CLEVER or SMART: Strategies for the Online Target Date Assignment Problem

Elisabeth Gassner* Johannes Hatzl* Sven O. Krumke† Sleman Saliba†

May 29, 2008

Abstract

In this paper, we consider the Online Target Date Assignment Problem (OnlineTDAP) with deferral time one and unsplittable requests for general downstream problems, where the downstream cost are nonnegative, additive and satisfy the triangle inequality.

We show that the lower bound on the competitive ratio of any online algorithm for this problem is greater than $3/2 - \varepsilon$. The first online algorithm analyzed is SMART, which was introduced by Angelelli et al. [3]. We prove that its competitive ratio is at most $2\sqrt{2} - 1 \approx 1.8284$ for this setting. This result answers the question posed in Angelelli et al. [4], if SMART has a competitive ratio strictly less than 2 for the Dynamic Multi-Period Routing Problem (Dmprp) with customers located on the Euclidean plane, provided splitting of request sets is prohibited.

Finally, we present the online algorithm CLEVER and show that this strategy is asymptotically optimal with a competitive ratio of 3/2.

1 Introduction

Online optimization problems have seen an increasing attention over the last years, since the issue of planning with incomplete knowledge and making decisions without detailed knowledge about the future grows permanently. A detailed description of the state of the art and an extensive bibliography can be found in [1, 2, 5, 6]. A general framework for online problems with a two stage decision process structure is the *Online Target Date Assignment Problem*. It was introduced by Heinz et al. [7]. Requests are released at certain dates. In the first stage an online algorithm has to assign a target date for the request at which the request will be processed. This decision must be made immediately upon release and is irrevocable. In the second stage an offline optimization problem for all requests assigned to a certain target date is solved. This offline problem is called the *downstream problem*. We assume that the downstream problem can be solved to optimality and the cost at a target date are the optimal cost (*downstream cost*) of the downstream problem solved for all requests assigned to this date. The downstream problem can be any classical optimization problem like, e.g., a routing, bin packing or machine scheduling problem.

An instance of this online problem consists of a sequence of requests $\sigma = (r_0, r_1, r_2, \ldots, r_m)$ and a *downstream problem* Π, which is an offline optimization problem having arbitrary subsets of σ as feasible inputs. The planning horizon is divided into $m+1$ time periods $t_0, t_1, \ldots, t_m$ with $t_i \neq t_j$ for all $i \neq j$. At the beginning of each period t_i $(i = 0, \ldots, m)$ a request r_i is released. The decision maker must immediately and irrevocably assign the request to a target date in time

*Graz University of Technology, Institute of Optimization and Discrete Mathematics, Steyrergasse 30, Graz, Austria. {gassner,hatzl}@opt.math.tu-graz.ac.at

†University of Kaiserslautern, Department of Mathematics, P.O.Box 3049, Paul-Ehrlich-Str. 14, 67653 Kaiserslautern, Germany. {krumke,saliba}@mathematik.uni-kl.de

*This research is partially supported by the Austrian Science Fund Project P18918-N18 *Efficiently solvable variants of location problems.*

period $t_i, \ldots, t_{i+\delta(r_i)}$, where $\delta(r_i)$ is the allowed deferral time of request r_i. In the general setting of an Online Target Date Assignment Problem the allowed deferral time becomes known at the request's release. In a feasible solution with respect to a downstream problem Π each request is allocated a feasible target date and for each target date, the instance of the corresponding offline problem Π is feasible as well. For a detailed description of the Online Target Date Assignment Problem and competitive online algorithms we refer the reader to [7].

Angelelli et al. [3, 4] introduce the Dynamic Multi-Period Routing Problem (DMPRP), where at the beginning of each time period a set of customers located on the Euclidean plane becomes known. The customers have to be served either in the current or in the next time period. The DMPRP is a special case of the Online Target Date Assignment Problem, where the underlying downstream problem is a *traveling salesman problem* and a request r_i is a set of customers. For a detailed description of the problem, lower bounds and online strategies we refer to [3, 4].

We investigate a special case of the Online Target Date Assignment Problem, where at the beginning of each time period one request is released that can either be served immediately or postponed to the next time period, that means requests are allowed to be delayed for at most one period, i.e., $\delta(r_i) = 1$ for all $i = 0, \ldots, m$. Note that in this model at most two requests are served in one time period. Moreover, in the model considered in this paper it is not possible to split a request. For example, if in the DMPRP a request r_i contains several customers, all of them have to be served in the time period t_i or all of them are postponed.

The analysis is not only valid for a special downstream problem, but for a wide class of optimization problems with a natural structure. Let us denote by L_i the downstream cost if request r_i is served on its own either in time period t_i or t_{i+1} and by $L_{i,i+1}$ the minimum cost of serving both requests r_i and r_{i+1} in period t_{i+1}. Then, we only require that

$$L_i \geq 0, \qquad \text{(nonnegativity)} \tag{1}$$

$$L_{i,i+1} \leq L_i + L_{i+1}, \qquad \text{(triangle inequality)} \tag{2}$$

$$L_i \leq L_{i,i+1} \text{ and } L_{i+1} \leq L_{i,i+1} \qquad \text{(monotonicity).} \tag{3}$$

The objective is to minimize the total sum of the downstream cost. From now on, we denote the Online Target Date Assignment Problem with deferral time one, unsplittable requests and downstream problems, where the downstream cost satisfy Conditions (1)—(3) by ONLINETDAP.

An instance of ONLINETDAP consists of a finite sequence $\sigma = (r_0, r_1, r_2, \ldots, r_m)$ of requests and a solution of ONLINETDAP is given by a sequence $X(\sigma) = (X_0, X_1, \ldots, X_m)$, where X_i is either D or I (i.e., $X_i \in \{D, I\}$). If $X_i = D$ request r_i is delayed to time period t_{i+1} and if $X_i = I$ it is processed immediately in period t_i. Throughout the paper, the (offline) optimal objective value of an instance σ is denoted by $\text{OPT}(\sigma)$. Let ALG be an (online) algorithm for this problem and $X^{\text{ALG}}(\sigma) = (X_0^{\text{ALG}}, X_1^{\text{ALG}}, \ldots, X_m^{\text{ALG}})$ the obtained solution for instance σ with objective function value $z(\text{ALG}(\sigma))$. Then, the competitive ratio of an online algorithm ALG is given by

$$r(\text{ALG}) := \sup_{\sigma} \frac{z(\text{ALG}(\sigma))}{\text{OPT}(\sigma)}. \tag{4}$$

It is known that the competitive ratio of any algorithm for the Online Target Date Assignment Problem where splitting is allowed is at least 3/2 (see [3]). Our contribution is to show that if splitting is not allowed in an optimal solution the competitive ratio has the same lower bound for any online algorithm for ONLINETDAP.

