

Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MICHIEL BLOM

SVEN O. KRUMKE
LEEN STOUGIE

WILLEM DE PAEPE

The Online-TSP Against Fair Adversaries

THE ONLINE-TSP AGAINST FAIR ADVERSARIES

ABSTRACT. In the online traveling salesman problem requests for visits to cities (points in a metric space) arrive online while the salesman is traveling. The salesman moves at no more than unit speed and starts and ends his work at a designated origin. The objective is to find a routing for the salesman which finishes as early as possible.

Performance of algorithms is measured through their competitive ratio, comparing the outcome of the algorithms with that of an adversary who provides the problem instance and therefore is able to achieve the optimal offline solution. Objections against such omnipotent adversaries have lead us to devise an adversary that is in a natural way, in the context of routing problems, more restricted in power.

For the exposition we consider the online traveling salesman problem on the metric space given by \mathbb{R}_0^+ , the non-negative part of the real line. We show that a very natural strategy is $3/2$ -competitive against the conventional adversary, which matches the lower bound on competitive ratios achievable for algorithms for this problem.

Against the more “*fair adversary*”, that we propose, we show that there exists an algorithm with competitive ratio $\frac{1+\sqrt{17}}{4} \approx 1.28$ and provide a matching lower bound.

We also show competitiveness results for a special class of algorithms (called zealous algorithms) that do not allow waiting time for the server as long as there are requests unserved.

1. INTRODUCTION

The traveling salesman problem is a well studied problem in combinatorial optimization. In the classical setting, one assumes that the complete input of an instance is available for an algorithm to compute a solution. In many cases this *offline optimization* model does not reflect the real-world situation appropriately. In many applications not all requests for points to be visited are known in advance. Decisions have to be made *online* without knowledge about future requests.

Online algorithms are tailored to cope with such situations. Whereas offline algorithms work on the complete input sequence, online algorithms only see the requests released so far and thus, in planning a route, have to account for future requests that may or may not arise at a later time. A common way to evaluate the quality of online algorithms is *competitive analysis* [BEY98, FW98]. The outcome of the online

¹Combinatorial Optimization Group, Technical University of Eindhoven, P. O. Box 513, 5600MB Eindhoven, The Netherlands. Email: {mblom, leen}@win.tue.nl. Supported by the TMR Network DONET of the European Community ERB TMRX-CT98-0202

²Konrad-Zuse-Zentrum für Informationstechnik Berlin, Department Optimization, Takustr. 7, D-14195 Berlin-Dahlem, Germany. Email: krumke@zib.de. Research supported by the German Science Foundation (DFG, grant Gr 883/5-3)

³Faculty of Technology Management, Technical University of Eindhoven, P. O. Box 513, 5600MB Eindhoven, The Netherlands. Email: w.e.d.paepe@tm.tue.nl

Key words and phrases. Vehicle Routing, Online-Algorithms, Competitive Analysis, Traveling Salesman Problem.

algorithm is compared to that of an adversary, that provides the input sequence and therefore can achieve the optimal offline solution.

In this paper we consider the following online variant of the traveling salesman problem (called OLTSP in the sequel) which was introduced in [AFL⁺99]. Cities (requests) arrive online over time while the salesman is traveling. The requests are to be handled by a salesman-server that starts and ends his work at a designated origin. The objective is to find a routing for the server which finishes as early as possible (in scheduling theory this goal is usually referred to as minimizing the *makespan*). In this model the server is allowed to wait at the cost of time that elapses. Decisions are revocable, as long as they have not been executed. Only history is irrevocable.

1.1. Previous Work. Ausiello et al. [AFL⁺99] present a 2-competitive algorithm for OLTSP which works in general metric spaces. The authors also show that for general metric spaces no deterministic algorithm can be c -competitive with $c < 2$. For the special case that the metric space is \mathbb{R} , the real line, their best algorithm is $7/4$ -competitive, whereas a lower bound on the competitive ratio of any algorithm of $(9 + \sqrt{17})/8 \approx 1.64$ is derived [AFL⁺99]. Recently, Lipmann [Lip99] designed an algorithm for the problem on the real line with competitive ratio that matches the just mentioned lower bound.

1.2. Our Contribution. In this paper the effect of restricting the class of algorithms allowed and restricting the power of the adversary in the competitive analysis is studied. We introduce and analyze a new class of online algorithms which we call *zealous algorithms*. Roughly speaking, a zealous algorithm never sits idle while there is work to do. A similar concept was used for scheduling problems in [LL74]. A precise definition of zealousness is presented in Section 3 where we also show that in general zealous algorithms are strictly weaker than algorithms that allow waiting time. In particular we prove that no zealous algorithm can achieve a competitive ratio lower than $7/4$ for the OLTSP on the real line. The $7/4$ -competitive algorithm in [AFL⁺99] is in fact a zealous algorithm and therefore best possible within this restricted class of algorithms.

We then concentrate on the special case of OLTSP when the underlying metric space is \mathbb{R}_0^+ , the non-negative part of the real line. In Section 4 we show that an extremely simple and natural zealous strategy is $3/2$ -competitive and that this result is best possible for (zealous and non-zealous) deterministic algorithms on \mathbb{R}_0^+ .

