \left(f^{-1}(a_i, r)|T_\sigma\right).$$

**Claim 7.6.** For each $1 \leq j \leq \nu$, the following inequality holds:

$$\text{OPT}(b_j; \sigma|S) \geq \frac{d_{\text{min}}}{b-a} \sum_{(r, b_j) \in \text{cs}^+(b_j; \sigma)} c \left(f^{-1}(r, b_j)|T_\sigma\right).$$

The proofs of Claims 7.5 and 7.6 are given in Sections C.5 and C.6, respectively. Then

$$C(T_\sigma) = \sum_{(s_i, s_j) \in \text{cs}(T_\sigma)} c(s_i, s_j|T_\sigma) = \sum_{(r, r_j) \in \text{cs}^+(\sigma)} c \left(f^{-1}(r, r_j)|T_\sigma\right)$$

$$= \sum_{(r_i, r_j) \in \text{cs}^+(\sigma)} c \left(f^{-1}(r_i, r_j)|T_\sigma\right) + \sum_{(r_i, r_j) \in \text{cs}^+(\sigma)} c \left(f^{-1}(r_i, r_j)|T_\sigma\right)$$

$$= \sum_{i=1}^\mu \sum_{(a_i, r) \in \text{cs}^+(a_i; \sigma)} c \left(f^{-1}(a_i, r)|T_\sigma\right) + \sum_{j=1}^\nu \sum_{(r, b_j) \in \text{cs}^+(b_j; \sigma)} c \left(f^{-1}(r, b_j)|T_\sigma\right);$$

$$\text{OPT}(\sigma|S) \geq \sum_{i=1}^\mu \text{OPT}(a_i; \sigma|S) + \sum_{j=1}^\nu \text{OPT}(b_j; \sigma|S).$$

Thus the lemma immediately follows from Claims 7.5 and 7.6.

By applying Lemma 4.6 to $\text{IDAS}_{[a, b]}$ for OFAL($k, \ell$), we can show the following theorem.

**Theorem 7.2.** Let $\sigma$ be an opposite request sequence w.r.t. $\text{IDAS}_{[a, b]}$ for OFAL($k, \ell$). If $\sigma$ has a single tour on $S$, then

$$\text{IDAS}_{[a, b]}(\sigma|S) \leq \left(2 \cdot \frac{b-a}{d_{\text{min}} + 1}\right) \cdot \text{OPT}(\sigma|S).$$

**Proof:** Let $H(T_\sigma) = 2 \cdot C(T_\sigma)$. Then we have an upper bound on $\ell(T_\sigma)$, i.e.,

$$\ell(T_\sigma) = \ell(\tilde{T}_\sigma) \leq 2 \cdot C(\tilde{T}_\sigma) \leq 2 \cdot C(T_\sigma) = H(T_\sigma),$$

where the 1st inequality follows from Lemma 7.2 and the 2nd inequality follows from Lemma 7.3. On the other hand, we also have a lower bound on $\text{OPT}(\sigma|S)$, i.e.,

$$\text{OPT}(\sigma|S) \geq \frac{d_{\text{min}}}{b-a} \cdot C(T_\sigma) = \frac{2 \cdot C(T_\sigma)}{2 \cdot (\frac{b-a}{d_{\text{min}}} + 1)} = \frac{H(T_\sigma)}{2 \cdot (\frac{b-a}{d_{\text{min}}} + 1)},$$

where the inequality is due to Lemma 7.4. Thus it follows that

$$\text{IDAS}_{[a, b]}(\sigma|S) \leq \left(2 \cdot \frac{b-a}{d_{\text{min}} + 1}\right) \cdot \text{OPT}(\sigma|S)$$

by letting $c + 1 = 2 \cdot (\frac{b-a}{d_{\text{min}}} + 1)$ in Lemma 4.6. ■

26
The proof of Claim 7.7 is given in Section C.7. Recall that

\[ R \text{ Theorem 7.3.} \]