Moreover, we study two online algorithms and analyze their competitive ratios. The first online algorithm is called SMART and was suggested for the DMPRP in [3]. Therein, it was shown that this algorithm is optimal for two time periods and the question was posed if SMART has a competitive ratio strictly less than two for arbitrary time horizons. We answer this question and prove that $r(\text{SMART}) < 2$ provided requests are unsplittable.

The second analyzed online algorithm is called CLEVER and is introduced for the first time in this paper. We prove that the competitive ratio of CLEVER applied to instances of the

OnlineTDAP is 3/2. Thus, this algorithm is optimal.

2 A lower bound on the competitive ratio

In this section, it is shown that there cannot exist an online algorithm Alg for OnlineTDAP with a competitive ratio strictly better than 3/2. Note that this bound also holds for the Dmprp where it is allowed to split requests (see [3]).

Theorem 2.1. *Let* Alg *be any online algorithm for* OnlineTDAP*. Then, for all* $\varepsilon > 0$ *there exists an* $m(\varepsilon)$ *such that*

$$r(\textsc{Alg}) > \frac{3}{2} - \varepsilon.$$

Proof. We consider an instance of the OnlineTDAP with an input sequence $\sigma = (r_0, r_1, \ldots, r_m)$, where $m = 2n+1$ is an odd integer. The corresponding downstream problem is a routing problem on the real line, that means at each target date a server located in the origin is required to visit all requests assigned to this target and return to the origin afterwards. The distance traveled thereby comply with the downstream cost. The objective is to minimize the sum of the downstream cost, i.e., the distance traveled over all target dates. In this instance, we assume that a request consists of a single customer. Thus, it suffices to give a real number for each request representing the coordinate of the corresponding customer.

The first request is released at distance a_0 from the origin, i.e., $r_0 = a_0$ and the second one at distance $a_1 > a_0$. The online algorithm has two choices: it can serve r_1 either immediately together with r_0 in time period t_1 or delay it to the next time period. First consider an algorithm $\textsc{Alg}'$ that decides to serve r_1 immediately. Then, the adversary issues another request with the same distance and stops afterwards, i.e., $r_2 = a_1$ and all forthcoming requests will be released at the origin. In this case, the cost of $\textsc{Alg}'$ are $4a_1$, whereas the optimal cost are $2a_0 + 2a_1$ yielding

$$r(\textsc{Alg}') \geq \frac{2a_1}{a_0 + a_1}.$$

If we denote by $\textsc{Alg}''$ an algorithm that decides to delay r_1, the adversary issues request r_2 at distance $a_2 > a_1$ from the origin and the algorithm has again to decide if r_2 is processed immediately with r_1 or delayed to time period t_3. If $\textsc{Alg}''$ serves r_2 in time period t_2 then assume that $r_3 = r_2$ and the remaining requests $r_4, \ldots, r_m$ are again located at the origin. Thus, the cost obtained by $\textsc{Alg}''$ are $2a_0 + 4a_2$ whereas the optimal cost are given by $2a_2 + 2a_1$ and

$$r(\textsc{Alg}'') \geq \frac{2a_2 + a_0}{a_2 + a_1}$$

can be concluded. This process continues till either the online algorithm Alg chooses to serve a request r_i for some $1 \leq i \leq m-1$ immediately or the time period m is reached. In the first case, r_{i+1} is set to r_i and $r_{i+2} = \ldots = r_m = 0$ and we obtain the answer set $X^{\textsc{Alg}}(\sigma) = \{D, D, D, \ldots, D, I, D\}$ with cost

$$z(\textsc{Alg}(\sigma)) = 4a_i + 2\sum_{j=0}^{i-2} a_j.$$

If $i = 2k-1$ then the optimal solution is $X^*(\sigma) = \{D, D, I, D, \ldots, I, D, I\}$ and if $i = 2k$ then $X^*(\sigma) = \{D, I, D, I, \ldots, D, I, D, I\}$ with

$$\textsc{Opt}(\sigma) = \begin{cases} 2a_{2k-1} + 2a_{2k-2} + 2\sum_{j=0}^{k-2} a_{2j} & \text{if } i = 2k-1 \\ 2a_{2k} + 2a_{2k-1} + 2\sum_{j=1}^{k-1} a_{2j-1} & \text{if } i = 2k. \end{cases}$$

In the latter, the end of the time horizon is reached and the online algorithm has always delayed, i.e., $X^{\textsc{Alg}}(\sigma) = \{D, D, \ldots, D\}$. Then r_m is set to 0 and the optimal strategy is $X^*(\sigma) = \{D, D, I, D, \ldots, D, I\}$. In this situation the cost sum up to

$$z(\textsc{Alg}(\sigma)) = 2\sum_{j=0}^{2n} a_j \quad \text{and} \quad \textsc{Opt}(\sigma) = 2\sum_{j=0}^{n} a_{2j}.$$

Combining all these results implies that

$$r(\textsc{Alg}) \geq \min_{1\leq k\leq n} \left\{ \frac{2a_{2k} + \sum_{j=0}^{2k-2} a_j}{a_{2k} + a_{2k-1} + \sum_{j=1}^{k-1} a_{2j-1}}, \frac{2a_{2k-1} + \sum_{j=0}^{2k-3} a_j}{a_{2k-1} + a_{2k-2} + \sum_{j=0}^{k-2} a_{2j}}, \frac{\sum_{j=0}^{2n} a_j}{\sum_{j=0}^{n} a_{2j}} \right\} \tag{5}$$

holds for any online algorithm $\textsc{Alg}$. If we choose the numbers by the following recursion

$$a_0 := 1 \qquad a_n := 2a_{n-1} + F_n \quad \forall n \geq 1,$$

where

$$F_0 := 0,\ F_1 := 1 \qquad F_{n+2} = F_{n+1} + F_n \quad \forall n \geq 2$$

is the Fibonacci-sequence, then the first two terms in (5) are equal to 3/2 for all $1 \leq k \leq n$ and

$$\lim_{n\to\infty} \frac{\sum_{j=0}^{2n} a_j}{\sum_{j=0}^{n} a_{2j}} = \frac{3}{2}.$$

