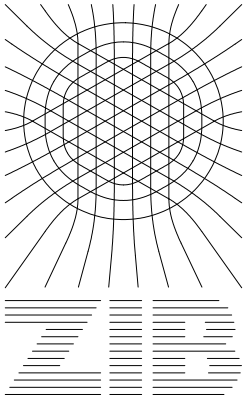

Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SVEN O. KRUMKE LUIGI LAURA MAARTEN LIPMANN
ALBERTO MARCHETTI-SPACCAMELA
WILLEM E. DE PAEPE DIANA POENSGEN LEEN STOUGIE

Non-Abusiveness Helps: An $\mathcal{O}(1)$-Competitive Algorithm for Minimizing the Maximum Flow Time in the Online Traveling Salesman Problem

ZIB-Report 02-36 (October 2002)

Non-Abusiveness Helps: An $\mathcal{O}(1)$-Competitive Algorithm for Minimizing the Maximum Flow Time in the Online Traveling Salesman Problem

Sven O. Krumke* Luigi Laura† Maarten Lipmann‡ Alberto Marchetti-Spaccamela†
Willem E. de Paepe¶ Diana Poensgen* Leen Stougie**

22nd October 2002

Abstract

In the online traveling salesman problem (OLTSP) requests for visits to cities arrive online while the salesman is traveling. We study the $F_{\max}$-OLTSP where the objective is to minimize the maximum flow time. This objective is particularly interesting for applications. Unfortunately, there can be no competitive algorithm, neither deterministic nor randomized. Hence, competitive analysis fails to distinguish online algorithms. Not even resource augmentation which is helpful in scheduling works as a remedy. This unsatisfactory situation motivates the search for alternative analysis methods.

We introduce a natural restriction on the adversary for the $F_{\max}$-OLTSP on the real line. A *non-abusive adversary* may only move in a direction if there are yet unserved requests on this side. Our main result is an algorithm which achieves a constant competitive ratio against the non-abusive adversary.

*Konrad-Zuse-Zentrum für Informationstechnik Berlin, Germany. Email: {krumke,poensgen}@zib.de

†Universita di Roma "La Sapienza", Italy Email: {alberto,laura}@dis.uniroma1.it

‡Technical University of Eindhoven, The Netherlands. Email: m.lipmann@tue.nl

¶Technical University of Eindhoven, The Netherlands. Email: w.e.d.paepe@tm.tue.nl

**Technical University of Eindhoven, and Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. Email: leen@win.tue.nl

1 Introduction

In the online traveling salesman problem (OLTSP) requests for visits to cities arrive online while the salesman is traveling. An online algorithm learns from the existence of a request only at its release time. The OLTSP has been studied for the objectives of minimizing the makespan [1, 2, 5], the weighted sum of completion times [5, 9], and the maximum/average flow time [6]. In view of applications, the maximum flow time is of particular interest. For instance, it can be identified with the maximal dissatisfaction of customers. Alas, there can be no competitive algorithm, neither deterministic nor randomized [6]. Moreover, in contrast to scheduling [11] resource augmentation, e.g. providing the online algorithm with a faster server, does not help, the crucial difference being that servers move in space.

The only hope to overcome the weaknesses of standard competitive analysis in the context of the $F_{\max}$-OLTSP is to restrict the powers of the adversary. In this paper we consider the $F_{\max}$-OLTSP on the real line and introduce a natural restriction on the adversary: a *non-abusive* adversary may move its server only in a direction, if yet unserved requests are pending on that side. We construct an algorithm, called DETOUR which achieves a competitive ratio of eight against the non-abusive adversary.

Related Work. Koutsoupias and Papadimitriou introduced the concept of *comparative analysis* for restricting the adversary [8]. The *fair adversary* of Blom et al. [3]

implements this concept in the context of the OLTSP as follows: a fair adversary may only move within the convex hull of all requests released so far. While one can obtain improved competitiveness results for the minimization of the makespan against a fair adversary [3], still a constant competitive ratio for the maximum flow time is out of reach (see Theorem 2.1). The non-abusive adversary presented in this paper can be viewed as a refinement of the fair adversary.

In [7] it is shown that minimizing the average flow time in scheduling is obnoxiously hard, both online and offline.

Paper Outline. In Section 2 we formally define the $F_{\max}$-OLTSP and the non-abusive adversary. We also show lower bound results for the competitive ratio against a fair and non-abusive adversary, respectively. Section 3 presents our algorithm DETOUR, the proof of its performance is sketched in Section 4.

2 Preliminaries

An instance of the *Online Traveling Salesman Problem* ($F_{\max}$-OLTSP) consists of a metric space $M = (X, d)$ with a distinguished origin $o \in X$ and a sequence $\sigma = r_1, \ldots, r_m$ of requests. A server is located at the origin o at time 0 and can move at most at unit speed. In this paper we are concerned with the special case that M is $\mathbb{R}$, the real line endowed with the Euclidean metric $d(x, y) = |x - y|$; the origin o equals the point 0. Each *request* is a pair $r_i = (t_i, x_i)$, where $t_i \in \mathbb{R}$ is the time at which request r_i is released, and $x_i \in X$ is the point in the metric space to be visited. We assume that the sequence $\sigma = r_1, \ldots, r_m$ of requests is given in order of non-decreasing release times. For a real number t, we denote by $\sigma_{\leq t}$ ($\sigma_{<t}$) the subsequence of requests in σ released up to time t (strictly before time t).

An online algorithm ALG gets to know request r_j only at its release time t_j. In particular, ALG has neither information about the release time of the last request nor about the total number of requests. Hence, at any moment in time t, ALG must make its decisions only knowing the requests in $\sigma_{\leq t}$. An offline algorithm has complete knowledge about the sequence σ already at time 0.

Given a sequence σ of requests, an algorithm ALG for the $F_{\max}$-OLTSP must find a route for the server which starts in the origin and visits each point in σ, but not earlier than its release time. By C_j^{ALG} and $F_j^{\mathsf{ALG}} = C_j^{\mathsf{ALG}} - t_j$ we denote the completion time and flow time of request r_j, respectively, in the solution produced by ALG. The goal in the $F_{\max}$-OLTSP is to minimize the maximum flow time $\mathsf{ALG}(\sigma) := \max_j F_j^{\mathsf{ALG}}$.

Let OPT denote an optimal offline algorithm. A deterministic online algorithm ALG for the $F_{\max}$-OLTSP is *c-competitive*, if there exists a constant c such that for any request sequence σ, $\mathsf{ALG}(\sigma) \leq c \cdot \mathsf{OPT}(\sigma)$. If ALG is randomized, then $\mathsf{ALG}(\sigma)$ is replaced by the expected solution value (this corresponds to the oblivious adversary, see [4]). The *competitive ratio* of ALG is the infimum over all c such that ALG is c-competitive.

The following lower bound result shows that the fairness restriction on the adversary introduced in [3] is still not strong enough to allow for competitive algorithms in the $F_{\max}$-OLTSP.

Theorem 2.1 *No randomized algorithm for the $F_{\max}$-OLTSP on $\mathbb{R}$ can achieve a constant competitive ratio against an oblivious adversary. This result still holds, even if the adversary is* fair, *i.e., if at any moment in time t the server operated by the adversary is within the convex hull of the origin and the requested points from $\sigma_{\leq t}$.*

Proof. Let $\varepsilon > 0$ and $k \in \mathbb{N}$. We give two request sequences $\sigma_1 = (\varepsilon, \varepsilon), (2\varepsilon, 2\varepsilon), \ldots, (k\varepsilon, k\varepsilon), (T, 0)$ and $\sigma_2 = (\varepsilon, \varepsilon), (2\varepsilon, 2\varepsilon), \ldots, (k\varepsilon, k\varepsilon), (T, k\varepsilon)$, each with probability $1/2$, where $T = 4k\varepsilon$.

The expected cost of an optimal fair offline solution is at most ε, while any deterministic online algorithm has cost at least $k\varepsilon/2$. The claim now follows by applying Yao's principle [4, 10]. □

The fair adversary is still too powerful in the sense that it can move to points where it knows that a request will pop up without revealing any information to the online server before reaching the point. A *non-abusive* adversary is stripped of this power.

Definition 2.2 (Non-Abusive Adversary)
An adversary ADV *for the* OLTSP *on* $\mathbb{R}$ *is* non-abusive*, if the following holds: At any moment in time t, where the adversary moves its server from its current position $p^{\mathsf{ADV}}(t)$ to the right (left), there is a request from $\sigma_{\leq t}$ to the right (left) of $p^{\mathsf{ADV}}(t)$ which* ADV *has not served yet.*

In the sequel we slightly abuse notation and denote by $\mathsf{OPT}(\sigma)$ the maximal flow time in an optimal *non-abusive* offline solution for the sequence σ.

The following result shows that the $F_{\max}$-OLTSP is still non-trivial against a non-abusive adversary.

Theorem 2.3 *No deterministic algorithm for the $F_{\max}$-OLTSP on $\mathbb{R}$ can achieve a competitive ratio less than 2 against a non-abusive adversary.*

Proof. Let ALG be any deterministic online algorithm. The adversary first presents the following $2m$ requests: $(0, \pm 1), (3, \pm 2), \ldots, (\sum_{k=1}^{m-1}(1+2k), \pm m)$. W.l.o.g., let ALG serve the request in $-m$ later than the on ein $+m$, and let T be the time it reaches $-m$. Clearly, $T \geq \sum_{k=1}^{m}(1+2k)$. At time T, the adversary presents one more request in $+(3m+1)$ which results in a flow time of at least $4m+1$ for ALG. On the other hand, a non-abusive offline algorithm can serve all of the first $2m$ requests with maximum flow time $2m+1$ by time $\sum_{k=1}^{m}(1+2k)$, ending with the request at $+m$. From there it can easily reach the last request with flow time $2m+1$. The theorem follows by letting $m \to \infty$. □

3 The Algorithm DETOUR

We denote by $p^{\mathsf{DTO}}(t)$ the position of the server operated by DETOUR (short DTO) at time t. The terms *ahead of* and *in the back of* the server refer to positions on the axis w.r.t. the direction the server currently moves: if it is moving from left to right on the $\mathbb{R}$-axis, "ahead" means to the right of the server's current position, while a request "in the back" of the server is to its left. The other case is defined analogously.