**Theorem 7.3.** \( R(IDAS_{[a,b]}(s)) \leq 2 \cdot \frac{b-a}{d_{\text{min}}} + 1 \) for OFAL\((k, \ell)\).

**Proof:** In Theorem 7.2, we already showed that \( IDAS_{[a,b]}(\sigma|S) \leq (2 \cdot \frac{b-a}{d_{\text{min}}} + 1) \cdot \text{OPT}(\sigma|S) \) for any request sequence \( \sigma = r_1 \cdots r_k \) with a single tour \( T_\sigma \) on \( S \), i.e., a bijection \( \pi_\sigma : \text{opt}(r_1; \sigma|S) \mapsto \text{idas}(r_1; \sigma|S) \) is cyclic on \( S \). In the remainder of the proof, we show that \( IDAS_{[a,b]}(\sigma|S) \leq (2 \cdot \frac{b-a}{d_{\text{min}}} + 1) \cdot \text{OPT}(\sigma|S) \) for any request sequence \( \sigma \) with multiple tours \( T_{\sigma_1}^{t_1}, \ldots, T_{\sigma_t}^{t_t} \) on \( S \), i.e., the bijection \( \pi_\sigma \) is not cyclic on \( S \). Assume that \( \pi_\sigma = \pi_\sigma^1 \circ \cdots \circ \pi_\sigma^t \) for some \( t \geq 2 \), where \( \pi_\sigma^h \) is a cyclic permutation on \( S_h \) and can be regarded as a directed cycle on \( S_h \) for each \( 1 \leq h \leq t \). Note that \( S_1, \ldots, S_t \) is a partition of \( S \). For each \( 1 \leq h \leq t \), we define a subsequence \( \sigma_h = r_{i_h}^{h} \cdots r_{k_h}^{h} \) of a request sequence \( \sigma \) such that \( \text{opt}(r_{j_h}^{h} \sigma_h|S) \in S_h \) and \( \text{opt}(r_{j_h}^{h} \sigma_h|S) \in S_h \) for each \( 1 \leq j \leq k_h \). Then we have that \( |\sigma_h| = |S_h| \) for each \( 1 \leq h \leq t \). The following claims hold.

**Claim 7.7.** \( IDAS_{[a,b]}(\sigma_h; \sigma|h|S_h) = IDAS_{[a,b]}(\sigma_h; \sigma|h|S_h) \) for each \( 1 \leq h \leq t \).

The proof of Claim 7.7 is given in Section C.7. Recall that \( \pi_\sigma^h \) is cyclic on \( S_h \) for each \( 1 \leq h \leq t \). Then from Theorem 7.2, it follows that for each \( 1 \leq h \leq t \),

\[
\text{IDAS}_{[a,b]}(\sigma_h; \sigma|h|S_h) \leq \left(2 \cdot \frac{b-a}{d_{\text{min}}^h} + 1\right) \cdot \text{OPT}(\sigma_h; \sigma|h|S_h),
\]

(7.2)

where \( d_{\text{min}}^h = \min\{|s_i - s_j| : s_i, s_j \in S_h \ (i \neq j)\} \). Thus we have that

\[
\text{IDAS}_{[a,b]}(\sigma|S) = \text{IDAS}_{[a,b]}(\sigma; \sigma|S) = \sum_{h=1}^{t} \text{IDAS}_{[a,b]}(\sigma_h; \sigma|h|S_h)
\]

\[
= \sum_{h=1}^{t} \text{IDAS}_{[a,b]}(\sigma_h; \sigma|h|S_h) \leq \sum_{h=1}^{t} \left(2 \cdot \frac{b-a}{d_{\text{min}}^h} + 1\right) \cdot \text{OPT}(\sigma_h; \sigma|h|S_h)
\]

\[
\leq \left(2 \cdot \frac{b-a}{d_{\text{min}}^h} + 1\right) \cdot \sum_{h=1}^{t} \text{OPT}(\sigma_h; \sigma|h|S_h)
\]

\[
= \left(2 \cdot \frac{b-a}{d_{\text{min}}^h} + 1\right) \cdot \sum_{h=1}^{t} \text{OPT}(\sigma_h; \sigma|h|S_h) = \left(2 \cdot \frac{b-a}{d_{\text{min}}^h} + 1\right) \cdot \text{OPT}(\sigma|S_h),
\]

where the 3rd equality is due to Claim 7.7, the 1st inequality is due to (7.2), and the 4th equality is due to Claim 5.2, and this completes the proof of the theorem. \( \blacksquare \)

As an immediate consequence, we have the following corollary to Theorem 7.3.

**Corollary 7.1.** \( R(IDAS_{[s_1,s_k]}(s)) \leq 2k - 1 \) for OFAL\(_{eq}(k, \ell)\).

**Proof:** Apply Theorem 7.3 for OFAL\(_{eq}(k, \ell)\) by setting \( a = s_1 \) and \( b = s_k \). Then

\[
2 \cdot \frac{b-a}{d_{\text{min}}} + 1 = 2 \cdot \frac{s_k - s_1}{1} + 1 = 2 \cdot \frac{k-1}{1} + 1 = 2k - 1.
\]

Thus it follows \( R(IDAS_{[s_1,s_k]}(s)) \leq 2k - 1 \) for OFAL\(_{eq}(k, \ell)\). \( \blacksquare \)
8 Concluding Remarks

In this paper, we dealt with the online facility assignment problem OFA($k, \ell$), where $k \geq 1$ is the number of servers and $\ell \geq 1$ is a capacity for each server. As special cases of OFA($k, \ell$), we also dealt with OFA($k, \ell$) on a line, which is denoted by OFAL($k, \ell$) and OFAL$_{eq}$($k, \ell$), where the latter is the case of OFAL($k, \ell$) with equidistant servers.