□

3 Instance Splitting

To find the competitive ratio $r(\textsc{Alg})$ of an algorithm $\textsc{Alg}$ it is necessary to give an instance σ that maximizes the ratio $\frac{z(\textsc{Alg}(\sigma))}{\textsc{Opt}(\sigma)}$. Along the lines of the ideas in [4], we may restrict ourselves to instances with a very special structure. It is sufficient to consider unsplittable instances of the following two types:

- Type 1: $X^{\textsc{Alg}}(\sigma) = (D, D, I, D, I, \ldots)$ and $X^*(\sigma) = (D, I, D, I, D, \ldots)$
- Type 2: $X^{\textsc{Alg}}(\sigma) = (D, I, D, I, D, \ldots)$ and $X^*(\sigma) = (D, D, I, D, I, \ldots)$.

Let

$$\mathcal{S}(i, m) = \{\sigma \mid \sigma \text{ is of Type } i, \text{ with } m+1 \text{ requests}\}$$

denote the set of all sequences that are of Type 1 or 2 and have $m+1$ requests. For a given online algorithm $\textsc{Alg}$ we write

$$A(m) = \sup_{\sigma\in\mathcal{S}(1,m)} \frac{z(\textsc{Alg}(\sigma))}{\textsc{Opt}(\sigma)},$$

$$B(m) = \sup_{\sigma\in\mathcal{S}(2,m)} \frac{z(\textsc{Alg}(\sigma))}{\textsc{Opt}(\sigma)}.$$

The following lemma gives some monotonicity properties about $A(j)$ and $B(j)$.

Lemma 3.1. *For all $n \in \mathbb{N}$ the following inequalities are satisfied:*

$$A(2n) \leq A(2n-1) \text{ and } B(2n+1) \leq B(2n).$$

Proof. We only show that $A(2n) \leq A(2n-1)$, because the other case can be done in an analogous way. Let $\sigma \in \mathcal{S}(1,2n)$ then deleting the last request of σ leads to a new instance $\sigma' \in \mathcal{S}(1,2n-1)$ such that

$$\frac{z(\text{ALG}(\sigma))}{\text{OPT}(\sigma)} \leq \frac{z(\text{ALG}(\sigma'))}{\text{OPT}(\sigma')}.$$

□

Using these results the following statement follows easily.

Corollary 3.2. *Let σ be an instance with $m+1$ requests. Then, the competitive ratio of an online algorithm* ALG *is given by*

$$r(\text{ALG}) = \max\left\{ A\left(2\left\lfloor\frac{j-1}{2}\right\rfloor + 1\right), B\left(2\left\lfloor\frac{j}{2}\right\rfloor\right) : j = 1, \ldots, m \right\}.$$

4 Analysis of the online algorithm SMART

In this section, we discuss the online algorithm SMART(q) which depends on a parameter $q \geq 0$. Formally, this algorithm can be described as follows:

> **Algorithm** SMART(q)
> At the beginning of each time period $t_i (i = 0 \ldots, m)$, do the following: If request r_{i-1} was postponed to period t_i and $L_{i-1,i} \leq qL_{i-1}$, then process the requests r_{i-1} and r_i together, otherwise postpone r_i.

Note that it has already been shown in [3] that if $q \leq 1$, i.e., the requests are always postponed, the competitive ratio of SMART(q) is 2. Thus, we restrict our analysis to the case where $q > 1$. The aim of this section is to give the competitive ratio of SMART(q) for any $q > 1$. This enables us to compute an optimal value for q, i.e., the value which minimizes the competitive ratio. Using this value it can be shown that SMART has a better competitive ratio than 2. In fact, we prove the following theorem for ONLINETDAP.

Theorem 4.1. *Let $m \in \mathbb{N}$ and $\sigma = (r_0, r_1, r_2, \ldots, r_m)$ an instance of the* ONLINETDAP. *Then,*

$$r(\text{SMART}(q)) = \begin{cases} \frac{q^2 - q + 2 - 2q(q-1)^{\left\lfloor\frac{m-1}{2}\right\rfloor + 1}}{q - q(q-1)^{\left\lfloor\frac{m-1}{2}\right\rfloor + 1}} & \text{if } 1 < q < 2, \\ \frac{3 + 4\left\lfloor\frac{m-1}{2}\right\rfloor}{2 + 2\left\lfloor\frac{m-1}{2}\right\rfloor} & \text{if } q = 2. \end{cases}$$

Moreover,

$$r(\text{SMART}(\sqrt{2})) \leq 2\sqrt{2} - 1 < 2 \qquad \text{and} \qquad \lim_{m\to\infty} r(\text{SMART}(\sqrt{2})) = 2\sqrt{2} - 1$$

holds.