Given a point $p \in \mathbb{R}$, we call the pair (t_i, x_i) *more critical* than the request $r_j = (t_j, x_j)$ w.r.t. p if both x_i and x_j are on the same side of p, and $d(p, x_i) - t_i > d(p, x_j) - t_j$. If at time t, request r_i is more critical than r_j w.r.t. DTO's position $p^{\mathsf{DTO}}(t)$, this yields that DTO cannot serve r_i with the same (or a smaller) flow time than r_j. Moreover, r_i remains more critical than r_j after time t as long as both requests are unserved. Conversely, we have the following observation.

Observation 3.1 *If, at time t, request r_i is more critical than r_j w.r.t. $p^{\mathsf{DTO}}(t)$, and DTO moves straight ahead to the more distant request after having served the one closer to $p^{\mathsf{DTO}}(t)$, then DTO's flow time for r_j is at most the flow time it achieves for r_i.* □

The *critical region* $\vee(r_j, p, G)$ w.r.t. a request r_j, a point $p \in \mathbb{R}$ and a bound G for the allowed maximal flow time contains all those pairs $(t, x) \in \mathbb{R}_+ \times \mathbb{R}$ with the property that (i) (t, x) is more critical than r_j w.r.t. p, and (ii) $t + d(x, x_j) - t_j \leq G$.

In the setting of DTO, p will be the position of the online server at a certain time t'. Condition (ii) has the meaning that a request in (t, x) could be served before r_j in an offline tour serving both r_j and (t, x) with maximal flow time at most G.

DTO's decisions at time t are based on an approximation, called the *guess* $G(t)$, of $\mathsf{OPT}(\sigma_{\leq t})$. DTO's rough strategy is to serve all requests in a first-come-first-serve (FCFS) manner. However, blindly following an FCFS-scheme makes it easy for the adversary to fool the algorithm. DTO enforces the offline cost in a malicious sequence to increase by making a detour on its way to the next target: it first moves its server in the "wrong direction" as long as it can serve the target with flow time thrice the guess. If the guess changes, the detour, and possibly the target, are adjusted accordingly (this requires some technicalities in the description of the algorithm).

Our algorithm DTO can assume three modes:

idle In this mode, DTO's server has served all unserved requests, and is waiting for new requests at the point at which it served the last request.

focus Here, the server is moving in one direction serving requests until a request in its back becomes the oldest unserved one or all requests have been served.

detour In this case, the server is moving away from its current target (possibly serving requests on the way), thus making a "detour".

At any time, at most one unserved request is marked as a *target* by DTO. Moreover, there it keeps at most one critical region, denoted by $\vee$. Before we formalize the behavior of DTO we specify important building blocks for the algorithm.

- *Guess Update:* Replace the current guess value G by G', defined as follows: If $G = 0$, then $G' := \mathsf{OPT}(\sigma_{\leq t})$. If $G > 0$, then $G' := 2^a G$, where a is the smallest integer k such that $\mathsf{OPT}(\sigma_{\leq t}) \leq 2^k G$.

- *Target Selection:* Given a candidate set C and the current time t, let $s_0 = (t_0, x_0)$ be the most critical request from C w.r.t. $p^{\mathsf{DTO}}(t)$ with the property that s_0 is *feasible* in the following sense:

  Let X_0 be the point ahead of the server such that $t + d(p^{\mathsf{DTO}}(t), X_0) + d(X_0, x_0) = t_0 + 3G$, provided such a point exists, otherwise let $X_0 := p^{\mathsf{DTO}}(t)$. Define the *turning point* $\mathrm{TP}_0 = (T_0, X_0)$, where $T_0 := t + d(p^{\mathsf{DTO}}(t), X_0)$. There is no unserved request ahead of the server further away from $p^{\mathsf{DTO}}(t)$ than TP_0 and older than s_0.

  If necessary, unmark the current target and turning point. Mark s_0 as a target and set TP_0 to be the current turning point.

- *Mode Selection:* If $\mathrm{TP}_0 \neq p^{\mathsf{DTO}}(t)$, then set $\vee := \vee(s_0, p^{\mathsf{DTO}}(t), G)$ and enter the detour mode. Otherwise, set $\vee := \emptyset$ and unmark s_0 as a target. Change the direction and enter the focus mode.

We now specify for each of the three states how DTO reacts to possible events. All events not mentioned are ignored. In the beginning, the guess value is set to $G := 0$, $\vee := \emptyset$ and the algorithm is in the idle mode.

Idle Mode In the idle mode DTO waits for the next request to occur.

- A new request is released at time t.

  The pair $(t, p^{\mathsf{DTO}}(t))$ is called a *selection point*. DTO performs a *guess update*. The direction of the server is defined to be such that the oldest unserved request is in its back. In case that there is one oldest request on both sides of the server chooses to have the most critical one in the server's back.

  DTO defines C to be the set of unserved requests in the back of the server and performs a *target selection*, followed by a *mode selection*.

Detour Mode In this mode, DTO has a current target $s_m = (t_m, x_m)$, a critical region $\vee \neq \emptyset$ and a turning point TP. Let T be the time DTO entered the detour mode and s_0 the then chosen target. The server moves towards TP until one of following two events happens, where the first one has a higher priority than the second one (i.e., the first event is always checked for first).

- A new request is released at time t or an already existing request has just been served.

  DTO performs a *guess update*. Then it enlarges the critical region to $\vee := \vee(s_0, p^{\mathsf{DTO}}(T), G)$ where G is the updated guess value. Replace the old turning point by a new point TP which satisfies $t + d(p^{\mathsf{DTO}}(t), \mathrm{TP}) + d(\mathrm{TP}, x_m) = t_m + 3G$ for the updated guess value G.

  DTO defines C to be the set of unserved requests which are in $\vee$ and more critical than the current target s_m. If $C \neq \emptyset$, it executes the *target selection* and the *mode selection*. If $C = \emptyset$, DTO remains in the detour mode.

- The turning point TP is reached at time t.

  DTO unmarks the current target, sets $\vee := \emptyset$ and clears the turning point. The server reverses direction and enters the focus mode.

Focus Mode When entering the focus mode, DTO's server has a direction. It moves in this direction, reacting to the following events:

- A new request is released at time t.

  A *guess update* is performed, and the server remains in the focus mode.

- The last unserved request has been served.

  The server stops, forgets its direction and enters the idle mode.

- A request in the back of the server becomes the oldest unserved request.

  If this happens at time t, the pair $(t, p^{\mathsf{DTO}}(t))$ is also called a *selection point*.

  DTO defines C to be the set of unserved requests in the back of the server and performs a *target selection*, followed by a *mode selection* (exactly as in the idle mode).

4 Analysis of DETOUR

For the analysis of DTO it is convenient to compare intermediate solutions $\mathsf{DTO}(\sigma_{\le t})$ not only to the optimal non-abusive solution on $\sigma_{\le t}$, but to a whole class of offline solutions $\mathcal{ADV}(t)$ which depend on the guess value $G(t)$ maintained by DTO. Notice that for any time t we have $\mathsf{OPT}(\sigma_{\le t}) \le G(t) \le 2\mathsf{OPT}(\sigma_{\le t})$.

Definition 4.1 *By $\mathcal{ADV}(t)$, we denote the set of all non-abusive offline solutions for the sequence $\sigma_{\le t}$ with the property that the maximum flow time of any request in $\sigma_{\le t}$ is bounded from above by $G(t)$.*

Let $r_i \in \sigma_{\le t}$. The smallest achievable flow time $\alpha_i(t)$ *is defined to be minimum flow time of r_i taken over all solutions in $\mathcal{ADV}(t)$.*

Note that, by definition, $\alpha_i(t) \le G(t)$. In general, $\alpha_i(t) \ge \alpha_i(t')$ for $t' > t$, as an increase in the allowed flow time can help an adversary to serve a request r_i earlier. On the other hand, we have the following property.

Observation 4.2 *If $t \le t'$ and $G(t) = G(t')$, then $\alpha_i(t) \le \alpha_i(t')$ for any request $r_i = (t_i, x_i)$ with $t_i \le t$.*

To derive bounds on the flow times for DTO we would like to conclude as follows: if request r_i is served by DTO "in time" and r_j is served directly after r_i (i.e., without any detour in between), then r_j is also served "in time".

Definition 4.3 (Served in time)
Given $r_i = (t_i, x_i)$, we define $\tau_i := \max\{t_i, T\}$, where T is the last time DTO *reverses direction before serving r_i. Furthermore, we say that r_i is* served in time *by* DTO*, if $C_i^{\mathsf{DTO}} \le t_i + 3G(\tau_i) + \alpha_i(\tau_i)$.*

Notice that any request r_i served in time is served with a flow time of at most $4G(\tau_i) \le 4G(C_i^{\mathsf{DTO}})$ since $\alpha_i(\tau_i) \le G(\tau_i)$ by definition.

Lemma 4.4 *Let $r_i = (t_i, x_i)$ and $r_j = (t_j, x_j)$ be two requests such that:*

(i) r_i is served in time by DTO*,*
(ii) $C_j^{\mathsf{DTO}} \le C_i^{\mathsf{DTO}} + d(x_i, x_j)$,
(iii) $\tau_i \le \tau_j$,
(iv) r_j is served after r_i by all $\mathsf{ADV} \in \mathcal{ADV}(\tau_j)$.