In Section 3, we introduced the class of MPFS (Most Preferred Free Servers) algorithms and showed that any MPFS algorithm has the capacity-insensitive property (in Corollary 3.1). In Section 4, we formulated the faithful property crucial for the competitive analysis in the paper. In Section 5, we analyzed the competitive ratio of grdy for OFAL$_{eq}$($k, \ell$) and showed that $R$ (grdy) = $4k - 5$ (in Corollary 5.2). In Section 6, we showed that for OFAL$_{eq}$($k, \ell$), $R$(alg) $\geq 2k - 1$ for any alg $\in$ MPFS (in Corollary 6.1). In Section 7, we proposed a new MPFS algorithm idas (Interior Division for Adjacent Servers) for OFAL($k, \ell$) and showed that for OFAL$_{eq}$($k, \ell$), $R$(idas) $\leq 2k - 1$ (in Corollary 7.1), i.e., idas for OFAL$_{eq}$($k, \ell$) is best possible in all of the MPFS algorithms.

Notice that for OFAL$_{eq}$($k, \ell$), any algorithm in MPFS has the capacity-insensitive property and the competitive ratio of idas $\in$ MPFS matches the lower bound of any algorithm in MPFS. This implies that for OFAL$_{eq}$($k, \ell$), there does not exist an algorithm in MPFS with the competitive ratio better than that of idas. Thus for OFAL$_{eq}$($k, \ell$), one of the most interesting problems is to design capacity-insensitive algorithms not in MPFS with the better competitive ratio than that of idas.

References

[1] A.R. Ahmed, M.S. Rahman and S.G. Kobourov, Online facility assignment, Theoretical Computer Science 806, 455-467 (2020).

[2] A. Antoniadis, M. Barcelo, M. Nugent, N. Pruhs, and M. Scquizzato, An o(1)-competitive deterministic algorithm for online matching on a line, in Proc. of Int. Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science 8952, Springer, pp.11-22 (2014).

[3] A. Antoniadis, C. Fischer and A. Tönnis. A collection of lower bounds for online matching on the line, in Proc. of LATIN, Lecture Notes in Computer Science 10807, Springer, pp.52-65 (2018).

[4] I. Ashlagi, Y. Azar, A. Chiplunkar, O. Geri, H. Kaplan, R. Makhijani, Y. Wang, and R. Wattenhofer, Min-cost bipartite perfect matching with delays, in Proc. of Approx/Random, LIPIcs 207, Schloss Dagstuhl, pp.1:1-1:20 (2017).

[5] Y. Azar, A. Chiplunkar, and H. Kaplan, Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays, in Proc. of Symposium on Discrete Algorithms, SIAM-ACM, pp.1051-1061 (2017).

[6] Y. Azar and A. Fanani, Deterministic min-cost matching with delays, in Proc. of Int. Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science 11312, Springer, pp.21-35 (2018).
[7] N. Bansal, N. Buchbinder, A. Gupta, and J. Naor, An $o(\log^2 k)$-competitive algorithm for metric bipartite matching, in Proc. of European Symposium on Algorithms, Lecture Notes in Computer Science 4698, Springer, pp.522-533 (2007).

[8] M. Beinkowski, A. Kraska, H. Liu, and P. Schmidt, A primal-dual online deterministic algorithm for matching with delays, in Proc. of Int. Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science 11312, Springer, pp.51-68 (2018).

[9] M. Beinkowski, A. Kraska, and P. Schmidt, A matching in time saves nine: deterministic online matching with delays, in Proc. of Int. Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science 11312, Springer, pp.132-146 (2017).

[10] Y. Emek, S. Kutten, and R. Wattenhofer, Online matching: haste makes waster!, in Proc. of Symposium on Theory of Computing, ACM, pp.333-344 (2016).

[11] Y. Emek, Y. Shapiro, and Y. Wang, Minimum cost perfect matching with delays for two sources, in Proc. of International Conference on Algorithms and Complexity, Lecture Notes in Computer Science 10236, Springer, pp.209-221 (2017).

[12] B. Fuchs, W. Hochstättler and W. Kern, Online matching on line, Theoretical Computer Science 332, 251-264 (2005).