The main contribution is contained in Section 5. Here we deal with an objection frequently encountered against competitive analysis concerning the unrealistic power of the adversary against which performance is measured. Indeed, for the OLTSP on the real line the before mentioned $7/4$ -competitive algorithm reaches its competitive ratio against an adversary that moves away from the previously released requests without giving any information to the online algorithm. We introduce an adversary who is in a natural way restricted in the context of the online traveling salesman problem studied here. We call it a *fair adversary*. It should be seen as a more reasonable adversary model. A fair adversary always keeps its server within the convex hull of the requests released so far. We show that this adversary model indeed allows for lower competitive ratios. For instance, the above mentioned $3/2$ -competitive zealous strategy against the

conventional adversary is $4/3$ -competitive against the fair adversary. This result is best possible for zealous algorithms against a fair adversary.

	Zealous Algorithms	General Algorithms
General Adversary	LB = UB = $3/2$	LB = UB = $3/2$
Fair Adversary	LB = UB = $4/3$	LB = UB = $(1 + \sqrt{17})/4$

TABLE 1. Overview of the lower bound (LB) and upper bound (UB) results for the competitive ratio of deterministic algorithms for OLTSP on \mathbb{R}_0^+ in this paper.

We also present a non-zealous algorithm with competitive ratio $(1 + \sqrt{17})/4 \approx 1.28 < 4/3$ competing against the fair adversary. Our result is the first one that shows that waiting is actually advantageous in the OLTSP. The before mentioned algorithm in [Lip99] also uses waiting, but became known after the one presented in this paper and has not been published officially yet. Such results are known already for online scheduling problems (see e.g. [HV96, CVW97, PSW95]) and, again very recently, also for an online dial-a-ride problem [AKR00]. Our competitiveness result is complemented by a matching lower bound on the competitive ratio of algorithms against the fair adversary. Table 1 summarizes our results for OLTSP on \mathbb{R}_0^+ .

We first continue the paper by giving precise descriptions of the problems studied, and the measure for analyzing performance of the algorithms.

2. PRELIMINARIES

An instance of the *Online Traveling Salesman Problem* (OLTSP) consists of a metric space $M = (X, d)$ with a distinguished origin $o \in M$ and a sequence $\sigma = \sigma_1, \dots, \sigma_m$ 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 mainly concerned with the special case that M is \mathbb{R}_0^+ , the non-negative part of the real line endowed with the Euclidean metric, i.e., $X = \mathbb{R}_0^+ = \{x \in \mathbb{R} : x \geq 0\}$, and $d(x, y) = |x - y|$; the origin o equals the point 0.

Each *request* is a pair $\sigma_i = (t_i, x_i)$, where $t_i \in \mathbb{R}$ is the time at which request σ_i is released (becomes known), and $x_i \in X$ is the point in the metric space requested to be visited. We assume that the sequence $\sigma = \sigma_1, \dots, \sigma_m$ of requests is given in order of non-decreasing release times. For a real number t we denote by $\sigma_{\leq t}$ the subsequence of requests in σ released up to time t . Similarly, $\sigma_{< t}$ is the subsequence of σ consisting of those requests with release time strictly smaller than t .

It is assumed that the online algorithm does neither have information about the time when the last request is released nor about the total number of requests.

An online algorithm for OLTSP must determine the behaviour of the server at a certain moment t of time as a function of all the requests in $\sigma_{\leq t}$ (and of the current time t). In contrast, the offline algorithm has information about all requests in the whole sequence σ already at time 0. A feasible online/offline solution is a route for the server which serves all requested points, where each request is served not earlier than the time it is released, and which starts and ends in the origin o .

The objective in the OLTSP is to minimize the total completion time (also called the *makespan* in scheduling) of the server, that is, the time when the server has served all requests and returned to the origin.

Let $\text{ALG}(\sigma)$ denote the completion time of the server moved by algorithm ALG on the sequence σ of requests. We use OPT to denote the optimal offline algorithm. An online algorithm ALG for OLTSP is *c-competitive*, if there exists a constant c such that for every request sequence σ the inequality $\text{ALG}(\sigma) \leq c \cdot \text{OPT}(\sigma)$ holds.

3. ZEALOUS ALGORITHMS

In this section we introduce a particular class of algorithms for OLTSP which we call *zealous algorithms*. Intuitively, a zealous algorithm should never sit and wait when it could serve yet unserved requests. A zealous server should also move towards work that has to be done directly without any detours. To translate this intuition into a rigorous definition some care has to be taken.

Definition 3.1 (Zealous Algorithm). An algorithm ALG for OLTSP is called *zealous*, if it satisfies the following conditions:

1. If there are still unserved requests, then the direction of the server operated by ALG changes only if a new request becomes known, or the server is either in the origin or at a request that has just been served.
2. At any time when there are yet unserved requests, the server operated by ALG either moves towards an unserved request or the origin at maximum (i.e. unit) speed. (The latter case is only allowed if the server operated by ALG is not yet in the origin).

We emphasize that a zealous algorithm is allowed to move its server towards an unserved request and change his direction towards another unserved request or to the origin at the moment a new request becomes known.

Theorem 3.2 ([AFL⁺99]). *Let ALG be a zealous online algorithm for OLTSP on the real line \mathbb{R} . Then the competitive ratio of ALG is at least $(9 + \sqrt{17})/8 \approx 1.64$.*

Lemma 3.3. *Let ALG be a zealous online algorithm for OLTSP on the real line \mathbb{R} . Then the competitive ratio of ALG is at least $7/4$.*

Proof. Suppose that ALG is a zealous algorithm for OLTSP on the real line. Consider the following adversarial input sequence. At time 0 and $1/2$ two requests $\sigma_1 = (0, 1/2)$ and $\sigma_2 = (1/2, 0)$, respectively, are released. There will be no further requests before time 1. Thus, by the zealousness of the algorithm the server will be at the origin at time 1.