Then, DTO *serves r_j in time.*

Proof. From (iv) we have

$$t_j + \alpha_j(\tau_j) \ge t_i + \alpha_i(\tau_j) + d(x_i, x_j). \tag{1}$$

In particular, this means that

$$t_i + d(x_i, x_j) \le t_j + G(\tau_j). \tag{2}$$

By (i), (ii), and the definition of in time, we get:

$$C_j^{\mathsf{DTO}} \le t_i + \alpha_i(\tau_i) + 3G(\tau_i) + d(x_i, x_j). \tag{3}$$

If $G(\tau_i) = G(\tau_j)$, we have that $\alpha_i(\tau_i) \le \alpha_i(\tau_j)$ by Observation 4.2. In this case, inequality (3) yields that

$$\begin{aligned} C_j^{\mathsf{DTO}} &\le t_i + \alpha_i(\tau_j) + 3G(\tau_j) + d(x_i, x_j) \\ &\le t_j + \alpha_j(\tau_j) + 3G(\tau_j) \qquad \text{by (1).} \end{aligned}$$

If $G(\tau_i) < G(\tau_j)$, it must hold that $2G(\tau_i) \le G(\tau_j)$, and inequality (3) yields

$$\begin{aligned} C_j^{\mathsf{DTO}} &\le t_i + 4G(\tau_i) + d(x_i, x_j) \\ &\le t_j + G(\tau_j) + 4G(\tau_i) \qquad \text{by (2)} \\ &\le t_j + 3G(\tau_j). \end{aligned}$$

□

An easy but helpful condition which ensures that assumption (iv) of Lemma 4.4 holds, is that $t_j + d(x_j, x_i) > t_i + G(t)$. This yields the following observation which will be used frequently in order to apply Lemma 4.4:

Observation 4.5 *(i) If $d(x_i, x_j) > G(t)$ and $t_i \le t_j \le t$, then r_i, the older request, must be served before r_j in any offline solution in $\mathcal{ADV}(t)$.*
(ii) If a request r_j is outside the critical region $\vee(r_i, p, G(t))$ valid at time t, request r_j is served after r_i in any offline solution in $\mathcal{ADV}(t)$.

We define a *busy period* to be the time between the moment in time when DTO leaves the idle mode and the next time the idle mode is entered again.

Lemma 4.6 *Suppose that at the beginning of a busy period at time t each request r_j served in one of the preceeding busy periods was served by* DTO *with a flow time at most $4\,G(C_j^{\mathsf{DTO}})$. Then, $d(p^{\mathsf{DTO}}(t), p^{\mathsf{ADV}}(t)) \le 5/2G(t)$ for any $\mathsf{ADV} \in \mathcal{ADV}(t)$.*

Proof. The claim of the lemma is trivially true for the first busy period, since a non-abusive adversary must keep its server in the origin until the first request is released. Hence, it suffices to consider the later busy periods.

Let r_l be last request served by DTO in the preceeding busy period, so DTO enters the idle mode at time C_l^{DTO} again. Consider that $\mathsf{ADV} \in \mathcal{ADV}(t)$ in which the adversary's server is furthest away from $p^{\mathsf{DTO}}(t) = x_l$.

Case 1: At time t, ADV has served all requests in $\sigma_{<t}$.

In this case, since ADV is non-abusive, its server satisfies $p^{\mathsf{ADV}}(t) = x_k$ for the request r_k it served last. DTO must have served r_k in the preceeding busy period, hence no later than r_l. This gives $C_k^{\mathsf{DTO}} \leq C_l^{\mathsf{DTO}} - d(x_k, x_l)$ and we obtain that

$$t_k \leq C_l^{\mathsf{DTO}} - d(x_k, x_l) \leq t_l + 4G(C_l^{\mathsf{DTO}}) - d(x_k, x_l),$$

because r_l was served with a flow time of at most $4G(C_l^{\mathsf{DTO}})$ according to the assumption of the lemma. On the other hand, ADV serves r_l no later than r_k, which implies that $t_l + d(x_k, x_l) \leq t_k + G(C_l^{\mathsf{DTO}})$. The two inequalities together yield

$$d(x_k, x_l) \leq t_k - t_l + G(C_l^{\mathsf{DTO}}) \leq 5G(C_l^{\mathsf{DTO}}) - d(x_k, x_l).$$

hence the claim, since $d(p^{\mathsf{DTO}}(t), p^{\mathsf{ADV}}(t)) = d(x_k, x_l)$.

Case 2: At time t, there is a request from $\sigma_{<t}$ which has not been served yet by ADV.

If r_l has not been served by ADV at time t, then the distance $d(p^{\mathsf{ADV}}(t), x_l) = d(p^{\mathsf{ADV}}(t), p^{\mathsf{DTO}}(t))$ is at most $G(t)$, because otherwise the adversary's flow time for r_l would be greater than $G(t)$.

Otherwise, r_l has been served, but another request in $\sigma_{<t}$ is yet unserved by ADV. Let r_k be the request in $\sigma_{<t}$ which is furthest away from r_l and yet unserved by ADV. The same reasoning as in Case 1 shows that $d(x_k, x_l) \leq \frac{5}{2}G(C_l^{\mathsf{DTO}})$. So, if the adversary's server is between r_k and r_l, the claim holds true. Assume that the adversary's server is further away from r_l than r_k. Since the adversary is non-abusive, there must be a request r_j even further away from r_l than $p^{\mathsf{ADV}}(t)$, which ADV served last before (or at) time t. In particular, ADV served r_l before r_j. Thus, the same arguments as in Case 1 apply to r_j in place of r_k, showing that $d(x_j, x_l) \leq \frac{5}{2}G(C_l^{\mathsf{DTO}})$. □

We further subdivide each busy period into *phases*, where a phase is defined to be the time between two subsequent selection points of DTO. Remember that DTO reaches a selection point whenever it leaves the idle mode, and each time at which a request in the server's back becomes the oldest unserved one. The following statement is the key theorem of our analysis.

Theorem 4.7 *The following is true for any phase $\rho \geq 1$:*

(a) At any time t in phase ρ at which DTO *is in the detour mode, it holds that $d(X_i, x_i) \geq G(t)$ for the turning point TP$_i = (X_i, T_i)$ valid at that time and its corresponding target $r_i = (t_i, x_i)$. Moreover, if at some time t during phase ρ, a request r_i failed to become a new target only because it was infeasible, the above inequality holds as well for r_i and its hypothetical turning point TP$_i$.*

(b) Any request r_j served in phase ρ is served with a flow time of at most $4G(C_j^{\mathsf{DTO}})$.

(c) The last request served in phase ρ is served in time.

Proof. We prove the statement by induction on the total number of phases. In the inductive step we distinguish whether phase ρ is the first phase of a busy period or not. The former case includes the induction base ($\rho = 1$), i.e., the first phase of the first busy period, as a special case.

Let $\rho \geq 1$ be the number of the phase under consideration and assume that the three statements of the theorem all hold true for each phase $\rho' < \rho$, provided such a phase exists. Note that in the case that phase ρ is the first phase of a busy period, Lemma 4.6 can be applied.

At the beginning of phase ρ, DTO determines a turning point which might be replaced later on in the phase. We call the turning point TP$^\rho = (T^\rho, X^\rho)$ at which the server actually reverses direction the *realized turning point* of the phase ρ. Each turning point ever considered has a corresponding target. Note that both the realized turning point TP$^\rho$ and its corresponding target $s^\rho = (t^\rho, x^\rho)$ are reached by DTO in the same phase. Moreover, at any time when DTO is in the detour mode, the algorithm has a valid turning point and a corresponding target.

Throughout the whole proof we assume without loss of generality that the realized turning point is to the right of the final target, that is, in phase ρ, DTO moves to the right while in the detour mode and to the left after entering the focus mode. We may also assume without loss of generality that at time 0 a request appears in the origin since this request does not increase the offline cost.

Proof of Statement (a):

Let $\text{TP}_0 = (T_0, X_0)$ be the first turning point chosen in phase ρ, TP_1 the next one, etc. until TP^ρ, the realized turning point of the phase. Let $s_i = (t_i, x_i)$ be the target corresponding to TP_i (s_i is released at time t_i). Part (a) is proven by induction on the number of turning points in the considered phase ρ.

1. Phase ρ is the first phase of a busy period.

The first target $s_0 = (t_0, x_0)$ must be among the requests whose release initiates the start of the busy period at time t_0, and $\text{TP}_0 = (T_0, X_0)$ is chosen such that $t_0 + d(p^{\mathsf{DTO}}(t_0), X_0) + d(x_0, X_0) \geq t_0 + 3G(t_0)$. Since $d(p^{\mathsf{DTO}}(t_0), X_0) < d(x_0, X_0)$, it readily follows that $d(X_0, x_0) > \frac{3}{2}\, G(t_0)$.

Assume that (a) holds for the turning points $\text{TP}_0, \ldots, \text{TP}_{i-1}$ of phase ρ. We prove (a) for the next turning point $\text{TP}_i = (T_i, X_i)$. Assume that TP_i replaces TP_{i-1} at time t of phase 1, and let $s_i = (t_i, x_i)$ be TP_i's corresponding target.

If s_i was released at time t_0, then the turning point TP_i planned by DTO at time t is exactly the same as if the guess value at time t_0 had already been $G(t)$ and s_i had been selected as a target at time t_0. Exactly as before we can conclude that $d(X_i, x_i) \geq \frac{3}{2}\, G(t)$.

If s_i was released later than t_0, the detour taken by DTO is only longer as the one chosen if s_i had been released already at time t_0. Hence, the arguments of above apply again and $d(X_i, x_i) \geq \frac{3}{2}\, G(t)$.