[13] A. Gupta and K. Levi, The online metric matching problem for doubling metrics, in Proc. of Int. Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science 7391, Springer, pp.424-435 (2012).

[14] J.M. Harris, J.L. Hirst, and M.J. Mossinghoff, Combinatorics and graph theory. 2nd edition, Springer (2008).

[15] T. Itoh, S. Miyazaki, and M. Satake, Competitive analysis for two variants of online metric matching problem, in Proc. of the 14th International Conference on Combinatorial Optimization and Applications, Lecture Notes in Computer Science 12577, Springer, pp.486-498 (2020).

[16] T. Itoh, S. Miyazaki, and M. Satake, Competitive analysis for two variants of online metric matching problem, Discrete Mathematics, algorithms, and Applications 13(6), 2150156:1-2150156:16 (2021).

[17] B. Kalyanasundaram and K. Pruhs, Online weighted matching, J. of algorithms 14(3), 478-488 (1993).

[18] B. Kalyanasundaram and K. Pruhs, The online transportation problem, in Proc. of European Symposium on algorithms, Lecture Notes in Computer Science 979, Springer, pp.484-493 (1995).

[19] B. Kalyanasundaram and K. Pruhs, Online network optimization problems, in Proc. of Online algorithms, Lecture Notes in Computer Science 1442, Springer, pp.268-280 (1998).
[20] R.M. Karp, U.V. Vazirani, and V.V. Vazirani, An optimal algorithm for online bipartite matching, in Proc. of Symposium on Theory of Computing, pp.352-358 (1990).

[21] S. Khuller, S. G. Mitchell and V.V. Vazirani, On-line algorithms for weighted bipartite matching and stable marriages, Theoretical Computer Science 127(2), 255-267 (1994).

[22] E. Koutsoupias and A. Nanavati, The online matching problem on a line, in Proc. of International Workshop on Approximation and Online algorithms, Lecture Notes in Computer Science 3351, Springer, pp.179-191 (2004).

[23] E. Koutsoupias and C.H. Papadimitriou, On the k-server conjecture, Journal of the ACM, 42(5), 971-983 (1995).

[24] X. Liu, Z. Pan, Y. Wang, and R. Wattenhofer, Impatient online matching, in Proc. of Int. Symposium on Algorithms and Computation, LIPIcs 123, Schloss Dagstuhl, pp.162-62:12 (2018).

[25] A. Mehta, Online matching and ad auction, Foundations and Trends in Theoretical Computer Science, 8(4), 265-368 (2012).

[26] A. Meyerson, A. Nanavati, and L.J. Poplawski, Randomized online algorithms for minimum metric bipartite matching, in Proc. of Symposium on Discrete Algorithms, SIAM-ACM, pp.954-959 (2006).

[27] S.A. Muttakee, A.R. Ahmed and M.S. Rahman, New results and bounds on online facility assignment problem, preprint, arXiv:2009.01446v4 (2020).

[28] K. Nayyar and S. Raghvendra, An input sensitive online algorithm for the metric bipartite matching problem, in Proc. of Symposium on Foundations of Computer Science, IEEE, pp.505-515 (2017).

[29] E. Peserico and M. Scquizzato, Matching on the line admits no o(√log n)-competitive algorithm, in Proc. of International Colloquium on Automata, Languages, and Programming, LICIcs 198, Schloss Dagstuhl, pp.103:1-103:3 (2021).

[30] S. Raghvendra, A robust and optimal online algorithm for minimum metric bipartite matching, in Proc. of APPROX/RANDOM, LICIcs 99, Schloss Dagstuhl, pp.18:1-18:16.39 (2016).

[31] S. Raghvendra, Optimal analysis of an online algorithm for the bipartite matching problem on a line, in Proc. of International Symposium on Computational Geometry, LICIcs 99, Schloss Dagstuhl, pp.67:1-67:14 (2018).

[32] M. Satake, Competitive analysis for the Online facility assignment problem on a line, Master Thesis, Graduate School of Informatics, Kyoto University, 2022.