At time 1 two new requests at points 1 and -1 , respectively, are released. Since the algorithm is zealous, starting at time 1 it must move its server to one of these requests at maximum, i.e., unit, speed. Without loss of generality assume that this is the request at 1. ALG 's server will reach this point at time 2. Starting at time 2, ALG will have to move its server either directly towards the unserved request at -1 or towards the origin, which essentially gives the same movement and implies that the server is at the origin at time 3. At that time, the adversary issues another request at 1. Thus, ALG 's server will still need at least 4 time units to serve -1 and 1 and return at the origin. Therefore, it will not be able to complete its work before time 7.

The offline adversary handles the sequence by first serving the request at -1 , then the two requests at 1 and finally returns to the origin at time 4, yielding the desired result. \square

This lower bound shows that the $7/4$ -competitive algorithm presented in [AFL⁺99], which is in fact a zealous algorithm, is best possible within the class of zealous algorithms for the OLTSP on the real line.

4. THE OLTSP ON THE NON-NEGATIVE PART OF THE REAL LINE

We first consider OLTSP on \mathbb{R}_0^+ when the offline adversary is the conventional (omnipotent) opponent.

Theorem 4.1. *Let ALG be any deterministic algorithm for OLTSP on \mathbb{R}_0^+ . Then the competitive ratio of ALG is at least $3/2$.*

Proof. At time 0 the request $\sigma_1 = (0, 1)$ is released. Let $T \geq 1$ be the time that the server operated by ALG has served the request σ_1 and returned to the origin 0. If $T \geq 3$, then no further request is released and ALG is no better than $3/2$ -competitive since $\text{OPT}(\sigma_1) = 2$. Thus, assume that $T < 3$.

In this case the adversary releases a new request $\sigma_2 = (T, T)$. Clearly, $\text{OPT}(\sigma_1, \sigma_2) = 2T$. On the other hand $\text{ALG}(\sigma_1, \sigma_2) \geq 3T$, yielding a competitive ratio of $3/2$. \square

The following extremely simple strategy achieves a competitive ratio that matches this lower bound (as we will show below):

Strategy MRIN(“Move-Right-If-Necessary”): If a new request is released and the request is to the right of the current position of the server operated by MRIN, then the MRIN-server starts to move right at full speed. The server continues to move right as long as there are yet unserved requests to the right of the server. If there are no more unserved requests to the right, then the server moves towards the origin 0 at full speed.

It is easy to verify that Algorithm MRIN is in fact a zealous algorithm. The following theorem shows that the strategy has a best possible competitive ratio for OLTSP on \mathbb{R}_0^+ .

Theorem 4.2. *Strategy MRIN is a zealous $3/2$ -competitive algorithm for the OLTSP on the non-negative part \mathbb{R}_0^+ of the real line.*

Proof. We show the theorem by induction on the number of requests in the sequence σ . It clearly holds if σ contains at most one request. The induction hypothesis states that the claim of the theorem holds for any sequence of $m - 1$ requests.

Suppose that request $\sigma_m = (t, x)$ is the last request of $\sigma = \sigma_1, \dots, \sigma_{m-1}, \sigma_m$. If $t = 0$, then MRIN is obviously $3/2$ -competitive, so we will assume that $t > 0$. Let f be the position of the request unserved by the MRIN-server at time t (excluding σ_m), which is furthest away from the origin (if all requests in $\sigma_1, \dots, \sigma_{m-1}$ have already been served by MRIN at time t then we set $f = 0$).

In case $x \leq f$, MRIN’s cost for serving σ is equal to the cost for serving the sequence consisting of the first $m - 1$ requests of σ . Since new requests can never decrease the optimal offline cost, the induction hypothesis implies the theorem.

Now assume that $f < x$. Thus, at time t the request in x is the furthest unserved request. If the position of MRIN at time t is to the right of x , then its cost does not increase by the release of σ_m . The claim then follows from the induction hypothesis as above. On the other hand, if at time t MRIN is to the left of x , then MRIN will complete

its work no later than time $t + 2x$. The optimal offline cost $\text{OPT}(\sigma)$ is bounded from below by $\max\{t + x, 2x\}$. Therefore,

$$\frac{\text{MRIN}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{t + x}{\text{OPT}(\sigma)} + \frac{x}{\text{OPT}(\sigma)} \leq \frac{t + x}{t + x} + \frac{x}{2x} = \frac{3}{2}.$$

□

The result established above can be used to obtain competitiveness results for the generalization of the OLTSP on the real line when there are $k \geq 2$ servers, and the goal is to minimize the time when the last of its servers returns to the origin 0 after all requests have been served.

Lemma 4.3. *There is an optimal offline strategy for OLTSP on the real line with $k \geq 2$ servers such that no server ever crosses the origin.*

Proof. Let σ be any request sequence and let (t_0^-, x_0^-) and (t_0^+, x_0^+) be the leftmost and rightmost request in σ . For any time t we denote by $\sigma_{>t}$ the subsequence of requests released strictly after time t .

Now consider the following offline strategy for routing the servers. Only two of the servers are used. At time 0 the first server moves to the right until point x_0^+ , the second moves to the left until it reaches x_0^- . We now describe the strategy for the server at x_0^+ for moving left. The situation for the other server is symmetric.