2. Phase ρ is not the first phase of a busy period.

Again, we first consider $\text{TP}_0 = (T_0, X_0)$, the turning point planned first in phase ρ. Both s_0 and T_0 are determined in the selection point SP which marks the end of phase $\rho - 1$ and the start of phase ρ. It must hold that $\text{SP} = (C_l^{\mathsf{DTO}}, x_l)$ for some request r_l. When DTO serves request r_l, the oldest unserved request, call it r_z, is in its back. Observe that SP cannot be reached before the final target $s^{\rho-1}$ of phase $\rho - 1$ is served: If that was the case, there would be an unserved request in the back of DTO's server before $s^{\rho-1}$ is reached which is older than $s^{\rho-1}$. But in that case, $s^{\rho-1}$ would have been infeasible at the time it became a target, which is a contradiction.

Assume first that s_0 is located between $X^{\rho-1}$ and x_l. Then, the release time of s_0 satisfies

$$t_0 \geq C_l^{\mathsf{DTO}} - d(x_0, x_l), \tag{4}$$

because otherwise s_0 would have been served on the way to r_l. Since TP_0 is chosen at time C_l^{DTO} in such a way that s_0 is served not earlier than time $t_0 + 3G(C_l^{\mathsf{DTO}})$, we have

$$C_l^{\mathsf{DTO}} + d(x_l, X_0) + d(X_0, x_0) \geq t_0 + 3G(C_l^{\mathsf{DTO}}).$$

It follows that

$$\begin{aligned} d(X_0, x_0) &\geq t_0 + 3G(C_l^{\mathsf{DTO}}) - C_l^{\mathsf{DTO}} - d(x_l, X_0) \\ &\geq 3G(C_l^{\mathsf{DTO}}) - d(x_0, x_l) - d(x_l, X_0) \quad \text{by (4)} \\ &= 3G(C_l^{\mathsf{DTO}}) - d(x_0, X_0). \end{aligned}$$

Hence $d(x_0, X_0) \geq \frac{3}{2}\, G(C_l^{\mathsf{DTO}})$.

We now have to cover the case that s_0 is further away from r_l than $\text{TP}^{\rho-1}$, that is, $d(x_0, x_l) > d(X^{\rho-1}, x_l)$. Notice that r_l must be older than s_0: If s_0 was older, the oldest unserved request would have been in DTO's back before reaching r_l, and $(C_l^{\mathsf{DTO}}, x_l)$ would not have been the selection point. Observe also that $d(X^{\rho-1}, x_l)$ is at least the distance between the realized turning point $\text{TP}^{\rho-1}$ and the corresponding target, as the final target of a phase is always served within that phase, as shown above. From the inductive hypothesis for phase $\rho - 1$, Statement (a), we obtain that

$$d(x_l, x_0) > G(T^{\rho-1}). \tag{5}$$

As $d(X_0, x_0) \geq d(x_l, x_0)$, the only interesting case to consider is that $G(T^{\rho-1}) < G(C_l^{\mathsf{DTO}})$. So let $2^a G(T^{\rho-1}) = G(C_l^{\mathsf{DTO}})$ for some integer $a \geq 1$. We need to distinguish two cases.

Case 1: $t_0 \geq T^{\rho-1}$. In this case we have

$$T^{\rho-1} + d(X^{\rho-1}, X_0) + d(X_0, x_0) \geq t_0 + 3G(C_l^{\mathsf{DTO}}),$$

as DTO's server started from $X^{\rho-1}$ at time $T^{\rho-1}$ and chooses the turning point TP_0 at time C_l^{DTO} in such that the corresponding target s_0 is not served with a smaller flow time than $3G(C_l^{\mathsf{DTO}})$. But since we consider the case in which $d(x_0, X_0) \geq d(X^{\rho-1}, X_0)$, this yields

$$2d(x_0, X_0) \geq t_0 - T^{\rho-1} + 3G(C_l^{\mathsf{DTO}}) \geq 3G(C_l^{\mathsf{DTO}}),$$

as $t_0 \geq T^{\rho-1}$. This implies the claim.

Case 2: $t_0 < T^{\rho-1}$. Here we get $t_l \leq t_0 < T^{\rho-1}$ since r_l is older than s_0, as reasoned above. By (5) and Observation 4.5 (i), request s_0 is served after r_l by every $\mathsf{ADV} \in \mathcal{ADV}(T^{\rho-1})$. Hence,

$$t_0 + \alpha_0(T^{\rho-1}) \geq t_l + \alpha_l(T^{\rho-1}) + d(x_0, x_l). \tag{6}$$

From the assumption that Statement (c) holds true for phase $\rho - 1$, we know that r_l is served in time, i.e.,

$$C_l^{\mathsf{DTO}} \leq t_l + \alpha_l(T^{\rho-1}) + 3G(T^{\rho-1}), \tag{7}$$

because $T^{\rho-1}$ was the last time DTO turned around before it served r_l, and because of $t_l \leq T^{\rho-1}$. Hence, if DTO's server turned around immediately after serving r_l, it would hold that

$$\begin{aligned} C_0^{\mathsf{DTO}} &= C_l^{\mathsf{DTO}} + d(x_0, x_l) \\ &\leq t_l + \alpha_l(T^{\rho-1}) + 3G(T^{\rho-1}) + d(x_0, x_l) && \text{by (7)} \\ &\leq t_0 + \alpha_0(T^{\rho-1}) + 3G(T^{\rho-1}) && \text{by (6)} \\ &\leq t_0 + 4G(T^{\rho-1}) \leq t_0 + 2^{-a+2}G(C_l^{\mathsf{DTO}}). \end{aligned}$$

Thus, s_0 would be served with a flow time of at most $2\,G(C_l^{\mathsf{DTO}})$, because $a \geq 1$. Since DTO never plans its turning point in such a way that the target is reached with a flow time of less than three times the current guess value, we can deduce that the server does in fact not turn around, but has time of at least $(3-2^{-a+2})G(C_l^{\mathsf{DTO}}) > 0$ to spend on a detour, starting at time C_l^{DTO}. Thus, the distance between x_l and the turning point TP_0 planned at time C_l^{DTO} is at least $\frac{1}{2}(3-2^{-a+2})G(C_l^{\mathsf{DTO}})$, and we conclude that

$$\begin{aligned} d(X_0, x_0) &= d(X_0, x_l) + d(x_l, x_0) \\ &\geq \frac{1}{2}\left(3 - 2^{-a+2}\right) G(C_l^{\mathsf{DTO}}) + d(x_l, x_0) \\ &\geq \left(1 - 2^{-a}\right) G(C_l^{\mathsf{DTO}}) + G(T^{\rho-1}) && \text{by (5)} \\ &= \left(1 - 2^{-a}\right) G(C_l^{\mathsf{DTO}}) + 2^{-a} G(C_l^{\mathsf{DTO}}) \\ &\geq G(C_l^{\mathsf{DTO}}), \end{aligned}$$

which was our claim for TP_0.

Assume now that (a) holds for the turning points $\mathrm{TP}_0, \ldots, \mathrm{TP}_{i-1}$ of the considered phase ρ. We have to prove (a) for the next turning point $\mathrm{TP}_i = (T_i, X_i)$. Assume that TP_i replaces TP_{i-1} at time t of phase ρ, and let $t' < t$ be the time when TP_{i-1} was valid for the first time. Denote by $s_i = (t_i, x_i)$ the target corresponding to TP_i. Recall that we assumed w.l.o.g. that TP_i is to the right of s_i.

In the case that $s_i = s_0$, we can deduce that $G(t) \geq 2G(t')$ because the turning point changes but the target does not. That is, in order to serve the target with a flow time of $3G(t)$ instead of $3G(t')$, DTO has now at least $3(G(t) - G(t'))$ additional time units to spend on the way to the target. Hence, it can enlarge its detour by

$$d(X_i, X_{i-1}) \geq \frac{3}{2}(G(t) - G(t')) \geq G(t) - G(t').$$

By the inductive assumption, $d(X_{i-1}, x_0) \geq G(t')$. Consequently,

$$\begin{aligned} d(X_i, x_0) &= d(X_i, X_{i-1}) + d(X_{i-1}, x_0) \\ &\geq G(t) - G(t') + G(t') = G(t). \end{aligned}$$

If $s_i \neq s_0$ and the new target s_i was released later than time $T^{\rho-1}$, then independent of whether s_i is between $T^{\rho-1}$ and r_l or further away from r_l, we can conclude as in the proof for TP_0, that $d(X_i, x_i) \geq \frac{3}{2}G(t)$.

If $s_i \neq s_0$ and s_i had already been released at time $T^{\rho-1}$, it follows that s_i must have been infeasible at time C_l^{DTO} when the initial target was chosen, as s_i was more critical than s_0 at that time: otherwise it could not have become a target later on. Hence, there exists a request r_j older than s_i and to the right of s_i's hypothetical turning point $\mathrm{TP}_i' = (T_i', X_i')$ considered at time C_l^{DTO}.

By exactly the same arguments used for TP_0 above, we can deduce for the hypothetical turning point TP_i' considered at time C_l^{DTO} that

$$d(X_i', x_i) \geq G(C_l^{\mathsf{DTO}}). \tag{8}$$

If $G(C_l^{\mathsf{DTO}}) = G(t)$, we obtain that

$$d(X_i, x_i) > d(X_i', x_i) \geq G(C_l^{\mathsf{DTO}}) = G(t).$$

On the other hand, if $G(t) \geq 2G(C_l^{\mathsf{DTO}})$, DTO has at least $3(G(t) - G(C_l^{\mathsf{DTO}}))$ time units more to spend on its detour to x_i than it would have had if s_i had become the target at time C_l^{DTO}. Hence, the distance of the hypothetical turning point TP_i' considered at time C_l^{DTO} and the one chosen at time t, namely TP_i, is at least half of that additional time. More precisely, we have that

$$d(X_i', X_i) \geq \frac{3}{2}(G(t) - G(C_l^{\mathsf{DTO}})) \geq G(t) - G(C_l^{\mathsf{DTO}}).$$

Together with (8), we obtain that

$$\begin{aligned} d(X_i, x_i) &= d(X_i, X_i') + d(X_i', x_i) \\ &\geq G(t) - G(C_l^{\mathsf{DTO}}) + G(C_l^{\mathsf{DTO}}) = G(t), \end{aligned}$$

which proves the inductive step.