[33] D.D. Sleator and R.E. Tarjan, Amortized efficiency of list update and paging rules, Communications of the ACM 28(2), 202-208 (1985).
A Proof of Claims in Section 5.2

A.1 Proof of Claim 5.1

For the set \( S \) of \( k \) servers and a request sequence \( \sigma = r_1 \cdots r_k \), consider the case that \( \text{GRDY} \) matches \( r^h_j \) with \( s^h_j = s_{\text{grdy}}(r^h_j; \sigma | S) \) for each \( 1 \leq j \leq k_h \). From the definition of \( \text{GRDY} \), it is immediate that just before \( \text{GRDY} \) matches \( r^h_j \) with \( s^h_j \), all of \( s^h_j \) are free and \( s^h_j \) is the nearest to \( r^h_j \) among the free servers \( s^h_1, \ldots, s^h_{k_h} \). This is preserved to the case that just before \( \text{GRDY} \) with the set \( S_h \) of \( k_h \) servers matches \( r^h_j \) with a free server \( s \in S_h \) on a request sequence \( \sigma_h \) as an input. Thus

\[
s_{\text{grdy}}(r^h_j; \sigma | S) = s^h_j = s_{\text{grdy}}(r^h_j; \sigma_h | S_h)
\]

for each \( 1 \leq j \leq k_h \), and this completes the proof of the claim.

A.2 Proof of Claim 5.2

For each \( 1 \leq h \leq t \), we have that

\[
\{ s_{\text{opt}}(r^h_j; \sigma | S) : 1 \leq j \leq k_h \} = S_h = \{ s_{\text{opt}}(r^h_j; \sigma_h | S_h) : 1 \leq j \leq k_h \},
\]

and \( \text{OPT}(\sigma_h; \sigma_h | S_h) \leq \text{OPT}(\sigma_h; \sigma | S) \). Assume that there exists a request sequence \( \sigma_g \) such that \( \text{OPT}(\sigma_g; \sigma_g | S_g) < \text{OPT}(\sigma_g; \sigma | S) \) and we use \( \sigma - \sigma_g \) to denote the request sequence defined by deleting \( \sigma_g \) from \( \sigma \). Then from (A.1), it is immediate that

\[
\text{OPT}(\sigma | S) = \text{OPT}(\sigma; \sigma | S) = \text{OPT}(\sigma_g; \sigma | S) + \text{OPT}(\sigma - \sigma_g; \sigma | S)
\]

\[
> \text{OPT}(\sigma_g; \sigma_g | S_g) + \text{OPT}(\sigma - \sigma_g; \sigma - \sigma_g | (S \setminus S_g)),
\]

and this contradicts the optimality of \( \text{OPT} \) on \( \sigma \). Thus for each \( 1 \leq h \leq t \), it follows that \( \text{OPT}(\sigma_h; \sigma | S) = \text{OPT}(\sigma_h; \sigma_h | S_h) \), and this completes the proof of the claim.

B Proof of Theorem 7.1

For any \( a, b \in \mathbb{R} \) such that \( a < b \), let \( [a, b] \) be the closed interval. Fix \( \rho \in \mathbb{R} \) arbitrarily.

(Reflexivity) For any \( x \in [a, b] \), \( x \leq_{\rho} x \) by Definition 7.1.

(Antisymmetry) For any \( x, y \in [a, b] \) such that \( x \leq y \) (and the case that \( y \leq x \) can be discussed analogously), assume that \( x \leq_{\rho} y \) and \( y \leq_{\rho} x \). Then we have that

\[
x \leq_{\rho} y \quad \rightarrow \quad x = y \text{ or } B(x, y) < \rho;
\]

\[
y \leq_{\rho} x \quad \rightarrow \quad y = x \text{ or } \rho \leq B(x, y).
\]

By the assumption that \( x \leq_{\rho} y \) and \( y \leq_{\rho} x \), the only possible case is \( x = y \).

(Transitivity) For any \( x, y, z \in [a, b] \), assume that \( x \leq_{\rho} y \) and \( y \leq_{\rho} z \). If \( x = y \) or \( y = z \), then it is immediate that \( x \leq_{\rho} z \). We show that \( x \leq_{\rho} z \) for all the other cases.

1. \( x < y < z \): It is immediate that \( B(x, z) < B(y, z) \) by Property 7.1 and we also have that \( B(y, z) < \rho \) by the assumption that \( y \leq_{\rho} z \). Then it follows that \( x \leq_{\rho} z \).

2. \( x < z < y \): It is immediate that \( B(x, z) < B(x, y) \) by Property 7.1 and we also have that \( B(x, y) < \rho \) by the assumption that \( x \leq_{\rho} y \). Then it follows that \( x \leq_{\rho} z \).
(3) $y < x < z$: It is obvious that $\rho \leq B(y, x)$ and $B(y, z) < \rho$ by the assumptions $x \preceq \rho y$ and $y \preceq \rho z$, respectively, and we also have that $B(y, x) < B(y, z)$ by Property 7.1. Then $\rho \leq B(y, z) < B(y, z) < \rho$, which is the contradiction. Thus for $y < x < z$, the assumptions that $x \preceq \rho y$ and $y \preceq \rho z$ do not hold.