4.1 A lower bound on the competitive ratio of SMART

We consider an instance of the ONLINETDAP with an input sequence $\sigma = (r_0, r_1, \ldots, r_m)$. The corresponding downstream problem is a routing problem on the real line as in the proof of Theorem 2.1. Let us consider the following instance where m is odd and $\varepsilon > 0$ (see Figure 1):

$$r_0 = 1 - \varepsilon, \quad r_1 = q \quad \text{and } r_{2k} = r_{2k+1} = (-1)^k q(q-1)^k \text{ for } k = 1, \ldots, \left\lfloor\frac{m}{2}\right\rfloor. \tag{6}$$

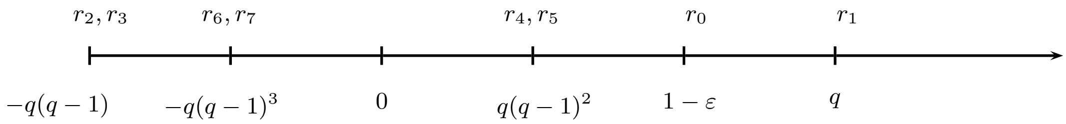

Figure 1: Request locations for the example using SMART(q)

Observe that $2q = L_{0,1} > qL_0 = 2q(1-\varepsilon)$ and therefore r_1 is delayed by $\textsc{smart}(q)$. In general,

$$L_{2k-1,2k} = L_{2k-1} + L_{2k} = 2q(q-1)^{k-1} + 2q(q-1)^k = 2q^2(q-1)^{k-1} \quad \text{for } k = 1, \ldots, \left\lfloor \frac{m}{2} \right\rfloor$$

holds. Thus, the solution obtained by $\textsc{smart}(q)$ is $X^{\textsc{smart}(q)}(\sigma) = (D, D, I, D, I, \ldots)$ and the corresponding cost are

$$\begin{aligned} z(\textsc{smart}(q)(\sigma)) &= L_0 + L_{1,2} + L_{3,4} + \ldots + L_{m-2,m-1} + L_m \\ &= \begin{cases} \frac{2}{2-q}\left(2 - q + q^2 - 2q(q-1)^{\lfloor \frac{m-1}{2} \rfloor + 1}\right) - 2\varepsilon & \text{if } 1 < q, q \neq 2, \\ 6 + 8\left\lfloor \frac{m-1}{n} \right\rfloor - 2\varepsilon & \text{if } q = 2. \end{cases} \end{aligned}$$

On the other hand, an offline optimal solution will accept r_1 immediately and perform r_{2k} and r_{2k+1} for $k = 1, \ldots, \lfloor \frac{m}{2} \rfloor$ together, which leads to $X^*(\sigma) = (D, I, D, I, \ldots)$ and the optimal objective function value is given by

$$\begin{aligned} \textsc{Opt}(\sigma) &= L_{0,1} + L_{2,3} + \ldots + L_{m-1,m} \\ &= \begin{cases} \frac{2q}{2-q}\left(1 - (q-1)^{\lfloor \frac{m-1}{2} \rfloor + 1}\right) & \text{if } 1 < q, q \neq 2, \\ 4 + 4\left\lfloor \frac{m-1}{2} \right\rfloor & \text{if } q = 2. \end{cases} \end{aligned}$$

Thus,

$$\frac{z(\textsc{smart}(q)(\sigma))}{\textsc{Opt}(\sigma)} = \begin{cases} \dfrac{\left(2-q+q^2-2q(q-1)^{\lfloor \frac{m-1}{2} \rfloor + 1}\right) - (2-q)\varepsilon}{\left(1-(q-1)^{\lfloor \frac{m-1}{2} \rfloor + 1}\right)} & \text{if } 1 < q, q \neq 2, \\ \dfrac{3+4\lfloor \frac{m-1}{2} \rfloor - \varepsilon}{2+2\lfloor \frac{m-1}{2} \rfloor} & \text{if } q = 2. \end{cases} \tag{7}$$

In a similar way it is also possible to construct an instance $\sigma = (r_0, r_1, \ldots, r_m)$ where m is even which leads to the ratio given on the right hand side of equation (7). Using $\varepsilon \to 0$

$$r(\textsc{smart}(q)) \geq \begin{cases} \dfrac{q^2-q+2-2q(q-1)^{\lfloor \frac{m-1}{2} \rfloor + 1}}{q-q(q-1)^{\lfloor \frac{m-1}{2} \rfloor + 1}} & \text{if } 1 < q, q \neq 2, \\ \dfrac{3+4\lfloor \frac{m-1}{2} \rfloor}{2+2\lfloor \frac{m-1}{2} \rfloor} & \text{if } q = 2, \end{cases}$$

follows. It will be shown later (see inequality (16)) that the right hand side of this inequality is exactly the competitive ratio of $\textsc{smart}(q)$. Since the lower bound is monotonically increasing for all $q \geq 2$, the analysis is restricted to the case $1 < q \leq 2$.

4.2 Competitive analysis

In this subsection, we determine the competitve ratio of the algorithm $\textsc{smart}(q)$ for $1 < q \leq 2$. According to Corollary 3.2 it suffices to bound $A(2n+1)$ and $B(2n)$.

Case 1: $A(2n+1)$

Assume that we are given an unsplittable sequence σ of Type 1 with $m = 2n+1$, i. e.,

$$z(X^{\text{SMART}(q)}(\sigma)) = L_0 + L_{1,2} + L_{3,4} \ldots + L_{2n-1,2n} + L_{2n+1}$$
$$\text{OPT}(\sigma) = L_{0,1} + L_{2,3} + \ldots + L_{2n,2n+1}.$$

Since SMART (q) produces $X^{\text{SMART}(q)}(\sigma)$ we know that request r_1 is delayed because

$$L_{0,1} > qL_0 \tag{8}$$

and requests r_{2j-1} and r_{2j} for $j = 1, \ldots, n$ are performed together because

$$L_{2j-1,2j} \leq qL_{2j-1} \quad \forall j = 1, \ldots, n. \tag{9}$$

At the beginning the special case where $n = 0$ is investigated. Straightforward calculations lead to

$$A(1) \leq \frac{L_0 + L_1}{L_{0,1}} < \frac{\frac{1}{q}L_{0,1} + L_1}{L_{0,1}} \leq \frac{\frac{1}{q}L_{0,1} + L_{0,1}}{L_{0,1}} = \frac{1}{q} + 1.$$

Observe that $\frac{1}{q} + 1$ is equal to $\frac{q^2-q+2-2q(q-1)^1}{q-q(q-1)^1}$ and hence corresponds to the bound stated in Theorem 4.1 for the special case of $m = 1$ (and therefore $n = 0$).