Notice that exactly the same arguments apply whenever a request is not made a target because it is not feasible: its hypothetical turning point considered at that time t must be at least at distance $G(t)$ to the corresponding target.

Proof of Statements (b) and (c):

Let SP denote the selection point which defines the end of the previous phase. If no previous phase exists, we define SP $= (0, 0)$. By definition, SP $= (C_l^{\mathsf{DTO}}, x_l)$ for some request r_l. We distinguish two cases: in Case I we consider the situation that DTO's server immediately turns around in the selection point, thus not entering the detour mode, in Case II, we assume that it enters the detour mode at the selection point. Furthermore, we partition the set of requests served in phase ρ into three classes, depending on which part of DTO's route they are served in: Class 1 contains all requests served between the realized turning point and its corresponding target. Class 2 consists of those requests served in phase ρ after the target; whereas all requests served between the selection point and the realized turning point belong to Class 3 (see Figure 1).

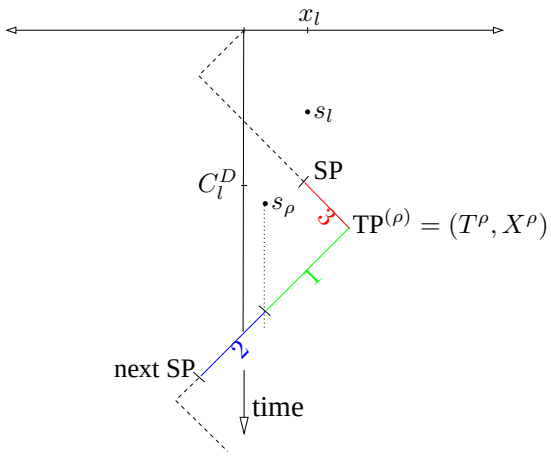

Figure 1: Classification of the requests served in phase ρ.

Let r_z be the oldest unserved request at time C_l^{DTO}. Denote by TP$_0 = (T_0, X_0)$ be the turning point chosen at time C_l^{DTO}, and by $s_0 = (t_0, x_0)$ its corresponding target.

Case I: DTO turns around in the selection point.

1. Phase ρ is the first phase of a busy period.

Hence, all requests served in that phase are released at time t_0 or later, and since DTO immediately enters the focus mode at the beginning of the phase, they are all served without any detour. By 4.6, we know that $d(p^{\mathsf{DTO}}(t_0), p^{\mathsf{ADV}}(t_0)) \leq \frac{5}{2}G(t_0)$. Therefore, DTO reaches all requests served in phase 1 at most $\frac{5}{2}G(t_0)$ time units later than the adversary, so all requests are served in time.

2. Phase ρ is not the first phase of a busy period.

We have TP$_0$ = SP $= (C_l^{\mathsf{DTO}}, x_l)$, because the server turns around immediately as it cannot serve s_0 with a flow time of $3G(C_l^{\mathsf{DTO}})$ or less. Thus, TP$_0$ equals the realized turning point TP$^\rho = (T^\rho, X^\rho)$, request s_0 is the final target and $C_0^{\mathsf{DTO}} = C_l^{\mathsf{DTO}} + d(x_0, x_l)$. Note that Class 3 is empty in this case. First of all, we know that s_0 cannot be older than r_l: If s_0 was older, DTO would not have served r_l anymore, as it only remains in the focus mode until the oldest unserved request is in its back. Furthermore, by part (a) it holds that $d(x_l, x_0) = d(X^\rho, x_0) \geq G(T^\rho) = G(C_l^{\mathsf{DTO}})$.

Therefore, by Observation 4.5 (i), in all offline solutions in $\mathcal{ADV}(T^\rho)$, request r_l must be served before s_0. It is easy to see that $\tau_l \leq \tau_0 = T^\rho$, and since Statement (c) for phase $\rho - 1$ tells us that r_l is served in time, we can apply Lemma 4.4 and conclude that s_0 is also served in time. Notice that exactly the same arguments apply to all requests in Class 2. This ensures Statement (c).

It remains to consider all other requests of Class 1. To this end, let r_j be a request served between r_l and s_0 by DTO. If r_j was more critical than s_0, it must be at least as old as s_0, because it is to the right of s_0. Since r_j was not chosen as target at time C_l^{DTO}, it must have been infeasible, that is, there is a request r_b older than r_j and to the right of r_j's hypothetical turning point TP$'_j$. But as r_j is more critical than s_0, and DTO turns around immediately for s_0, we have that TP$'_j$ = TP$^\rho$, which implies that also s_0 cannot have been feasible at time C_l^{DTO}, a contradiction. Thus, r_j must be less critical than s_0. This means that it is served with the same or a smaller flow time than s_0, hence with flow time of at most $4G(T^\rho) \leq 4G(C_j^{\mathsf{DTO}})$. This proves Case I.

Case II: DTO enters the detour mode at the selection point.

The proof of this case is omitted due to lack of space and can be found in the appendix. The key arguments

used are similar to the ones in Case I, but more involved: We first show that all requests ever marked as a target, excluding the final target (set S), are served with a flow time at most $3G(T^\rho)$. This lets us conclude that requests which are less critical than those in S are served with smaller flow times. In order to show that other requests r_j are served with the desired flow time, we apply Lemma 4.4 with a careful choice of the request r_i which is served before r_j by ADV. Observation 4.5 will be used to determine a suitable r_i. Another helpful ingredient is the following: if $d(p^{\mathsf{DTO}}(T^\rho), p^{\mathsf{ADV}}(T^\rho)) \leq 3G(T^\rho)$, then all requests served by both servers after time T^ρ are served in time ($\mathsf{ADV} \in \mathcal{ADV}(T^\rho)$). □

Theorem 4.8 DTO *is* 8*-competitive against a non-abusive adversary for the* $F_{\max}$-OLTSP.

Proof. By Theorem 4.7 we have that any request r_i is served with flow time at most $4G(C_i^{\mathsf{DTO}})$. If $C_{\text{last}}^{\mathsf{DTO}}$ is the time at which the last request is served by DTO, all requests are served with flow time at most $4G(C_{\text{last}}^{\mathsf{DTO}})$, which, by construction, is bounded by $2\mathsf{OPT}(\sigma)$, thence the claim. □

References

[1] N. ASCHEUER, S. O. KRUMKE, AND J. RAMBAU, *Online dial-a-ride problems: Minimizing the completion time*, in Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, vol. 1770 of Lecture Notes in Computer Science, Springer, 2000, pp. 639–650.

[2] G. AUSIELLO, E. FEUERSTEIN, S. LEONARDI, L. STOUGIE, AND M. TALAMO, *Algorithms for the on-line traveling salesman*, Algorithmica, 29 (2001), pp. 560–581.

[3] M. BLOM, S. O. KRUMKE, W. E. DE PAEPE, AND L. STOUGIE, *The online-TSP against fair adversaries*, Informs Journal on Computing, 13 (2001), pp. 138–148. A preliminary version appeared in the Proceedings of the 4th Italian Conference on Algorithms and Complexity, 2000, vol. 1767 of Lecture Notes in Computer Science.

[4] A. BORODIN AND R. EL-YANIV, *Online Computation and Competitive Analysis*, Cambridge University Press, 1998.

[5] E. FEUERSTEIN AND L. STOUGIE, *On-line single server dial-a-ride problems*, Theoretical Computer Science, (2001). To appear.

[6] D. HAUPTMEIER, S. O. KRUMKE, AND J. RAMBAU, *The online dial-a-ride problem under reasonable load*, Theoretical Computer Science, (2001). A preliminary version appeared in the Proceedings of the 4th Italian Conference on Algorithms and Complexity, 2000, vol. 1767 of Lecture Notes in Computer Science.

[7] H. KELLERER, T. TAUTENHAHN, AND G. J. WOEGINGER, *Approximability and nonapproximability results for minimizing total flow time on a single machine*, in Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, 1996, pp. 418–426.

[8] E. KOUTSOUPIAS AND C. PAPADIMITRIOU, *Beyond competitive analysis*, in Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science, 1994, pp. 394–400.

[9] S. O. KRUMKE, W. E. DE PAEPE, D. POENSGEN, AND L. STOUGIE, *News from the online traveling repairman*, in Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science, vol. 2136 of Lecture Notes in Computer Science, 2001, pp. 487–499.

[10] R. MOTWANI AND P. RAGHAVAN, *Randomized Algorithms*, Cambridge University Press, 1995.

[11] K. PRUHS AND B. KALYANASUNDARAM, *Speed is as powerful as clairvoyance*, in Proceedings of the 36th Annual IEEE Symposium on the Foundations of Computer Science, 1995, pp. 214–221.

A Proof of Theorem 4.7 (b) and (c), Case II

Case II: DTO enters the detour mode at the selection point.

Since in this case, the proof is in large parts the same for whether phase ρ is the first phase of a busy period or not, we only make the distinction when needed.

We start with the requests in Class 1. Let us first consider an arbitrary request s_i which was ever marked as target during the current phase ρ but did not become the final target. We prove the stronger statement that s_i is served with a flow time of at most $3G(T^\rho)$. Consider the time T^ρ at which DTO reverses direction. If s_i was still the target at T^ρ, and $\text{TP}_i = (T_i, X_i)$ the corresponding turning point valid at that time, s_i would be served with a flow time of $3G(T^\rho)$. As s_i was not the target at time T^ρ anymore, the then valid target s^ρ must be more critical than s_i. Hence, the realized turning point $\text{TP}^\rho = (T^\rho, X^\rho)$ must be closer to the selection point $\text{SP} = (C_l^{\mathsf{DTO}}, x_l)$ than the point $\text{TP}_i = (T_i, X_i)$ defined above, as DTO must turn around earlier for the more critical request s^ρ. Consequently, s_i is reached even earlier than in the case that it was not replaced. We can conclude that in both cases, s_i is served with a flow time of at most $3G(T^\rho)$.