The server waits at x_0^+ until time $T := \max\{x_0^+, t_0^+\}$. Then it moves left until it reaches the position of x_T^+ , the rightmost request in the subsequence $\sigma_{>T}$, with release time t_T^+ . It waits there until time $S := \max\{T + x_0^+ - x_T^+, t_T^+\}$. The time parameter T is updated to $T := S$ and the left movement is continued as above. If $\sigma_{>T}$ becomes empty the server moves back to the origin.

It is easy to see that the parameter T maintained by the right server always has the property that $T + x_T^+$ is a lower bound for the optimum offline completion time. Thus, the strategy described above yields in fact an optimal offline solution. □

Corollary 4.4. *There is a 3/2-competitive algorithm for the OLTSP with $k \geq 2$ servers on the real line.*

Proof. The online algorithm uses two servers. One server handles all request on \mathbb{R}_0^+ , the other one serves the requests on \mathbb{R}_0^- . Each server uses the MRIN-strategy. It follows from Theorem 4.2 and Lemma 4.3 that this strategy is 3/2-competitive. □

5. FAIR ADVERSARIES

The adversaries used in the bounds of the previous section are abusing their power in the sense that they can move to points where they know a request will pop up without revealing the request to the online server before reaching the point. As an alternative we propose the following more reasonable adversary that we baptized *fair adversary*. We show that one can obtain better competitive ratios for the OLTSP on \mathbb{R}_0^+ under this model. We will also see that under this adversary model there exists a distinction in competitiveness between zealous and non-zealous algorithms.

Recall that $\sigma_{<t}$ is the subsequence of σ consisting of those requests with release time strictly smaller than t .

Definition 5.1 (Fair Adversary). An offline adversary for the OLTSP in the Euclidean space $(\mathbb{R}^n, \|\cdot\|)$ is *fair*, if at any moment t , the position of the server operated by the adversary is within the convex hull of the origin 0 and the requested points from $\sigma_{<t}$.

In the special case of \mathbb{R}_0^+ a fair adversary must always keep its server in the interval $[0, F]$, where F is the position of the request with the largest distance to the origin 0 among all requests released so far.

The following lower bound result shows that the OLTSP on the real line against fair adversaries is still a non-trivial problem.

Theorem 5.2. *Let ALG be any deterministic algorithm for OLTSP on \mathbb{R} . Then the competitive ratio of ALG against a fair adversary is at least $(5 + \sqrt{57})/8 \approx 1.57$.*

Proof. Suppose there exists a ρ -competitive online algorithm ALG, with $\rho = (5 + \sqrt{57})/8$. The adversarial sequence starts with two requests at time 0, $\sigma_1 = (0, 1)$ and $\sigma_2 = (0, -1)$. Without loss of generality, we assume that the first request which is served by ALG is σ_1 . At time 2 the online server can not have served both requests. We distinguish two main cases divided in some subcases.

Case 1: None of the requests has been served at time 2.

- If at time 3 request σ_1 is still unserved, let t' be the first time the server crosses the origin after serving the request. Clearly, $t' \geq 4$. At time t' the online server still has to visit the request in -1 .

If $t' > 4\rho - 2$ the server can not be ρ -competitive because the fair adversary can serve the sequence and be back in the origin at time 4.

Thus, suppose that $4 \leq t' \leq 4\rho - 2$. At time t' a new request $\sigma_3 = (t', 1)$ is released. The online server can not finish the complete sequence before $t' + 4$, whereas the adversary needs time $t' + 1$. Therefore, $\rho \geq \frac{t'+4}{t'+1}$. For $4 \leq t' \leq 4\rho - 2$ we have that the expression $\frac{t'+4}{t'+1}$ is decreasing in t' . Thus

$$\rho \geq \frac{(4\rho - 2) + 4}{(4\rho - 2) + 1} = \frac{4\rho + 2}{4\rho - 1}$$

implying $\rho \geq (5 + \sqrt{57})/8$.

- If at time 3 the request σ_1 has already been served, the online server can not be to the left of the origin at time 3 (given the fact that at time 2 no request had been served). The adversary now gives a new request $\sigma_3 = (3, 1)$. There are two possibilities: either σ_2 , the request in -1 , is served before σ_3 or the other way round.

If the server decides to serve σ_2 before σ_3 then it can not complete before time 7. Since the adversary completes the sequence in time 4, the competitive ratio is at least $7/4$.

If the online server serves σ_3 first, then again, let t' be the time that the server crosses the origin after serving σ_3 . As before, we must have $4 \leq t' \leq 4\rho - 2$. At time t' the fourth request $\sigma_4 = (t', 1)$ is released. The same arguments as above apply to show that the algorithm is at least ρ -competitive with $\rho = (5 + \sqrt{57})/8$.

Case 2: σ_1 has been served at time 2 by the online server.

At time 2 the third request $\sigma_3 = (2, 1)$ is released. In fact, we are now back in the situation in which at time 2 none of the two requests are served. In case the movements

of the online server are such that no further request is released by the adversary, the latter will complete at time 4. In the other cases the last released requests are released after time 4 and the adversary can still reach them in time. \square

For comparison, the best possible algorithm for the OLTSP in \mathbb{R} against an adversary that is not restricted to be fair is $\frac{9+\sqrt{17}}{8}$ -competitive [Lip99]. So far we have not been able yet to design algorithm that has competitive ratio $(5 + \sqrt{57})/8$ against a fair adversary. In that sense the picture is complete for the problem on the non-negative part of the real line (see Theorems 5.4 and 5.5 for zealous algorithms and Theorems 5.3 and 5.6 for non-zealous algorithms below).