(4) $y < z < x$: We have that $\rho \leq B(y, x)$ by the assumption that $x \preceq \rho y$ and $B(y, x) < B(z, x)$ by Property 7.1. Then it follows that $x \preceq \rho z$.

(5) $z < x < y$: It is obvious that $\rho \leq B(z, y)$ and $B(x, y) < \rho$ by the assumptions $y \preceq \rho z$ and $x \preceq \rho y$, respectively, and we also have that $B(z, y) < B(x, y)$ by Property 7.1. Then $\rho \leq B(z, y) < B(x, y) < \rho$, which is the contradiction. Thus for $z < x < y$, the assumptions that $x \preceq \rho y$ and $y \preceq \rho z$ do not hold.

(6) $z < y < x$: We have that $\rho \leq B(z, y)$ by the assumption that $y \preceq \rho z$ and $B(z, y) < B(z, x)$ by Property 7.1. Then it follows that $x \preceq \rho z$.

(Comparability) For any $x, y \in [a, b]$, we show that $x \preceq \rho y$ or $y \preceq \rho x$.

For the case that $x = y$, we have that $x \preceq \rho y$ and $y \preceq \rho x$. Consider the case that $x < y$. From the definition of $\preceq \rho$, it follows that if $\rho \leq B(x, y)$, then $y \preceq \rho x$ and if $B(x, y) < \rho$, then $x \preceq \rho y$. For the case that $y > x$, we can show that $x \preceq \rho y$ or $y \preceq \rho x$ analogously.

C Proof of Claims in Section 7.2

C.1 Proof of Claim 7.1

Since $(t_1, t_j) \in \operatorname{cf}(\tilde{T}_{m-1}^*)$, we have that $t_1 \leq t_{j+1} < t_4 \leq t_j$ in $\tilde{T}_{m-1}^*$. If $t_3 \leq t_{j+1}$ in $\tilde{T}_m$, then it is immediate that $t_3 \leq t_{j+1} < t_4 \leq t_j$, i.e., $(t_3, t_j) \in \operatorname{cf}(\tilde{T}_m)$. If $t_3 > t_{j+1}$ in $\tilde{T}_m$, then from the assumption that $t_1 < t_3 < t_2 < t_4$, it follows that $t_1 \leq t_{j+1} < t_3 < t_2 < t_4 \leq t_j$. This implies that $t_1 \leq t_{j+1} < t_2 < t_j$ in $\tilde{T}_m$, i.e., $(t_1, t_j) \in \operatorname{cf}(\tilde{T}_m)$.

C.2 Proof of Claim 7.2

Since $(t_i, t_j) \in \operatorname{cf}^*(\tilde{T}_{m-1}^*)$ such that $i \neq 1$, we have that $t_i \leq t_{j+1} < t_{i+1} \leq t_j$, which is preserved in $\tilde{T}_m$. Thus it follows that $c(t_i, t_j|\tilde{T}_{m-1}^*) = c(t_i, t_j|\tilde{T}_m)$.

C.3 Proof of Claim 7.3

For a conflicting pair $(t_1, t_j) \in \operatorname{cf}^3(\tilde{T}_{m-1}^*)$, it follows that $(t_3, t_j) \in \operatorname{cf}(\tilde{T}_m)$ and $(t_1, t_j) \not\in \operatorname{cf}(\tilde{T}_m)$. Recall that $t_1 < t_3 < t_2 < t_4$ for the detour $D$ in $\tilde{T}_m$ and $t_3 \leq t_{j+1} < t_4 \leq t_j$ for the conflicting pair $(t_3, t_j) \in \operatorname{cf}(\tilde{T}_m)$. Thus we have that $t_1 < t_3 < t_{j+1} < t_4 \leq t_j$. Since $(t_1, t_j) \not\in \operatorname{cf}(\tilde{T}_m)$, we have that $t_2 \leq t_{j+1}$, i.e., $t_1 < t_3 < t_2 < t_{j+1} < t_4 \leq t_j$. This implies that $t_1 < t_{j+1} < t_4 \leq t_j$ and $t_3 < t_{j+1} < t_4 \leq t_j$.

If $t_4 = t_j$, then we have that $c(t_1, t_4|\tilde{T}_{m-1}^*) = t_4 - a = c(t_3, t_4|\tilde{T}_m)$, and if $t_4 < t_j$, then we have that $c(t_1, t_j|\tilde{T}_{m-1}^*) = b - a = c(t_3, t_j|\tilde{T}_m)$.