Conclusion 1 *Let S be the set of all requests which were ever marked as target during the current phase except for the final target. All requests in S are served with a flow time at most $3G(T^\rho)$.* □

From this we can conclude that the oldest request r_z is also served with a flow time of at most $3G(T^\rho)$, since it is less critical than the initial target s_0 of this phase: if it was more critical, it would have been selected as target at time C_l^{DTO} instead of s_0 (as the oldest unserved request overall r_z cannot be infeasible).

Conclusion 2 *The oldest unserved request r_z is served with a flow time at most $3G(T^\rho)$.* □

Before we consider the final target and requests from Class 1 which are less critical than the final target, let us show that any request $r_i = (t_i, x_i)$ in Class 1 which is more critical than the final target is served in time. As a member of Class 1, request r_i must lie between the turning point and the final target. Since it is also more critical than s^ρ, it must be older than s^ρ. Consider the time $t \le T^\rho$ at which the candidate setup was performed last during phase ρ. By definition, s^ρ was either made a target at time t or it remained the target valid at that time. Hence, $s^\rho \in \vee(s_0, p^{\mathsf{DTO}}(C_l^{\mathsf{DTO}}), G(t))$, the critical region valid at time t. Since r_i is more critical than s^ρ and also closer to the selection point, it must also be inside $\vee(s_0, p^{\mathsf{DTO}}(C_l^{\mathsf{DTO}}), G(t))$. Hence, the only reason why r_i was not made a target at time t is that it was infeasible. Notice that this also holds true if $t = C_l^{\mathsf{DTO}}$ and no critical region had yet been defined when s^ρ was marked as a target.

As no further candidate setup takes place, the guess value does not change anymore after time t, so r_i is still infeasible at time T^ρ. This means that there exists a request $r_b = (t_b, x_b)$ between the hypothetical turning point $\text{TP}'_i = (T'_i, X'_i)$ corresponding to r_i and the actually chosen turning point TP^ρ, which is older than r_i. By Part (a), this implies that $d(X'_i, x_i) \ge G(T^\rho)$. Hence, as r_b is even further away from x'_i than X'_i, we deduce that $d(x_b, x_i) \ge G(T^\rho)$, which by Observation 4.5 (i) implies that r_b must be served before r_i in any offline solution in $\mathcal{ADV}(T^\rho)$. Observe that the oldest request r_z must lie between r_b and r_i as r_i is more critical and younger than r_z. Hence, r_i is served after r_z in any such offline solution. Clearly, $C_i^{\mathsf{DTO}} = C_z^{\mathsf{DTO}} + d(x_z, x_i)$, and as $t_i \le T^\rho$, we have that $\tau_i = \tau_z = T^\rho$. Hence, we can apply Lemma 4.4 to deduce from Conclusion 2 that also r_i is served in time.

Conclusion 3 *All requests r_i of Class 1 which are more critical than the final target are served in time.*

Before continuing the proof for Class 1, let us briefly consider a subclass of Class 2. To this end, let r_j be a request of Class 2 which is released by time T^ρ and more critical than the final target s^ρ. Again, consider the last time $t \le T^\rho$ at which a candidate setup was performed. As above, we need to investigate why r_j was not made a target at time t. In this case, it was either infeasible or outside the critical region $\vee(s_0, p^{\mathsf{DTO}}(C_l^{\mathsf{DTO}}), G(t))$ valid at time t. If it was infeasible, the same argument as above proves that r_j is served in time (note that again, $\tau_j = \tau_z = T^\rho$). It remains to consider the case that r_j is outside $\vee(s_0, p^{\mathsf{DTO}}(C_l^{\mathsf{DTO}}), G(t))$. By definition of t,

we have that $\vee(s_0, p^{\mathsf{DTO}}(C_l^{\mathsf{DTO}}), G(t))$ is still valid at time T^ρ, so by Observation 4.5 (ii), r_j must be served after s_0 in all offline solutions in $\mathcal{ADV}(T^\rho)$. Since DTO serves r_j immediately after s_0, Conclusion 1 and the fact that $\tau_0 = \tau_j = T^\rho$ allow us to apply Lemma 4.4 and deduce that r_j is served in time.

Conclusion 4 *All requests r_j of class 2 which are more critical than the final target and released no later than T^ρ are served in time.*

Now consider the final target s^ρ of the phase. Clearly, if DTO does not turn around immediately when marking s^ρ as a target and switch to the focus mode, s^ρ is served with flow time $3G(T^\rho)$.

So assume that DTO enters the focus mode at the time it marks s^ρ as target. We need to distinguish two subcases: (i) $t^\rho < T^\rho$, and (ii) $t^\rho = T^\rho$.

Consider first case (i). Let $t' < T^\rho$ be the last time before T^ρ at which a candidate setup was performed. Hence, $t^\rho \leq t'$. Let $\mathrm{TP}' = (T', X')$ be the turning point chosen at time t'. Since s^ρ was not made a target at time t', it was either infeasible at time t' or feasible but outside the critical region valid at time t'.

If it was infeasible, there must be a request $r_b = (t_b, x_b)$ older than s^ρ yet unserved at time t' and which lies to the right of s^ρ's hypothetical turning point considered at time t'. Since a target candidate setup is also performed whenever DTO serves a request while in the detour mode, form the definition of t' and by the assumption that r_b is yet unserved at time t', it follows that $C_b^{\mathsf{DTO}} \geq T^\rho$. On the other hand, we assumed that the time T^ρ at which DTO turns around is the time at which it marks s^ρ as a target. Since s^ρ must be feasible at that time, request r_b must have been served by then, which means that $C_b^{\mathsf{DTO}} \leq T^\rho$. Hence, we obtain that $(C_b^{\mathsf{DTO}}, x_b) = (T^\rho, X^\rho)$. From part (a) it follows that $d(x_b, x^\rho) \geq G(T^\rho)$, so by Observation 4.5 (i), we have that s^ρ is served after r_b in all offline solutions in $\mathcal{ADV}(T^\rho)$. But then, s^ρ must be served after r_z in all such offline solutions as well, since r_z lies between r_b and s^ρ and is older than both. Since $\tau^\rho = \tau_z = T^\rho$, Conclusion 2 and Lemma 4.4 let us deduce that s^ρ is served in time in that case.

We now consider the case that s^ρ was feasible but outside the critical region at time t'. Since s^ρ was marked as target at time T^ρ, it was inside the critical region valid at time T^ρ, and we can deduce that $2G(t') \leq G(T^\rho)$. Now since s^ρ was feasible at time t', it would have satisfied the preconditions of Conclusion 3 or 4 if the sequence had ended at time t'. Consequently, the turning point TP′ chosen at time t' was chosen in such way that s^ρ would be served with a flow time of at most $4G(t')$.

Thus, it holds for all times $t \in [t', T^\rho)$ that

$$t + d(p^{\mathsf{DTO}}(t), X') + d(X', x^\rho) \leq t^\rho + 4G(t').$$

This implies that

$$T^\rho + d(p^{\mathsf{DTO}}(T^\rho), x^\rho) \leq t^\rho + 4G(t') \leq t^\rho + 2G(T^\rho),$$

contradicting the assumption that DTO had to turn around immediately at time t^ρ. Notice that we showed that in the case that $t^\rho < T^\rho$, the final target s^ρ must have been infeasible at the last time $t' < T^\rho$ at which a target candidate setup was performed.

It remains to consider case (ii), where s^ρ is made a target as soon as it is released: $t^\rho = T^\rho$.

Let us first investigate the length of the detour made by DTO. I.e., we first prove a bound on $d(x_l, X^\rho) = T^\rho - C_l^{\mathsf{DTO}} = t^\rho - C_l^{\mathsf{DTO}}$. To this end, we make use of the assumption that DTO can not serve s^ρ with a flow time of $3G(T^\rho)$. Hence, $d(X^\rho, x^\rho) > 3G(T^\rho)$. On the other hand, $d(x^\rho, x_0) \leq G(T^\rho) - (t^\rho - t_0)$ since s^ρ is inside the critical region valid at time T^ρ and younger than s_0. Putting the two inequalities together, we obtain

$$d(x_0, X^\rho) = d(x^\rho, X^\rho) - d(x^\rho, x_0) > 2G(T^\rho) + t^\rho - t_0. \tag{9}$$

Making use of the fact that DTO serves s_0 with a flow time of at most $3G(T^\rho)$ (Conclusion 1), we have

$$C_l^{\mathsf{DTO}} + d(x_l, X^\rho) + d(X^\rho, x_0) \leq t_0 + 3G(T^\rho).$$

Therefore,

$$\begin{aligned}
d(x_l, X^\rho) &\leq t_0 + 3G(T^\rho) - C_l^{\mathsf{DTO}} \\
&\quad - d(X^\rho, x_0) \\
&< t_0 + 3G(T^\rho) - C_l^{\mathsf{DTO}} \\
&\quad - 2G(T^\rho) - t^\rho + t_0 && \text{by (9)} \\
&= G(T^\rho) + (t_0 - C_l^{\mathsf{DTO}}) - t^\rho + t_0 \\
&\leq G(T^\rho) - t^\rho + C_l^{\mathsf{DTO}} && \text{as } t_0 \leq C_l^{\mathsf{DTO}} \\
&= G(T^\rho) - d(x_l, X^\rho).
\end{aligned}$$

We thus obtain that

$$d(x_l, X^\rho) = T^\rho - C_l^{\mathsf{DTO}} \leq \frac{1}{2} G(T^\rho). \qquad (10)$$

Recall that S is the set of requests which contains all requests in Class 1 that were ever marked as target, except for s^ρ itself. We showed before that each request in $S \cup \{r_z\}$ is served with a flow time of at most $3G(T^\rho)$ (Conclusions 1 and 2), in particular in time. Furthermore, each such request has been released before time T^ρ, which is the last time DTO reverses direction before serving that request. Hence, if for every offline solution in $\mathcal{ADV}(T^\rho)$ there exists a request from $S \cup \{r_z\}$ which is served before s^ρ, then Lemma 4.4 yields that s^ρ is served in time.

