Algorithms for the On-Line Quota Traveling Salesman Problem *

G. Ausiello[1], M. Demange[2], L. Laura[1], and V. Paschos[3]

[1] Dip. di Informatica e Sistemistica Università di Roma "La Sapienza" Via Salaria 113 00198 Roma Italy. {ausiello,laura}@dis.uniroma1.it
[2] ESSEC demange@essec.fr
[3] Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France paschos@lamsade.dauphine.fr

Abstract. The Quota Traveling Salesman Problem is a generalization of the well known Traveling Salesman Problem. The goal of the traveling salesman is, in this case, to reach a given *quota* of the sales, minimizing the amount of time. In this paper we address the on-line version of the problem, where requests are given over time. We present algorithms for various metric spaces, and analyze their performance in the usual framework of the competitive analysis. In particular we present a 2-competitive algorithm that matches the lower bound for general metric spaces. In the case of the half-line metric space, we show that it is helpful not to move at full speed, and this approach is also used to derive the best on-line polynomial time algorithm known so far for the more general On-Line TSP problem (in the homing version).

1 Introduction

Let us imagine that a traveling salesman is not forced to visit all cities in a single tour but in each city he can sell a certain amount of merchandise and his committment is to reach a given *quota* of sales, by visiting a sufficient number of cities; then he is allowed to return back home. The problem to minimize the amount of time in which the traveling salesman fulfills his commitment is known as the Quota Traveling Salesman Problem (QTSP for short, see [4, 9] for a definition of the problem) and it is also called Quorum-Cast problem in [11]. Such problem can be seen as a special case of the Prize-Collecting Traveling Salesman Problem (PCTSP[4], [6]) in which again the salesman has to fulfill a quota but now nonnegative penalties are associated to the cities and the cost of the salesman tour is the sum of the distance traveled and the penalties for the non visited cities. QTSP corresponds to the case of PCTSP in which all penalties

* Work of Giorgio Ausiello and Luigi Laura is partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT); and by "Progetto ALINWEB: Algoritmica per Internet e per il Web", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale.

[4] Note that some authors use to name PCTSP the special case that we have called QTSP [5, 8].

are 0. A special case of QTSP is the case in which the amount of merchandise that can be sold in any city is 1; this case coincides with the so called k-TSP problem, i.e. the problem to find the minimum tour which visits k cities among the given ones. Moreover this problem is related with the k-MST problem, that is the problem to find the minimum tree which spans k nodes in a graph. Clearly, if all weights are equal to 1 and the quota to be achieved is equal to the number of cities, the problem corresponds to the classic TSP.

The QTSP problem and the other problems mentioned above have been thoroughly studied in the past from the point of view of approximation algorithms. In particular, for the k-TSP problem and for the k-MST problem a polynomial time algorithm with an approximation ratio 3 has been shown in [15] while, for k-MST problem, in the non rooted case, an algorithm with ratio 2.5 has been shown in [1]. For the general PCTSP problem a polynomial time algorithm with polylogarithmic performance guarantee has been given in [4, 5].

In this paper we wish to address the on-line version of the QTSP problem, named OL-QTSP. On line versions of other routing problems such as the traveling salesman problem [3], the traveling repairman problem [13, 16, 18], variants of the dial-a-ride problem [2, 13], have been studied in the literature in the recent years. The most updated results regarding these problems can be found in [17]. In the on-line version of QTSP we imagine that requests are given over time in a metric space and a server (the traveling salesman) has to decide which requests to serve and in what order to serve them, without yet knowing the whole sequence of requests, with the aim of fulfilling the quota by traveling the minimum possible amount of time. As it is common in the evaluation of on-line algorithms [14], the performance of the algorithm (cost of serving the requests needed to fulfill the quota), is matched against the performance of an optimum off-line server, that is a traveling salesman that knows all the requests ahead of time and decides which requests to serve and in what order, in order to fulfill the assigned quota. Clearly the off-line server cannot serve a request before its release time. The ratio between the former and the latter is called *competitive ratio* of the on-line algorithm (see [10]).