Theorem 5.3. *Let ALG be any deterministic algorithm for OLTSP on \mathbb{R}_0^+ . Then the competitive ratio of ALG against a fair adversary is at least $(1 + \sqrt{17})/4 \approx 1.28$.*

Proof. Suppose that ALG is ρ -competitive for some $\rho \geq 1$. At time 0 the adversary releases request $\sigma_1 = (0, 1)$. Let T denote the time that the server operated by ALG has served this request and is back at the origin. For ALG to be ρ -competitive, we must have that $T \leq \rho \cdot \text{OPT}(\sigma_1) = 2\rho$. At time T the adversary releases a second request $\sigma_2 = (T, 1)$. The completion time of ALG becomes then at least $T + 2$.

On the other hand, starting at time 0 the fair adversary moves its server to 1, lets it wait there until time T and then goes back to the origin 0 yielding a completion time of $T + 1$. Therefore,

$$\frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)} \geq \frac{T + 2}{T + 1} \geq \frac{2\rho + 2}{2\rho + 1} = 1 + \frac{1}{2\rho + 1},$$

given the fact that $T \leq 2\rho$. Since by assumption ALG is ρ -competitive, we have that $1 + 1/(2\rho + 1) \leq \rho$, implying that $\rho \geq (1 + \sqrt{17})/4$. \square

For zealous algorithms we can show a higher lower bound against a fair adversary.

Theorem 5.4. *Let ALG be any deterministic zealous algorithm for OLTSP on \mathbb{R}_0^+ . Then the competitive ratio of ALG against a fair adversary is at least $4/3$.*

Proof. Consider the adversarial sequence consisting of the three requests $\sigma_1 = (0, 1)$, $\sigma_2 = (1, 0)$, and $\sigma_3 = (2, 1)$.

By its zealousness the online algorithm will start to travel to 1 at time 0, back to 0 at time 1, arriving there at time 2. Then its server has to visit 1 again, so that he will finish no earlier than time 4. Obviously, the optimal fair offline solution is to leave 1 not before time 2, and finishing at time 3. \square

We show now that the algorithm MRIN presented before has a better competitive ratio against the fair adversary than the $3/2$ against a conventional adversary. In fact we show the ratio matches the lower bound for zealous algorithms proved in Theorem 5.4.

Theorem 5.5. *Strategy MRIN is a $4/3$ -competitive algorithm for the OLTSP on \mathbb{R}_0^+ against a fair adversary.*

Proof. Again we use induction on the number of requests in the sequence σ to establish the claim of the theorem. The claim clearly holds if σ contains at most one request. The induction hypothesis states that the claim of the theorem holds for any sequence of $m - 1$ requests.

Let $\sigma = \sigma_1, \dots, \sigma_m$ be any sequence of requests. We consider the time t_m when the last set of requests $\sigma_{=t_m}$ is released. If $t = 0$, then the claim obviously holds, so we will assume for the remainder of the proof that $t > 0$. Let $\sigma_m = (t, x)$ be that request of $\sigma_{=t}$ which is furthest away from the origin.

In the sequel we denote by $s(t)$ and $s^*(t)$ the positions of the MRIN- and the fair adversary server at time t , respectively.

Let $r_f = (t_f, f)$ be the furthest unserved request by MRIN of the subsequence $\sigma_{<t}$ at time t , that is, the unserved request from $\sigma_{<t}$ most remote from the origin 0. Finally, let $r_F = (t_F, F)$ be the furthest request in $\sigma_{<t}$. Notice that by definition $f \leq F$.

We now distinguish different cases depending on the position x of the request σ_m relative to f and F .

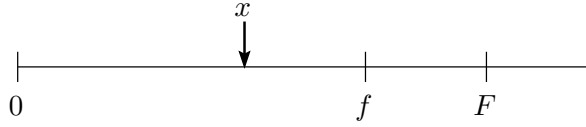


FIGURE 1. Case 1 of the proofs of Theorem 5.5 and Theorem 5.6.

Case 1: $x \leq f$ (see Figure 1)

Since the MRIN-server still has to travel to f , all the requests in $\sigma_{=t}$ will be served on the way back to the origin and the total completion time of the MRIN-server will not increase by releasing the requests $\sigma_{=t}$. Since new requests can never decrease the optimal offline solution value, the claim follows from the induction hypothesis.

Case 2: $f \leq x < F$

If $s(t) \geq x$, again MRIN's completion time does not increase compared to the situ-

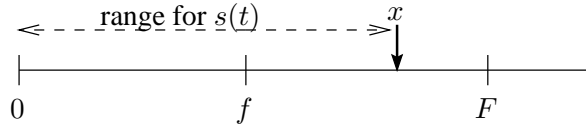


FIGURE 2. Case 2 of the proof of Theorems 5.5 and 5.6.

ation before the requests in $\sigma_{=t}$ were released, so we can assume that $s(t) \leq x$ (see Figure 2). The MRIN-server will now travel to point x which needs time $d(s(t), x)$, and then return to the origin. Thus, $\text{MRIN}(\sigma) = t + d(s(t), x) + x$. On the other hand $\text{OPT}(\sigma) \geq t + x$. It follows that

$$\frac{\text{MRIN}(\sigma)}{\text{OPT}(\sigma)} \leq 1 + \frac{d(s(t), x)}{\text{OPT}(\sigma)} \quad (1)$$

We now show that $\text{OPT}(\sigma)$ is at least 3 times $d(s(t), x)$, this will establish the claimed ratio of $4/3$. Notice that $f < F$ (since $f \leq x < F$) and the fact that f is the furthest unserved request at time t implies that the MRIN-server must have already visited F

at time t (otherwise the furthest unserved request would be at F and not at $f < F$). Therefore, $t \geq F + d(F, s(t))$, and

$$\text{OPT}(\sigma) \geq t + x \geq F + d(F, s(t)) + x. \quad (2)$$

Clearly, each of the terms on the right hand side of inequality (2) is at least $d(s(t), x)$.