Now consider an arbitrary, but fixed $\mathsf{ADV} \in \mathcal{ADV}(T^\rho)$ in which s^ρ is served before all requests in $S \cup \{r_z\}$. Recall that we assumed that the turning point's position X^ρ is to the right of x^ρ. At time T^ρ, ADV's server must be located left of all requests in $S \cup \{r_z\}$, and all requests in $S \cup \{r_z\}$ are yet unserved by ADV. Our aim is to show for the respective completions times of s^ρ that

$$C_\rho^{\mathsf{DTO}} \leq C_\rho^{\mathsf{ADV}} + 3G(T^\rho) \qquad (11)$$

holds. Alas, in one subcase we will only prove the weaker claim $C_\rho^{\mathsf{DTO}} \leq t^\rho + 4G(T^\rho)$. However, we will deduce in that subcase that there is a request in Class 2 which is served in time by DTO. In all other cases, (11) holds, i.e., the final target is reached by DTO no later than $3G(T^\rho)$ time units after the adversary reaches it in the considered offline solution, and as we considered an arbitrary $\mathsf{ADV} \in \mathcal{ADV}(T^\rho)$, we can deduce that DTO serves s^ρ in time. Hence if Class 2 is empty, (11) yields statement (c).

Consider $p^{\mathsf{ADV}}(T^\rho)$, which by our assumption is further to the left than any point in $S \cup \{r_z\}$. Since the adversary is non-abusive, there must be a request r_k left of its position at time T^ρ which it just served or which it is heading to. This request r_k must have been released strictly before time T^ρ, i.e., $t_k < T^\rho$. Another case distinction is needed.

- r_k was already served by DTO by time T^ρ.

  Notice that this situation can not occur if we are in the first phase of the first busy period. Hence we may assume that $\rho \geq 2$.

  We show that in this case,

  $$d(p^{\mathsf{ADV}}(T^\rho), x_l) \leq \frac{5}{2} G(T^\rho). \qquad (12)$$

  This, together with (10) then yields $d(p^{\mathsf{DTO}}(T^\rho), p^{\mathsf{ADV}}(T^\rho)) \leq 3G(T^\rho)$, from which (11) easily follows, as DTO immediately heads to serve s^ρ at time T^ρ, while ADV cannot proceed to serve $s^\rho = (t^\rho, x^\rho)$ before $t^\rho = T^\rho$.

  By assumption, r_k is served before r_l by DTO, and by Statement (b) for phase $\rho - 1$ applied to r_l we have the inequality

  $$t_k + d(x_k, x_l) \leq t_l + 4G(C_l^{\mathsf{DTO}}) \leq t_l + 4G(T^\rho). \qquad (13)$$

  If the adversary serves r_k after r_l, then

  $$t_l + d(x_k, x_l) \leq t_k + G(T^\rho),$$

  and together with (13), we obtain

  $$\begin{aligned} d(x_k, x_l) &\leq t_k + G(T^\rho) - t_l \\ &\leq t_l + 4G(T^\rho) - d(x_k, x_l) + G(T^\rho) - t_l \\ &= 5G(T^\rho) - d(x_k, x_l). \end{aligned}$$

  This yields (12).

  Now consider the case that ADV serves r_k before r_l. Since the adversary is heading to or just coming from r_k at time T^ρ and is still on the left of the set S, this means that it hasn't served r_l yet at time $T^\rho \geq C_l^{\mathsf{DTO}} \geq t_l$. Hence, at time T^ρ, its server must be within range $G(T^\rho)$ from x_l, so in particular (12) holds.

  Notice that, in the previous line of reasoning, we did neither make use of the assumption that DTO serves the final target with flow time more than $3G(T^\rho)$, nor that $t^\rho = T^\rho$.

- r_k has not been served yet by DTO at time T^ρ.

  Recall that we are in the situation that the final target s^ρ was made a target by DTO at its release time $t^\rho = T^\rho$, and that the server turns around immediately at that time. Let $t' < T^\rho$ be the last time before time T^ρ at which a target candidate setup is performed by DTO. As $t_k < T^\rho$, and since a target

candidate setup is performed whenever a new request is released, we have that $t_k \leq t'$. Note that request r_k must be served by DTO in the current phase ρ. If it wasn't served in phase ρ, there would be an older request which remains unserved at least until time T^ρ and which is in the server's back after it turned in TP^ρ. But then, this request must be older than s^ρ and would have caused s^ρ to be infeasible.

Let s_m be the target valid at time t' (after the target selection), and TP_m the corresponding turning point. In the proof for Conclusions 1 and 2 we showed that the oldest unserved request r_z, and all requests in S are served with a flow time of at most $3G(t')$ if TP_m is also the realized turning point. We are in the situation that TP_m is replaced at time T^ρ by TP^ρ. But as DTO turns around immediately when replacing TP_m, it turns earlier than planned and hence serves the requests in $S \cup \{r_z\}$ are even served earlier, in particular with a flow time of at most $3G(t')$.

Now consider request r_k. There are three possible reasons why r_k is not selected as target at time t': (i) it is less critical than s_m, (ii) r_k is more critical than s_m but infeasible at time t', or (iii) it is more critical than s_m but outside the critical region valid at that time.

In case (i), i.e. if r_k is less critical than s_m, it is also served with a flow time of at most $3G(t') \leq 3G(T^\rho)$, which yields in particular $C_k^{\mathsf{DTO}} \leq C_k^{\mathsf{ADV}} + 3G(T^\rho)$ for the considered $\mathsf{ADV} \in \mathcal{ADV}(T^\rho)$. Furthermore, s^ρ is more critical than r_k and younger. Therefore, it must be to the left of r_k, which in turn was left of $p^{\mathsf{ADV}}(T^\rho)$. Therefore, s^ρ is served after r_k by ADV, and we conclude

$$\begin{aligned} C_\rho^{\mathsf{DTO}} &= C_k^{\mathsf{DTO}} + d(x^\rho, x_k) \\ &\leq C_k^{\mathsf{ADV}} + 3G(T^\rho) + d(x_k, x^\rho) \\ &\leq C_\rho^{\mathsf{ADV}} + 3G(T^\rho), \end{aligned}$$

which was our claim (11).

Now let us investigate case (ii), in which r_k is infeasible at time t'. That means that there exists a request $r_b = (t_b, x_b)$ older than r_k and to the right of the hypothetical turning point corresponding to r_k at time t'. If ADV served r_b before time T^ρ, it must have served r_z on its way from r_b to its current position, as r_z is older than r_b and located between r_b and $p^{\mathsf{ADV}}(T^\rho)$. This contradicts our assumption that ADV has not served any of the requests in $S \cup \{r_z\}$ by time T^ρ. Consequently, r_b must be yet unserved by ADV at time T^ρ, which yields $d(p^{\mathsf{ADV}}(T^\rho), x_b) \leq G(T^\rho)$. Since r_b must be to the right of the selection point $(C_l^{\mathsf{DTO}}, x_l)$, we obtain in particular that $d(p^{\mathsf{ADV}}(T^\rho), x_l) \leq G(T^\rho)$, which together with (10) implies

$$d(p^{\mathsf{ADV}}(T^\rho), p^{\mathsf{DTO}}(T^\rho)) \leq \frac{3}{2} G(T^\rho).$$

Hence, DTO reaches s^ρ no more $\frac{3}{2}G(T^\rho)$ time units later than ADV, which means that s^ρ is served in time.

Finally, consider Case (iii): request r_k is outside the critical region valid at time t'. Consequently, r_k is served after s_0 in all offline solutions in $\mathcal{ADV}(t')$. Recall that we showed before that in the current case, all requests in $S \cup \{r_z\}$ are served with a flow time of at most $3G(t')$. Hence, we obtain that

$$\begin{aligned} T^\rho + d(p^{\mathsf{DTO}}(T^\rho), x_k) = C_k^{\mathsf{DTO}} &= C_0^{\mathsf{DTO}} + d(x_0, x_k) \\ &\leq t_0 + 3G(t') + d(x_0, x_k) \\ &\leq t_k + \alpha_k(t') + 3G(t'). \end{aligned}$$

If $2G(t') \leq G(T^\rho)$, we obtain together with $t_k < T^\rho$ that

$$d(p^{\mathsf{DTO}}(T^\rho), x_k) \leq 4G(t') \leq 2G(T^\rho),$$

which lets us conclude that $d(p^{\mathsf{DTO}}(T^\rho), p^{\mathsf{ADV}}(T^\rho)) \leq 2G(T^\rho)$. Thus, DTO serves s^ρ at most $2G(T^\rho)$ time units later than ADV.

If $G(t') = G(T^\rho)$, we obtain by Property 4.2 that

$$\alpha_k(t') \leq \alpha_k(T^\rho).$$

Thus, we have that r_k is served in time, since $\tau_k = T^\rho$. If s^ρ is served after r_k by ADV, we conclude with Lemma 4.4 that (11) holds. Otherwise, s^ρ must be to the right of r_k, is therefore, as the younger one, less critical than r_k and served with a flow time of at most $4G(T^\rho)$ by DTO.

Note that we proved the stronger statement that s^ρ is served in time for all cases except for the case that all of the following statements hold simultaneously true:

- $t^\rho = T^\rho$ and DTO turns around immediately upon s^ρ's release,
- there exists ADV $\in \mathcal{ADV}(T^\rho)$ in which s^ρ is served before all requests in $S \cup \{r_z\}$,
- there is a request r_k to the left of $p^{\text{ADV}}(T^\rho)$ with $t_k < T^\rho$ which is outside the critical region at the last time $t' < T^\rho$ at which a candidate setup was performed by DTO,
- r_k is served after s^ρ by ADV.