C.4 Proof of Claim 7.4

Since $(t_1, t_j) \in \operatorname{cf}^1(\tilde{T}_{m-1}^*)$, we have that $t_1 \leq t_{j+1} < t_4 \leq t_j$ in $\tilde{T}_{m-1}^*$ and that $t_1 \leq t_{j+1} < t_2 \leq t_j$ in $\tilde{T}_m$. Let us consider the following two cases: $t_1 < t_{j+1}$ and $t_1 = t_{j+1}$.
If \( t_1 < t_{j+1} \), then we have that \( c(t_1, t_j | \tilde{T}^{*}_{m-1}) \leq b - a = c(t_1, t_j | \tilde{T}_m) \). If \( t_1 = t_{j+1} \), then we consider the following cases: \( t_4 = t_j \) and \( t_4 < t_j \), and we have that

\[
c(t_1, t_j | \tilde{T}^{*}_{m-1}) = \begin{cases} 
    t_j - t_1 & \text{if } t_4 = t_j; \\
    b - t_1 & \text{if } t_4 < t_j.
\end{cases}
\]

Thus it follows that \( c(t_1, t_j | \tilde{T}^{*}_{m-1}) \leq b - t_1 = c(t_1, t_j | \tilde{T}_m) \).

C.5 Proof of Claim 7.5

For each \( 1 \leq i \leq \mu \), assume that \( cs^{+}_{\text{idas}}(a_i; \sigma) = \{(a_i, r_{i_1}), \ldots, (a_i, r_{i_u})\} \), where \( i_1, \ldots, i_u \) are ordered in such a way that

\[
s_{\text{opt}}(a_i; \sigma|S) \leq s_{\text{idas}}(r_{i_u}; \sigma|S) < \cdots < s_{\text{idas}}(r_{i_2}; \sigma|S) < s_{\text{idas}}(r_{i_1}; \sigma|S) < s_{\text{idas}}(a_i; \sigma|S).
\]

Since \( a_i \) is the earliest request among \( a_i, r_{i_1}, \ldots, r_{i_u} \) by the definition of \( cs^{+}_{\text{idas}}(\sigma) \), we have that \( s_{\text{idas}}(r_{i_1}; \sigma|S) \) and \( s_{\text{idas}}(a_i; \sigma|S) \) are free just before \( a_i \) arrives. Let us consider the case that \( s_{\text{opt}}(a_i; \sigma|S) < s_{\text{idas}}(r_{i_u}; \sigma|S) \). Then we have that

\[
\text{OPT}(a_i; \sigma|S) = a_i - s_{\text{opt}}(a_i; \sigma|S) \geq s_{\text{idas}}(r_{i_1}; \sigma|S) - s_{\text{opt}}(a_i; \sigma|S)
= s_{\text{idas}}(r_{i_u}; \sigma|S) - s_{\text{opt}}(a_i; \sigma|S) + \sum_{s=1}^{u-1} \{s_{\text{idas}}(r_s; \sigma|S) - s_{\text{idas}}(r_{s+1}; \sigma|S)\}
\geq d_{\text{min}} + \sum_{s=1}^{u-1} d_{\text{min}} = u \cdot d_{\text{min}} = \frac{d_{\text{min}}}{b-a+d_{\text{min}}} \cdot u \cdot (b-a+d_{\text{min}})
\geq \frac{d_{\text{min}}}{b-a+d_{\text{min}}} \cdot \sum_{(a_i,r) \in cs^{+}_{\text{idas}}(a_i;\sigma)} c \left( f^{-1}_{bij}(a_i, r)|T_{\sigma} \right).
\]

We turn to consider the case that \( s_{\text{opt}}(a_i; \sigma|S) = s_{\text{idas}}(r_{i_u}; \sigma|S) \). Then it is immediate that \( c(s_{\text{opt}}(a_i; \sigma|S), s_{\text{opt}}(r_{i_u}; \sigma|S)|T_{\sigma}) \leq b - s_{\text{opt}}(a_i; \sigma|S) \). Note that

\[
\sum_{(a_i,r) \in cs^{+}_{\text{idas}}(a_i;\sigma)} c \left( f^{-1}_{bij}(a_i, r)|T_{\sigma} \right) = \sum_{s=1}^{u} c \left( f^{-1}_{bij}(a_i, r)|T_{\sigma} \right)
\leq b - s_{\text{opt}}(a_i; \sigma|S) + \sum_{s=1}^{u-1} c \left( f^{-1}_{bij}(a_i, r_s)|T_{\sigma} \right)
\leq b - s_{\text{opt}}(a_i; \sigma|S) + \sum_{s=1}^{u-1} (b-a)
= b - s_{\text{opt}}(a_i; \sigma|S) + (u-1) \cdot (b-a).
\]