In a next step, we consider sequences of arbitrary length $m = 2n+1$ for $n \geq 1$. In order to find a bound on $r(\text{SMART}\,(q))$ we consider a sequence $\sigma \in \mathcal{S}(1, 2n+1)$ and bound $z(X^{\text{SMART}(q)}(\sigma))$. Using (8) and (9) yields several different bounds on $z(X^{\text{SMART}(q)}(\sigma))$: For $k = 1, \ldots, n$ we have

$$\begin{aligned}
z(X^{\text{SMART}(q)}(\sigma)) = L_0 + L_{1,2} + L_{3,4} \ldots + L_{2n-1,2n} + L_{2n+1} <& \\
\frac{1}{q}L_{01} + \sum_{j=1}^{k} L_{2j-1,2j} + \sum_{j=k+1}^{n} L_{2j-1,2j} + L_{2n+1} \leq& \\
\frac{1}{q}L_{0,1} + qL_1 + \sum_{j=2}^{k} qL_{2j-1} + \sum_{j=k+1}^{n} (L_{2j-1} + L_{2j}) + L_{2n+1} \leq& \\
\frac{1}{q}L_{0,1} + qL_{0,1} + q\sum_{j=2}^{k} L_{2j-1} + L_{2k+1} + \sum_{j=k+1}^{n-1} (L_{2j} + L_{2j+1}) + L_{2n} + L_{2n+1} \leq& \\
\left(q + \frac{1}{q}\right) L_{0,1} + q\sum_{j=2}^{k} L_{2j-2,2j-1} + L_{2k,2k+1} + 2\sum_{j=k+1}^{n} L_{2j,2j+1} =& \\
\left(q + \frac{1}{q}\right) L_{0,1} + q\sum_{j=1}^{k-1} L_{2j,2j+1} + L_{2k,2k+1} + 2\sum_{j=k+1}^{n} L_{2j,2j+1} =:& \bar{f}_k(\sigma).
\end{aligned}$$

For the special case of $k = 0$ we have

$$\begin{aligned}
L_0 + L_{1,2} + L_{3,4} \ldots + L_{2n-1,2n} + L_{2n+1} < \frac{1}{q}L_{0,1} + \sum_{j=1}^{n}(L_{2j-1} + L_{2j}) + L_{2n+1} \leq& \\
\frac{1}{q}L_{0,1} + L_{0,1} + 2\sum_{j=1}^{n} L_{2j,2j+1} = \left(1 + \frac{1}{q}\right) L_{01} + 2\sum_{j=1}^{n} L_{2j,2j+1} =:& \bar{f}_0(\sigma)
\end{aligned}$$

Let us define

$$f_k(\sigma) := \frac{\bar{f}_k(\sigma)}{\text{OPT}(\sigma)} = \frac{\bar{f}_k(\sigma)}{\sum_{j=0}^{n} L_{2j,2j+1}} \geq \frac{z(X^{\text{SMART}(q)}(\sigma))}{\text{OPT}(\sigma)} \tag{10}$$

for $k = 0, \ldots, n$. Observe that we are given $n+1$ different bounds on $\frac{z(X^{\textsc{smart}(q)}(\sigma))}{\textsc{Opt}(\sigma)}$. For every unsplittable sequence we consider its best bound, i. e., we get subsets of sequences of the form

$$\mathcal{S}_k(1, 2n+1) = \{\sigma \in \mathcal{S}(1, 2n+1) \mid f_k(\sigma) \leq f_i(\sigma), i = 0, \ldots, n\}.$$

Then it follows that

$$A(2n+1) \leq \sup_{\sigma \in \mathcal{S}(1,2n+1)} \min_{k=0,1,\ldots,n} f_k(\sigma) = \max_{k=0,\ldots,n} \sup_{\sigma \in \mathcal{S}_k(1,2n+1)} f_k(\sigma).$$

The main idea is to fix k and find a worst instance $\sigma \in \mathcal{S}_k(1, 2n+1)$. In order to simplify the notation we will write f_k instead of $f_k(\sigma)$. Now consider a fixed $k \in \{0, \ldots, n\}$ and assume that $f_k \leq f_i$ for all $i = 0, 1, \ldots, k-1$ and $i = k+1, k+2, \ldots, n$. Using the definition of f_k in equation (10) this is equivalent to

$$L_{2k,2k+1} \geq (q-1) L_{2i,2i+1} + (q-2) \sum_{j=i+1}^{k-1} L_{2j,2j+1} \quad i = 0, 1, \ldots, k-1, \tag{11}$$

$$(q-1) L_{2k,2k+1} \geq (2-q) \sum_{j=k+1}^{i-1} L_{2j,2j+1} + L_{2i,2i+1} \quad i = k+1, k+2 \ldots, n. \tag{12}$$

Finding a worst sequence in $\mathcal{S}_k(1, 2n+1)$ is equivalent to determine an upper bound on

$$f_k = \frac{\left(q + \frac{1}{q}\right) L_{01} + q \sum_{j=1}^{k-1} L_{2j,2j+1} + L_{2k,2k+1} + 2 \sum_{j=k+1}^{n} L_{2j,2j+1}}{L_{01} + \sum_{j=1}^{n} L_{2j,2j+1}}$$

while (11) and (12) are satisfied. Cumbersome and tedious calculations lead to

$$\sup_{\sigma \in \mathcal{S}_k(1,2n+1)} f_k(\sigma) \leq \begin{cases} \frac{q^2 - q + 2 - 2q(q-1)^{n+1}}{q - q(q-1)^{n+1}} & \text{if } 1 < q < 2 \\ \frac{3+4n}{2+2n} & \text{if } q = 2 \end{cases}$$

for $k = 0, \ldots, n$ and therefore

$$A(2n+1) \leq \max_{k=0,\ldots,n} \sup_{\sigma \in \mathcal{S}_k(1,2n+1)} f_k(\sigma) \leq \begin{cases} \frac{q^2 - q + 2 - 2q(q-1)^{n+1}}{q - q(q-1)^{n+1}} & \text{if } 1 < q < 2, \\ \frac{3+4n}{2+2n} & \text{if } q = 2. \end{cases} \tag{13}$$

Case 2: $B\,(2n)$
Assume that we are given an unsplittable sequence σ of Type 2 with $m = 2n$, i. e.,

$$z(X^{\textsc{smart}(q)}(\sigma)) = L_{0,1} + L_{2,3} + L_{4,5} \ldots + L_{2n-2,2n-1} + L_{2n}$$
$$\textsc{Opt}(\sigma) = L_0 + L_{1,2} + \ldots + L_{2n-1,2n}.$$

Since $X^{\textsc{smart}(q)}(\sigma)$ is the result of $\textsc{smart}\,(q)$ we know that

$$L_{2j,2j+1} \leq q L_{2j} \tag{14}$$

for all $j = 0, \ldots, n-1$.