Case 3: $f \leq F \leq x$ (see Figure 3)

First remember that the MRIN-server always moves to the right if there are yet unserved requests to the right of his present position available.

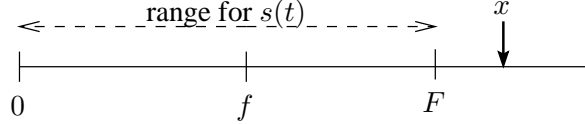


FIGURE 3. Case 3 of the proof of Theorem 5.5 and Theorem 5.6.

Since the last request (t, x) is at least as far away from the origin as F , the optimal offline server will only move left after it has served the furthest request in σ , in this case at x . In fact, the optimal fair offline strategy is as follows: as long as there are unserved requests to the right of the server, move right, otherwise wait at the current position. As soon as the last request (t, x) has been released and the offline server has reached x , it moves to the origin and completes its work (see also the description of the optimal offline strategy has been described in the proof of Lemma 4.3).

This implies that at any time in the interval $[0, t]$ the fair adversary's server is to the right of the MRIN-server (or at the same position).

Because the offline server does not return to the origin as long as there will be new requests released to the right of its current position, the distance between the MRIN-server and the offline server increases only if the offline server is waiting at some point. Let $W^*(t)$ be the total waiting time of the offline server at the moment t when the last request x is released. Then we know that

$$d(s(t), s^*(t)) \leq W^*(t). \quad (3)$$

Moreover, the following relation between the current time and the waiting time holds:

$$t = d(o, s^*(t)) + W^*(t). \quad (4)$$

Since the adversary is fair, its position $s^*(t)$ at time t can not be to the right of F . Thus, $d(s^*(t), x) = d(s^*(t), F) + d(F, x)$ which gives us

$$\text{OPT}(\sigma) \geq t + d(s^*(t), F) + d(F, x) + x \quad (5)$$

$$\stackrel{(4)}{=} d(o, s^*(t)) + W^*(t) + d(s^*(t), F) + d(F, x) + x$$

$$= W^*(t) + F + d(F, x) + x$$

$$= W^*(t) + 2x$$

$$\geq W^*(t) + 2d(s(t), s^*(t))$$

$$\stackrel{(3)}{\geq} 3d(s(t), s^*(t)) \quad (6)$$

At time t MRIN's server has to move from its current position $s(t)$ to x and from there move to the origin:

$$\begin{aligned} \text{MRIN}(\sigma) &= t + d(s(t), x) + x \\ &= t + d(s(t), s^*(t)) + d(s^*(t), F) + d(F, x) + x. \end{aligned}$$

Hence,

$$\begin{aligned} \frac{\text{MRIN}(\sigma)}{\text{OPT}(\sigma)} &= \frac{t + d(s^*(t), F) + d(F, x) + x}{\text{OPT}(\sigma)} + \frac{d(s(t), s^*(t))}{\text{OPT}(\sigma)} \\ &\stackrel{(5)}{\leq} 1 + \frac{d(s(t), s^*(t))}{\text{OPT}(\sigma)} \\ &\stackrel{(6)}{\leq} \frac{4}{3}. \end{aligned}$$

This proves the claim. \square

Given the lower bound for general non-zealous algorithms in Theorem 5.3 we conclude that online algorithms which may obtain better competitive ratio's against a fair adversary will have to be non-zealous, i.e., incorporate waiting times.

The problem with Algorithm MRIN is that shortly after it starts to return towards the origin from the furthest previously unserved request, a new request to the right of its server arrives (becomes known). In this case the MRIN-server has to return to a position it just left. Algorithm WS presented below attempts successfully to avoid this pitfall.

Strategy WS("Wait Smartly"): The WS-server moves right if there are yet unserved requests to the right of his present position. Otherwise, it takes the following actions. Suppose it arrives at his present position $s(t)$ at time t .

1. Compute the the optimal offline solution value $\text{OPT}(\sigma_{\leq t})$ for all requests released up to time t .
2. Determine a waiting time $W := \alpha \text{OPT}(\sigma_{\leq t}) - s(t) - t$, with $\alpha = (1 + \sqrt{17})/4$.
3. Wait at point $s(t)$ until time $t+W$ and then start to move back to the origin 0.

We note that when the server is moving back to the origin and no new requests are released until time $t + W + s(t)$, then the WS-server reaches the origin 0 at time $t + W + s(t) = \alpha \cdot \text{OPT}(\sigma_{\leq t})$ having served all requests released so far. If a new request is released at time $t' \leq W + t + s(t)$ and the request is to the right of $s(t')$, then the WS-server interrupts its waiting and starts to move to the right immediately until it reaches the furthest unserved request.