In that special case, s^ρ is shown to be served with flow time $4G(T^\rho)$, and r_k is shown to be served in time by DTO.

Conclusion 5 *The final target s^ρ is served either in time, or it is served with a flow time of at most $4G(T^\rho)$ and there exists a request r_k in Class 2 which is released before T^ρ and which is served in time by* DTO.

Note that this implies Statement (c) for the case that Class 2 is empty.

Finally, let $r_i = (t_i, x_i)$ be an arbitrary request of Class 1 which was never marked as target and which is less critical than the final target s^ρ. As it is less critical, it is served with the same or a smaller flow time than s^ρ, which is at most $4G(T^\rho) \leq 4G(C_i^{\text{DTO}})$, which was our claim.

Conclusion 6 *Any request r_i in Class 1 which does not belong to any of the sets of requests covered by Conclusions 1–3 or by Conclusion 5 is served with flow time at most $4G(T^\rho)$.*

We now consider Class 2. To this end, let $r_j = (t_j, x_j)$ be a request served in phase ρ after s^ρ by DTO. First consider the case that r_j is released no later than time T^ρ, that is, $t_j \leq T^\rho$. We proved already that r_j is served in time if it is more critical than the final target s^ρ (Conclusion 4). Consider the case that r_j is less critical than s^ρ. If s^ρ was served with a flow time of $3G(T^\rho)$, then also r_j is served with that flow time, hence in time. So assume that s^ρ is served with a bigger flow time. Since r_j is less critical and to the left of s^ρ, it must be strictly younger. Consequently, $t^\rho < t_j \leq T^\rho$. We showed before that in this case, s^ρ must have been infeasible at time t', the last time before T^ρ at which a candidate setup was performed, and that there exists a request r_b older than s^ρ for which $(C_b^{\text{DTO}}, x_b) = (T^\rho, X^\rho)$. By part (a), we deduce that

$$d(x_b, x_j) \geq d(x_b, x^\rho) = d(X^\rho, x^\rho) \geq G(T^\rho).$$

As r_b is older than r_j, Observation 4.5 (i) implies that it must be served before r_j in all offline solutions in $\mathcal{ADV}(T^\rho)$. As reasoned before, then also r_z must be served before r_j in all such offline solutions. Since DTO serves r_j immediately after r_z and as $\tau_z = \tau_j = T^\rho$, we can apply Lemma 4.4 to conclude that r_j is served in time.

It remains to consider those requests $r_j \in$ Class 2 for which $t_j > T^\rho$. Note that $\tau_j = t_j > T^\rho$ in this case. Let

$$A := \{r_z\} \cup S \cup \{r_i \in \text{Class 2} : t_i \leq T^\rho\}.$$

It is easy to see that each request $r_a \in A$ is released by time T^ρ, so we have that $\tau_a = T^\rho < \tau_j$. Furthermore, we showed before that all requests in A are served in time by DTO. Consider an arbitrary ADV $\in \mathcal{ADV}(\tau_j)$. Assume that there exists a request $r_a \in A$ which is served by ADV before r_j. No matter whether x_j is left of x_a or not, we have that $C_j^{\text{DTO}} \leq C_a^{\text{DTO}} + d(x_j, x_a)$. Hence, we can apply Lemma 4.4 in that case and deduce that r_j is served in time.

Therefore, we can restrict our attention to the case that ADV serves all requests in A after r_j. In particular, this means that is has not served any of the requests in A at at time T^ρ yet. We distinguish two subcases: (i) there is a request $r_a \in A$ which is left of $p^{\text{ADV}}(T^\rho)$, and (ii) ADV's position at time T^ρ is to the left of all requests in A.

In case (i), r_j cannot be to the left of r_a, since otherwise it would be served after r_a by ADV, contradicting our assumption in this case. Hence, it suffices to show that r_j is served with flow time at most $4G(C_j^{\text{DTO}})$, because it cannot be the last request served by DTO in the current phase. As an element of the set A, request r_a is served latest at time $t_a + 4G(T^\rho)$. Since r_a was released before T^ρ, we know that $t_a < t_j$, and because DTO serves r_j on the way to r_a, we have that

$$\begin{aligned} C_j^{\text{DTO}} \leq C_a^{\text{DTO}} &\leq t_a + 4G(T^\rho) \\ &\leq t_j + 4G(T^\rho) \leq t_j + 4G(C_j^{\text{DTO}}), \end{aligned}$$

which was our claim.

Now consider case (ii): ADV's position at time T^ρ is to the left of all requests in A. Since the adversary is non-abusive, there must be a request r_k to its left which it just

served or where it is heading to at time T^ρ. In particular, it holds that $t_k < T^\rho$, from which we can deduce that r_k must have been served already by DTO at time T^ρ: If it was served after T^ρ in phase ρ, it would belong to the set A, contradicting that it is left of ADV which in turn is left of all requests in A. Furthermore, r_k cannot be served by DTO in a later phase: if it was, there would be an older request in the server's back, and DTO would have to turn around before reaching r_j, thus serving r_j also in a later phase, contradicting the assumption that r_j is served in the current phase.

Exactly as in the proof of inequality (12) (used for the final target), we can in this case deduce for the distance of ADV's position at time T^ρ to the selection point $\mathrm{SP} = (C_l^{\mathsf{DTO}}, x_l)$ that

$$d(p^{\mathsf{ADV}}(T^\rho), x_l) \leq \frac{5}{2}\, G(T^\rho). \tag{14}$$

Let $L := d(X^\rho, x_l) = T^\rho - C_l^{\mathsf{DTO}}$ be the length of the detour made by DTO at the beginning of the current phase. If $L \leq G(T^\rho)/2$, we obtain from (14) that

$$d(p^{\mathsf{ADV}}(T^\rho), p^{\mathsf{DTO}}(T^\rho)) \leq 3G(T^\rho),$$

which implies that r_j is served in time as ADV cannot serve r_j before time $t_j > T^\rho$, and since DTO proceeds towards r_j without any detour after time T^ρ.

So assume that $L = T^\rho - C_l^{\mathsf{DTO}} > G(T^\rho)/2$. Since DTO serves r_z with a flow time of at most $3G(T^\rho)$, we have that

$$C_l^{\mathsf{DTO}} + L + d(X^\rho, x_z) \leq t_z + 3G(T^\rho),$$

which implies

$$d(X^\rho, x_z) \leq t_z + 3G(T^\rho) - L - C_l^{\mathsf{DTO}}. \tag{15}$$

Furthermore, by assumption, ADV serves $r_z \in A$ after r_j, and we have that

$$T^\rho + d(p^{\mathsf{ADV}}(T^\rho), x_j) + d(x_j, x_z) \leq t_z + G(T^\rho),$$

so in particular,

$$T^\rho + d(x_j, x_z) \leq t_z + G(T^\rho). \tag{16}$$

Note that $t_z \leq C_l^{\mathsf{DTO}} \leq T^\rho < t_j$. We obtain the following estimate on DTO's completion time for r_j.

$$\begin{aligned}
C_j^{\mathsf{DTO}} &= T^\rho + d(X^\rho, x_z) + d(x_z, x_j) \\
&\leq t_z + 3G(T^\rho) - L - C_l^{\mathsf{DTO}} + t_z + G(T^\rho) \\
&\quad \text{by (15) and (16)} \\
&= t_j + 3G(T^\rho) + [G(T^\rho) + 2t_z - C_l^{\mathsf{DTO}} - t_j - L] \\
&\leq t_j + 3G(T^\rho) + [G(T^\rho) + C_l^{\mathsf{DTO}} - T^\rho - L] \\
&\quad \text{as } t_z \leq C_l^{\mathsf{DTO}} \text{ and } t_j \geq T^\rho \\
&= t_j + 3G(T^\rho) + [G(T^\rho) - 2L] \\
&\quad \text{as } L = T^\rho - C_l^{\mathsf{DTO}} \\
&< t_j + 3G(T^\rho) \\
&\quad \text{by the assumption that } L > G(T^\rho)/2.
\end{aligned}$$

Hence, in this case, r_j is even served with a flow time of at most $3G(T^\rho)$, in particular in time.

Conclusion 7 *Each request r_j in Class 2 is either served in time, or there exists a request r_k served later in the phase which is served in time, while r_j is served with a flow time of $4G(C_j^{\mathsf{DTO}})$.*

Note that this implies Statement (c) for Case II if Class 2 is non-empty.

Finally, we consider the requests in Class 3: Let r_k be served between SP and TP^ρ. Since the oldest request r_z lies in the server's back at time C_l^{DTO}, request r_k is younger than r_z. At the time C_k^{DTO} at which r_k is served, DTO is either in the detour mode and has a valid turning point TP_m, or it has already turned in TP^ρ. In the first case, r_z would be served latest at time $t_z + 3G(C_k^{\mathsf{DTO}})$ if TP_m wasn't replaced, as shown before. Therefore, it must hold that

$$\begin{aligned}
C_k^{\mathsf{DTO}} + d(x_k, \mathrm{TP}_m) + d(\mathrm{TP}_m, x_z) &\leq t_z + 3G(C_k^{\mathsf{DTO}}) \\
&\leq t_k + 3G(C_k^{\mathsf{DTO}}).
\end{aligned}$$

In the second case, $C_k^{\mathsf{DTO}} \geq T^\rho$, and as r_z is served with flow time at most $3G(T^\rho)$, we conclude that $C_k^{\mathsf{DTO}} + d(x_k, x_z) \leq t_z + 3G(T^\rho) \leq t_k + 3G(C_k^{\mathsf{DTO}})$.

Conclusion 8 *Each request r_k in Class 3 is served with flow time at most $3G(C_k^{\mathsf{DTO}})$.*

This completes the proof of Theorem 4.7. □