From the definition of \( \text{IDAS}_{[a,b]} \), it is immediate that

\[
s_{\text{idas}}(r_{i_1}; \sigma|S) \leq B(s_{\text{idas}}(r_{i_1}; \sigma|S), s_{\text{idas}}(a_i; \sigma|S)) \leq a_i. \quad (C.1)
\]
Then \( \text{OPT}(a_i; \sigma|S) \) can be estimated as follows:

\[
\text{OPT}(a_i; \sigma|S) = a_i - s_{\text{opt}}(a_i; \sigma|S) \geq B(s_{\text{idas}}(r_{i_1}; \sigma|S), s_{\text{idas}}(a_i; \sigma|S)) - s_{\text{opt}}(a_i; \sigma|S) \\
= \frac{\{b - s_{\text{idas}}(r_{i_1}; \sigma|S)\} s_{\text{idas}}(a_i; \sigma|S) + \{s_{\text{idas}}(a_i; \sigma|S) - a\} s_{\text{idas}}(r_{i_1}; \sigma|S)}{b - a + s_{\text{idas}}(a_i; \sigma|S) - s_{\text{idas}}(r_{i_1}; \sigma|S)} - s_{\text{opt}}(a_i; \sigma|S) \\
\geq \frac{\{b - s_{\text{idas}}(r_{i_1}; \sigma|S)\} d_{\min}}{b - a + d_{\min}} + s_{\text{idas}}(r_{i_1}; \sigma|S) - s_{\text{opt}}(a_i; \sigma|S) \\
= \frac{d_{\min}}{b - a + d_{\min}} \left\{ b - s_{\text{idas}}(r_{i_1}; \sigma|S) + \frac{b - a + d_{\min}}{d_{\min}} (s_{\text{idas}}(r_{i_1}; \sigma|S) - s_{\text{opt}}(a_i; \sigma|S)) \right\} \\
\geq \frac{d_{\min}}{b - a + d_{\min}} \left\{ b - s_{\text{opt}}(a_i; \sigma|S) + \frac{b - a}{d_{\min}} (s_{\text{idas}}(r_{i_1}; \sigma|S) - s_{\text{opt}}(a_i; \sigma|S)) \right\} \\
\geq \frac{d_{\min}}{b - a + d_{\min}} \sum_{(a_i, r) \in \mathcal{C}_{\text{idas}}^+(a_i, r)} c \left( f^{-1}_{ij}(a_i, r)|T_\sigma \right),
\]

where the 1st inequality follows from (C.1).

### C.6 Proof of Claim 7.6

For each \( 1 \leq j \leq \nu \), we assume that \( \mathcal{C}_{\text{idas}}^+(b_j; \sigma) = \{(r_{j_1}, b_j), \ldots, (r_{j_\nu}, b_j)\} \), where \( j_1, \ldots, j_\nu \) are ordered in such a way that

\[
s_{\text{idas}}(b_j; \sigma|S) < s_{\text{idas}}(r_{j_1}; \sigma|S) < s_{\text{idas}}(r_{j_2}; \sigma|S) < \cdots s_{\text{idas}}(r_{j_\nu}; \sigma|S) \leq s_{\text{opt}}(b_j; \sigma|S).
\]

The rest of the proof can be shown in a way similar to the proof of Claim 7.5.

### C.7 Proof of Claim 7.7

For the set \( S \) of \( k \) servers and a request sequence \( \sigma = r_1 \cdots r_k \), consider the case that \( \text{IDAS}_{[a,b]} \) matches \( r_j^h \) with \( s_j^h = s_{\text{idas}}(r_j^h; \sigma|S) \) for each \( 1 \leq j \leq k_h \). From the definition of \( \text{IDAS}_{[a,b]} \), we have that just before \( \text{IDAS} \) matches \( r_j^h \) with \( s_j^h \), all of \( s_{j_1}^h, \ldots, s_{j_{k_h}}^h \) are free and \( s_j^h \) has the highest priority w.r.t. \( \preceq r_j^h \) among the free servers \( s_{j_1}^h, \ldots, s_{j_{k_h}}^h \). This is preserved to the case that just before \( \text{IDAS}_{[a,b]} \) with the set \( S_h \) of \( k_h \) servers matches \( r_j^h \) with a free server \( s \in S_h \) on a request sequence \( \sigma_h \) as an input. Thus

\[
s_{\text{idas}}(r_j^h; \sigma|S) = s_j^h = s_{\text{idas}}(r_j^h; \sigma_h|S_h)
\]

for each \( 1 \leq j \leq k_h \), and this completes the proof of the claim.