Theorem 5.6. *Algorithm WS is α -competitive for the OLTSP on the non-negative part \mathbb{R}_0^+ of the real line against a fair adversary, where $\alpha = (1 + \sqrt{17})/4 \approx 1.28$.*

Proof. By the definition of the waiting time it is sufficient to prove that at any point where a waiting time is computed this waiting time is non-negative. In that case the server will always return at 0 before time $\alpha \text{OPT}(\sigma)$. This is clearly true if the sequence σ contains at most one request. We make the induction hypothesis that WS is α -competitive for any sequence of at most $m - 1$ requests.

Let $\sigma = \sigma_1, \dots, \sigma_m$ be any sequence of requests. As in the proof of Theorem 5.5 we consider the time $t = t_m$ when the last set of requests $\sigma_{=t}$ is released and let

$\sigma_m = (t, x)$ be that requests of $\sigma_{=t}$ which is furthest away from the origin. If $t = 0$, then the claim obviously holds, so we will assume for the remainder of the proof that $t > 0$.

We denote by $s(t)$ and $s^*(t)$ the positions of the WS- and the fair adversary's server at time t , respectively. As before we let $r_f = (t_f, f)$ be the furthest (i.e. most remote from the origin) yet unserved request by WS at time t of $\sigma_{<t}$. Finally, let $r_F = (t_F, F)$ be the furthest released request in $\sigma_{<t}$.

We now distinguish three different cases depending on the position of x relative to f and F . Recall that $f \leq F$.

Case 1: $x \leq f$ (see Figure 1)

Since the WS-server has to travel to f anyway and by the induction hypothesis there was a non-negative waiting time in f or $s(t)$ (depending on whether $s(t) > f$ or $s(t) \leq f$) before requests $\sigma_{=t}$ were released, the waiting time in f or $s(t)$ can not decrease since the optimal offline completion time can not decrease by an additional request.

Case 2: $f \leq x < F$

If $s(t) \geq x$, then again by the induction hypothesis and the fact that the route length of WS's server does not increase, the possible waiting time at $s(t)$ is non-negative.

Thus we can assume that $s(t) < x$ (see Figure 2). The WS-server will now travel to point x , arrive there at time $t + d(s(t), x)$, and possibly wait there some time W before returning to the origin, with

$$W = \alpha \text{OPT}(\sigma) - (t + d(s(t), x) + x).$$

Inserting the obvious lower bound $\text{OPT}(\sigma) \geq t + x$ yields

$$W \geq (\alpha - 1)\text{OPT}(\sigma) - d(s(t), x). \quad (7)$$

To bound $\text{OPT}(\sigma)$ in terms of $d(s(t), x)$ consider the time t' when the WS-server had served the request at F and started to move left. It must be that $t' < t$ since otherwise $s(t)$ could not be smaller than x as assumed. Thus, the subsequence $\sigma_{\leq t'}$ of σ does not contain (t, x) . By the induction hypothesis, WS is α -competitive for the sequence $\sigma_{\leq t'}$. At time t' when it left F it would have arrived at α times the optimal offline cost $\text{OPT}(\sigma_{\leq t'})$ on that subsequence:

$$t' + F = \alpha \cdot \text{OPT}(\sigma_{\leq t'}). \quad (8)$$

Notice that $t \geq t' + d(F, s(t))$. Since $\text{OPT}(\sigma_{\leq t'}) \geq 2F$ we obtain from (8) that

$$t \geq \alpha 2F - F + d(F, s(t)) = (2\alpha - 1)F + d(s(t), F). \quad (9)$$

Since by assumption we have $s(t) < x < F$ we get that $d(s(t), x) \leq d(s(t), F)$ and $d(s(t), x) \leq F$, which inserted in (9) yields

$$t \geq (2\alpha - 1)d(s(t), x) + d(s(t), x) = 2\alpha d(s(t), x). \quad (10)$$

We combine this with the previously mentioned lower bound $\text{OPT}(\sigma) \geq t + x$ to obtain:

$$\text{OPT}(\sigma) \geq 2\alpha d(s(t), x) + x \geq (2\alpha + 1)d(s(t), x). \quad (11)$$

Using inequality (11) in (7) gives

$$\begin{aligned}
W &\geq (\alpha - 1)(2\alpha + 1)d(s(t), x) - d(s(t), x) \\
&= (2\alpha^2 - \alpha - 2)d(s(t), x) \\
&= \left(\frac{9 + \sqrt{17}}{4} - \frac{1 + \sqrt{17}}{4} - 2 \right) d(s(t), x) \\
&= 0.
\end{aligned}$$

This completes the proof for the second case.

Case 3: $f \leq F \leq x$ (see Figure 3)

Starting at time t the WS-server moves to the right until it reaches x , and after waiting there an amount W returns to 0, with

$$W = \alpha \text{OPT}(\sigma) - (t + d(s(t), x) + x). \quad (12)$$

Again we will show that $W \geq 0$, i.e., that also in this case the computed waiting time at x for WS is nonnegative. At time t the adversary's server still has to travel at least $d(s^*(t), x) + x$ units. This results in

$$\text{OPT}(\sigma) \geq t + d(s^*(t), x) + x.$$

Since the offline adversary is fair, its position $s^*(t)$ at time t can not be strictly to the right of F . This yields

$$\text{OPT}(\sigma) \geq t + d(F, x) + x. \quad (13)$$

Insertion into (12) yields

$$\begin{aligned}
W &\geq (\alpha - 1)\text{OPT}(\sigma) + \text{OPT}(\sigma) - (t + d(s(t), x) + x) \\
&\geq (\alpha - 1)\text{OPT}(\sigma) + d(F, x) - d(s(t), x) \\
&= (\alpha - 1)\text{OPT}(\sigma) - d(s(t), F)
\end{aligned} \quad (14)$$

since $s(t) \leq F$ by definition of the algorithm WS.