First we are interested in the case where $n = 1$. Let $\sigma \in \mathcal{S}(2,2)$ then the corresponding competitive ratio is equal to

$$\frac{z(X^{\textsc{smart}(q)}(\sigma))}{\textsc{Opt}(\sigma)} = \frac{L_{0,1} + L_2}{L_0 + L_{1,2}}.$$

However, the result of $\textsc{smart}\,(q)$ can be bounded in two ways:

$$L_{0,1} + L_2 \leq L_0 + L_1 + L_2 \leq L_0 + 2L_{1,2}$$
$$L_{0,1} + L_2 \leq qL_0 + L_2 \leq qL_0 + L_{1,2}$$

Therefore, the competitive ratio is equal to

$$\frac{z(X^{\text{SMART}(q)}(\sigma))}{\text{OPT}(\sigma)} = \max\left\{\frac{L_0 + 2L_{1,2}}{L_0 + L_{1,2}}, \frac{qL_0 + L_{1,2}}{L_0 + L_{1,2}}\right\} \overset{(q-1)L_0 = L_{1,2}}{=} 2 - \frac{1}{q}.$$

Now consider an unsplittable sequence of $\mathcal{S}(2, m)$ with $m = 2n$ and $n \geq 2$. Then the objective value of $\text{SMART}(q)$ can be bounded as follows:

$$z(X^{\text{SMART}(q)}(\sigma)) = L_{0,1} + L_{2,3} + L_{4,5} \ldots + L_{2n-2,2n-1} + L_{2n} \leq L_0 + L_1 + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}.$$

Assume that $\sigma \in \mathcal{S}(2, 2n)$, then we have

$$\begin{aligned}
\frac{z(X^{\text{SMART}(q)}(\sigma))}{\text{OPT}(\sigma)} &= \frac{L_{0,1} + L_{2,3} + L_{4,5} \ldots + L_{2n-2,2n-1} + L_{2n}}{L_0 + L_{1,2} + \ldots + L_{2n-1,2n}} \\
&\leq \frac{L_0 + L_1 + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}}{L_0 + L_{1,2} + \ldots + L_{2n-1,2n}} \\
&\leq \frac{L_1 + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}}{L_{1,2} + \ldots + L_{2n-1,2n}}.
\end{aligned}$$

Consider a new sequence $\bar{\sigma}$ which results from σ by deleting the first request r_0. Due to the optimality of $\text{OPT}(\sigma)$ we know that

$$\text{OPT}(\bar{\sigma}) = L_{1,2} + L_{3,4} + \ldots + L_{2n-1,2n}.$$

On the other hand, if $L_{1,2} > qL_1$ then

$$X^{\text{SMART}(q)}(\bar{\sigma}) = L_1 + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}$$

holds. However, if $L_{1,2} \leq qL_1$ then no information on $X^{\text{SMART}(q)}(\bar{\sigma})$ is available. Therefore, we distinguish two cases:

- If $L_{1,2} > qL_1$ then

  $$\frac{z(X^{\text{SMART}(q)}(\sigma))}{\text{OPT}(\sigma)} \leq \frac{L_1 + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}}{L_{1,2} + \ldots + L_{2n-1,2n}} = \frac{X^{\text{SMART}}(q)(\bar{\sigma})}{\text{OPT}(\bar{\sigma})} \leq A(2n-1)$$

  holds because $\bar{\sigma} \in \mathcal{S}(2, 2n-1)$.

- If $L_{1,2} \leq qL_1$ then $L_2 \leq L_{1,2} \leq qL_1$ implies

  $$L_{1,2} = L_{1,2} - L_2 + L_2 \geq L_{1,2} - qL_1 + L_2 \geq (1-q)L_1 + L_2.$$

  Hence, we get

  $$\begin{aligned}
  \frac{z(X^{\text{SMART}(q)}(\sigma))}{\text{OPT}(\sigma)} &\leq \frac{L_1 + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}}{(1+q)L_1 + L_2 + L_{3,4} + \ldots + L_{2n-1,2n}} \\
  &\leq \max\left\{\frac{1}{1+q}, \frac{L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n}}{L_2 + L_{3,4} + \ldots + L_{2n-1,2n}}\right\} \leq \max\left\{\frac{1}{1+q}, B(2n-2)\right\}.
  \end{aligned}$$

  The last inequality holds, since the sequence $\tilde{\sigma}$ obtained from $\bar{\sigma}$ by deleting r_1 is an element of $\mathcal{S}(2, 2n-2)$. This procedure can be repeated and we get

  $$\begin{aligned}
  \frac{z(X^{\text{SMART}(q)}(\sigma))}{\text{OPT}(\sigma)} &\leq \max\left\{A(2n-1), A(2n-3), \ldots, A(3), \frac{1}{1+q}, B(2)\right\} \\
  &= \max\left\{A(2n-1), A(2n-3), \ldots, A(3), \frac{1}{1+q}, 2 - \frac{1}{q}\right\}.
  \end{aligned}$$

Observe that $2 - \frac{1}{q} \geq \frac{1}{1+q}$ holds for $1 < q \leq 2$. Therefore, we get

$$B(2n) \leq \begin{cases} 2 - \frac{1}{q} & \text{for } n = 1 \\ \max\left\{A(3), A(5), \ldots, A(2n-1), 2 - \frac{1}{q}\right\} & \text{for } n \geq 2. \end{cases} \tag{15}$$

Putting all together:

The available bounds on $A(2n+1)$ and $B(2n)$ are now used to get a bound on $r(\text{SMART}\,(q))$. Consider the following observations:

- The bound on $A(2n+1)$ is monotonically increasing in n.
- $2 - \frac{1}{q} \leq \min\left\{1 + \frac{1}{q}, \frac{3}{2}\right\}$ holds for all $1 < q \leq 2$, i.e., the bound on $B(2)$ is dominated by the bound on $A(1)$ for $1 < q \leq 2$.