The rest of the arguments are similar to those used in the previous case. Again suppose that WS's server started to move to the left from F at some time $t' \leq t$ (where t' is chosen maximal). By the induction hypothesis the WS-server would have returned to the origin at time $\alpha \text{OPT}(\sigma_{<t'})$ (if the requests in $\sigma_{=t}$ had not been released). Hence

$$t' + F = \alpha \text{OPT}(\sigma_{<t'}). \quad (15)$$

Notice that $t \geq t' + d(s(t), F)$ and $\text{OPT}(\sigma_{<t'}) \geq 2F$ (by the fact that $\sigma_{<t'}$ must contain at least one request at F since otherwise WS would not have moved its server to F). Hence we obtain from (15) that

$$t \geq \alpha 2F - F + d(s(t), F) = (2\alpha - 1)F + d(s(t), F) \geq 2\alpha d(s(t), F).$$

We combine this with (13) and the fact that $x \geq F \geq d(s(t), F)$ to achieve

$$\text{OPT}(\sigma) \geq 2\alpha d(s(t), F) + d(F, x) + x \geq (2\alpha + 1)d(s(t), F).$$

Using this inequality in (14) gives

$$\begin{aligned}
 W &\geq (\alpha - 1)(2\alpha + 1)d(s(t), F) - d(s(t), F) \\
 &= (2\alpha^2 - \alpha - 2)d(s(t), F) \\
 &= \left(\frac{9 + \sqrt{17}}{4} - \frac{1 + \sqrt{17}}{4} - 2 \right) d(s(t), F) \\
 &= 0.
 \end{aligned}$$

This completes the proof. \square

6. CONCLUSIONS

We introduced an alternative more fair performance measure for online algorithms for the traveling salesman problem. The first results are encouraging. On the non-negative part of the real line the fair model allows a strictly lower competitive ratio than the conventional model with an omnipotent adversary.

Next to that we considered a restricted class of algorithms for the online traveling salesman problems, suggestively called zealous algorithms. We showed that in general zealous algorithms have strictly higher competitive ratios than algorithms that sometimes leaves the server idle, to wait for possible additional information. In online routing companies, like courier services or transportation companies waiting instead of immediately starting as soon as requests are presented is common practice. Our results support this strategy.

For the problem on the real line our results together with the recent result of Lipmann [Lip99] show that non-zealous algorithms can do strictly better than zealous algorithms against a conventional adversary, and our results suggest that this is the same against a fair adversary. However, it remains open to find a non-zealous algorithm that beats the best possible zealous ones against a fair adversary in case of the real line as underlying metric space.

We notice here that for general metric spaces the lower bound of 2 on the competitive ratio of algorithms in [AFL⁺99] is established with a fair adversary as opponent. Moreover, a zealous algorithm is presented which has a competitive ratio that meets the lower bound.

We hope to have encouraged research into ways to restrict the power of adversaries in online competitive analysis.

Acknowledgement: Thanks to Maarten Lipmann for providing the lower bound in Theorem 5.2.

REFERENCES

- [AFL⁺99] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo, *Algorithms for the on-line traveling salesman*, *Algorithmica* (1999), To appear.
- [AKR00] N. Ascheuer, S. O. Krumke, and J. Rambau, *Online dial-a-ride problems: Minimizing the completion time*, Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 1770, Springer, 2000, pp. 639–650.
- [BEY98] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*, Cambridge University Press, 1998.

- [CVW97] B. Chen, A. P. A. Vestjens, and G. J. Woeginger, *On-line scheduling of two-machine open shops where jobs arrive over time*, Journal of Combinatorial Optimization **1** (1997), 355–365.
- [FW98] A. Fiat and G. J. Woeginger (eds.), *Online algorithms: The state of the art*, Lecture Notes in Computer Science, vol. 1442, Springer, 1998.
- [HKR00] D. Hauptmeier, S. O. Krumke, and J. Rambau, *The online dial-a-ride problem under reasonable load*, Proceedings of the 4th Italian Conference on Algorithms and Complexity, Lecture Notes in Computer Science, vol. 1767, Springer, 2000, pp. 125–136.
- [HV96] J. A. Hoogeveen and A. P. A. Vestjens, *Optimal on-line algorithms for single-machine scheduling*, Proceedings of the 5th Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, 1996, pp. 404–414.
- [Lip99] M. Lipmann, *The online traveling salesman problem on the line*, Master’s thesis, Department of Operations Research, University of Amsterdam, The Netherlands, 1999.
- [LL74] J. W. S. Liu and C. L. Liu, *Performance analysis of heterogenous multi-processor computign systems*, Computer architectures and networks (E. Gelenbe and R. Mahl, eds.), North Holland, 1974, pp. 331–343.
- [NS99] J. Noga and S. Seiden, *Scheduling two machines with release times*, Proceedings of the 7th Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, vol. 1610, Springer, 1999, pp. 381–399.
- [PSW95] C. Philips, C. Stein, and J. Wein, *Minimizing average completion time in the presence of release dates*, Proceedings of the 4th Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, vol. 955, 1995, pp. 86–97.