These two observations immediately imply the following result:

$$\begin{aligned} r(\text{SMART}\,(q)) &= \max\left\{A\left(2\left\lfloor\frac{j-1}{2}\right\rfloor + 1\right), B\left(2\left\lfloor\frac{j}{2}\right\rfloor\right) \mid j = 1, \ldots, m\right\} \\ &\leq \begin{cases} \frac{q^2 - q + 2 - 2q(q-1)^{\left\lfloor\frac{m-1}{2}\right\rfloor+1}}{q - q(q-1)^{\left\lfloor\frac{m-1}{2}\right\rfloor+1}} & \text{if } 1 < q < 2 \\ \frac{3 + 4\left\lfloor\frac{m-1}{2}\right\rfloor}{2 + 2\left\lfloor\frac{m-1}{2}\right\rfloor} & \text{if } q = 2 \end{cases} \end{aligned} \tag{16}$$

This shows the correctness of Theorem 4.1.

The optimal choice of q depends on m. The following table contains an optimal value of q for $m \in \{1, \ldots, 8\}$:

m	ratio for $q = 2$	ratio for $q < 2$	optimal q^*	optimal ratio
$\{1,2\}$	1.5	$1 + \frac{1}{q}$	$q^* = 2$	$= 1.5$
$\{3,4\}$	1.75	$2 + \frac{1}{q^2} - \frac{1}{q}$	$q^* = 2$	$= 1.75$
$\{5,6\}$	≈ 1.83333	$\frac{1+q-2q^2+2q^3}{q-q^2+q^3}$	$q^* \approx 1.5652$	≈ 1.8084
$\{7,8\}$	1.875	$\frac{q^2-q+2-2q(q-1)^4}{q-q(q-1)^4}$	$q^* \approx 1.4694$	≈ 1.8219

Unfortunately, an optimal q is hard to find because one has to determine the roots of polynomials of high degree. Nevertheless, a good choice is to take $q^* = \sqrt{2}$ since $\sqrt{2}$ is optimal for $m \to \infty$. Simple calculations yield

$$r(\text{SMART}(\sqrt{2})) \leq 2\sqrt{2} - 1 < 2.$$

5 Analysis of the online algorithm CLEVER

In this section, we present and analyze the online algorithm CLEVER. The idea of this algorithm is that two requests are processed together in one time period if combined service is preferable to serving each request on its own. We will show that the competitive ratio of CLEVER is 3/2. Due to Theorem 2.1 this algorithm is optimal.

> **Algorithm** CLEVER
> At the beginning of each time period $t_i (i = 0 \ldots, m)$, do the following: If request r_{i-1} was postponed to period t_i and $L_{i-1,i} \leq \frac{2}{3}(L_{i-1} + L_i)$, then process the requests r_{i-1} and r_i together, otherwise postpone r_i.

Theorem 5.1. *Let m be an integer and $\sigma = (r_0, r_1, r_2, \ldots, r_m)$ an instance of the* OnlineTDAP*. Then,*

$$r(\text{CLEVER}) = \frac{3}{2}.$$

Proof. To compute the competitve ratio, we need to analyze $A\,(2n+1)$ and $B\,(2n)$.

We start by analyzing short sequences with two requests r_0 and r_1 ($n = 0$), where the online algorithm has to decide whether to process the requests separately or together. It is easy to see that serving both requests in time period t_1 is optimal, since we assumed Condition (2) to hold (triangle inequality). However, if $L_{0,1} > \frac{2}{3}\,(L_0 + L_1)$, then applying CLEVER requests r_0 and r_1 will be served separately yielding a competitive ratio of

$$\frac{L_0 + L_1}{L_{0,1}} < \frac{3}{2}\frac{L_{0,1}}{L_{0,1}} = \frac{3}{2}.$$

Case 1: $A\,(2n+1)$

Now assume an unsplittable sequence σ of Type 1 consisting of an even number of requests. Then the optimal strategy is of the form $X^*(\sigma) = \{D, I, D, \ldots, D, I\}$ with optimal value

$$\text{Opt}(\sigma) = L_{0,1} + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n,2n+1},$$

whereas CLEVER produces the answer set $X^{\text{CLEVER}}(\sigma) = \{D, D, I, \ldots, I, D\}$ with value

$$z\,(\text{CLEVER}(\sigma)) = L_0 + L_{1,2} + \cdots + L_{2n-1,2n} + L_{2n+1}.$$

Therefore the competitive ratio is

$$A\,(2n+1) = \frac{z\,(\text{CLEVER}(\sigma))}{\text{Opt}(\sigma)} = \frac{L_0 + L_{1,2} + \cdots + L_{2n-1,2n} + L_{2n+1}}{L_{0,1} + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n,2n+1}}.$$

Augmenting the nominator yields

$$\begin{aligned} A\,(2n+1) = & \frac{L_0 - L_{0,1} + L_{1,2} - L_{2,3} + \cdots - L_{2n-2,2n-1} + L_{2n-1,2n}}{L_{0,1} + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n,2n+1}} \\ & + \frac{L_{0,1} + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n+1}}{L_{0,1} + L_{2,3} + \cdots + L_{2n-2,2n-1} + L_{2n,2n+1}}. \end{aligned}$$

Since the second ratio is less than or equal to 1 (Condition (3): $L_{2n+1} \leq L_{2n,2n+1}$), it remains to be shown that the first one is at most $1/2$. Let us define

$$\begin{aligned} f\,(n) := & \frac{L_0 + \sum_{i=1}^{n} L_{2i-1,2i} - \sum_{i=0}^{n-1} L_{2i,2i+1}}{\sum_{i=0}^{n} L_{2i,2i+1}} \\ = & \frac{L_0 + \sum_{i=1}^{n-1} L_{2i-1,2i} + L_{2n-1,2n} - \sum_{i=0}^{n-2} L_{2i,2i+1} - L_{2n-2,2n-1}}{\sum_{i=0}^{n-1} L_{2i,2i+1} + L_{2n,2n+1}}. \end{aligned} \tag{17}$$

We know that $L_{2n-1,2n} \leq 2/3L_{2n-1} + 2/3L_{2n}$, since requests r_{2n-1} and r_{2n} are served together. Additionally, we know from Condition (3) that $L_{2n,2n+1} \geq L_{2n}$ and $L_{2n-2,2n-1} \geq L_{2n-1}$. Using this information we get

$$f\,(n) \leq \frac{L_0 + \sum_{i=1}^{n-1} L_{2i-1,2i} - \sum_{i=0}^{n-2} L_{2i,2i+1} + 2/3L_{2n-1} + 2/3L_{2n} - L_{2n-1}}{\sum_{i=0}^{n-1} L_{2i,2i+1} + L_{2n}}. \tag{18}$$

Moreover, we know that

$$L_{2n} \leq L_{2n-1,2n} \leq 2/3 L_{2n-1} + 2/3 L_{2n}.$$

The first inequality follows from Condition (3) and the second one, because requests r_{2n-1} and r_{2n} are served in the same time period, resulting in $L_{2n} \leq 2L_{2n-1}$. Observe that the right hand side of inequality (18) is monotonically increasing in L_{2n} and hence

$$\begin{aligned} f(n) &\leq \frac{L_0 + \sum_{i=1}^{n-1} L_{2i-1,2i} - \sum_{i=0}^{n-2} L_{2i,2i+1} - \frac{1}{3}L_{2n-1} + \frac{4}{3}L_{2n-1}}{\sum_{i=0}^{n-1} L_{2i,2i+1} + 2L_{2n-1}} \\ &= \frac{L_0 + \sum_{i=1}^{n-1} L_{2i-1,2i} - \sum_{i=0}^{n-2} L_{2i,2i+1} + L_{2n-1}}{\sum_{i=0}^{n-1} L_{2i,2i+1} + 2L_{2n-1}} \leq \max\left\{ \frac{L_0 + \sum_{i=1}^{n-1} L_{2i-1,2i} - \sum_{i=0}^{n-2} L_{2i,2i+1}}{\sum_{i=0}^{n-1} L_{2i,2i+1}}, \frac{1}{2} \right\}. \end{aligned} \tag{19}$$

Comparing inequality (17) with (19), we observe that

$$f(n) \leq \max\left\{ f(n-1), \frac{1}{2} \right\}.$$

Hence, we have to prove that the statement is true for $n = 1$, i.e., $f(1) \leq \frac{1}{2}$. Since $L_{1,2} \leq 2/3 (L_1 + L_2)$ and $L_{2,3} \geq L_2$ we get

$$f(1) = \frac{L_0 + L_{1,2} - L_{0,1}}{L_{0,1} + L_{2,3}} \leq \frac{L_0 - L_{0,1} + \frac{2}{3}L_1 + \frac{2}{3}L_2}{L_{0,1} + L_2}. \tag{20}$$

We know that $L_2 \leq 2L_1$ and (20) is monotonically increasing in L_2. Thus,

$$f(1) \leq \frac{L_0 - L_{0,1} + \frac{2}{3}L_1 + \frac{4}{3}L_1}{L_{0,1} + 2L_1} = \frac{L_0 - L_{0,1} + 2L_1}{L_{0,1} + 2L_1}.$$

Because requests r_0 and r_1 are not served in one time period

$$L_{0,1} > \frac{2}{3}L_0 + \frac{2}{3}L_1 \text{ and } L_0 < \frac{3}{2}L_{0,1} - L_1.$$

Therefore,

$$f(1) \leq \frac{\frac{3}{2}L_{0,1} - L_1 - L_{0,1} + 2L_1}{L_{0,1} + 2L_1} = \frac{\frac{1}{2}L_{0,1} + L_1}{L_{0,1} + 2L_1} = \frac{1}{2}.$$

Summarizing the calculations for the competitive ratio in case of an unsplittable sequence of Type 1 we get

$$A(2n+1) \leq f(n) + 1 \leq \max\left\{ \frac{1}{2}, \ldots, \frac{1}{2} \right\} + 1 \leq 3/2.$$

Case 2: $B(2n)$ In an analogue way we get

$$B(2n) \leq \max\left\{ \frac{1}{3}, \frac{1}{2}, \ldots, \frac{1}{2} \right\} + 1 \leq \frac{3}{2}.$$

Therefore, the competitive ratio of the online algorithm CLEVER is

$$\begin{aligned} r(\text{CLEVER}) \leq \frac{z(\text{CLEVER}(\sigma))}{\text{OPT}(\sigma)} &= \max\left\{ A\left(2\left\lfloor \frac{j-1}{2} \right\rfloor + 1\right), B\left(2\left\lfloor \frac{j}{2} \right\rfloor\right) \mid j = 1, \ldots, m \right\} \\ &\leq \max\left\{ \frac{4}{3}, \frac{3}{2}, \ldots, \frac{3}{2} \right\} \leq \frac{3}{2}. \end{aligned}$$

This proves the desired result. □

References

[1] S. Albers. Online algorithms: A survey. *Mathematical Programming*, 97:3–26, 2003. Invited paper at ISMP 2003.

[2] S. Albers. *Interactive Computation: The New Paradigm*, chapter Online algorithms, pages 143–164. Springer-Verlag, 2006.

[3] E. Angelelli, M. G. Speranza, and M. W. P. Savelsbergh. Competitive analysis for dynamic multiperiod uncapacitated routing problems. *Networks*, 49(4):308–317, 2007.

[4] E. Angelelli, M. G. Speranza, and M. W. P. Savelsbergh. Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters*, 35:713–721, 2007.

[5] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[6] A. Fiat and G. J. Woeginger, editors. *Online algorithms: The state of the art*, volume 1442 of *Lecture Notes in Computer Science*. Springer, 1998.

[7] S. Heinz, S. O. Krumke, N. Megow, J. Rambau, A. Tuchscherer, and T. Vredeveld. The online target date assignment problem. In Thomas Erlebach and G. Persiano, editors, *Approximation and Online Algorithms (WAOA 2005)*, volume 3879 of *Lecture Notes in Computer Science*, pages 230–243. Springer, Berlin, Germany, 2006.