In the rest of the paper we first provide a formal definition of the problem and we introduce the corresponding notation. Subsequently, in Section 2 we provide the proof that no algorithm for the OL-QTSP problem can achieve a competitive ratio smaller than 2 in a general metric space, and we also give a simple 2-competitive (hence optimal) on-line algorithm. In Section 3 we discuss the case in which the metric space is the halfline. In such case we introduce an algorithm which achieves a competitive ratio 3/2 by not moving at full speed. We also show a matching lower bound. In Section 4 we apply the same 'slow is good' paradigm used in Section 3, to the design of an on-line algorithm for the OL-TSP and in such way we derive the best on-line polynomial algorithm known so far, whose competitive ratio is 2.78. In the Conclusion section we illustrate some possible developments and open problems.

1.1 Statement of the problem

Let us define the OL-QTSP problem in a formal way. Let M be a general metric space, let us denote with O the origin of M, and let $r_1 \dots r_n$ be an ordered sequence of requests in M, any request r_i being a triple (t_i, p_i, w_i) where t_i is the release time of the request, p_i is a point in the metric space where the request is released and w_i is the weight of the request. When we refer to the non-weighted problem a request is simply a couple (t_i, p_i). Note that the sequence is ordered in the sense that if $i < j$ then $t_i < t_j$. The OL-QTSP problem is the problem in which an on-line server (traveling salesman) is supposed to serve at least a quota Q out of the overall amount of requests $\sum_{j=1}^{n} w_j$ and come back to the origin in the minimum amount of time. In the non-weighted version of the problem the quota Q is simply the minimum number of requests to be served. Clearly no request can be served before its release time.

In the following we will denote by OL the On-Line Server, and by OPT the server moved by the optimum off line server (adversary). Besides, for any sequence of requests σ, by $Z^{OL}(\sigma)$ and $Z^*(\sigma)$ we denote, respectively, the completion time of OL and OPT over the sequence σ; when there are no ambiguities, we write Z^{OL} and Z^* instead of $Z^{OL}(\sigma)$ and $Z^*(\sigma)$.

Furthermore, with $p^{OL}(t)$ and $p^*(t)$ we denote respectively the position of OL and OPT at time t, and with $d(p_i, p_j)$ we represent the distance between the points p_i and p_j in the metric space M.

2 General metric spaces

2.1 Lower bound of the problem

We first show the following result, that will be extensively used in the rest of the paper and referenced as the *speed constraint*.

Lemma 1. *No ρ-competitive algorithm, in any metric space, at time t can have its server in a position distant from the origin more than $t \cdot (\rho - 1)$.*

Proof. We simply show that any algorithm that violates the constraint has a competitive ratio bigger than ρ. Assume that time t_1 is the first instant in which the above constraint is not respected, i.e. $d(p^{OL}(t_1), O) > t_1 \cdot (\rho - 1)$. Then, in the same time, the adversary releases a set of requests with overall quota Q in the origin, and these are served immediately by the adversary ($Z^* = t_1$). The completion time of the on-line server is not less than $Z^{OL} > t_1 \cdot (\rho - 1) + t_1 = \rho \cdot t_1$, and this means that the competitive ratio is strictly bigger than ρ: $\frac{Z^{OL}}{Z^*} > \frac{\rho \cdot t_1}{t_1} = \rho$. □

Just as the TSP is a special case of QTSP, it is easy to observe that the OL-TSP (in the Homing version, see [3]) is a special case of OL-QTSP . From this point of view from [3] we may trivially derive a lower bound of 2 on the competitive ratio for OL-QTSP . We now prove a stronger result that provides a lower bound for the competitiveness of any on-line algorithm for OL-QTSP in a general metric space, even when the quote Q is bounded by a constant.

Theorem 1. *There is no ρ-competitive algorithm for the OL-QTSP on the line with $\rho < 2$, even with a maximum of 3 requests released.*

Proof. Assume the metric space is the real line. The quota Q is equal to 2. Denote by A and B the abscissa points -1 and +1 on the real line, respectively. At moment $t = 0$, a request r_- is released on A and a request r_+ is released at B; then, no request is released until moment $t = 1$.

Observe first that $p^{OL}(1) \in [-1, 1]$, but, from Lemma 1, we know that at time 1 OL cannot be in -1 or 1 so $p^{OL}(1) \in]-1, 1[$. Then, for $x \in [0, 1]$, define $f(x) = \min\{d(A, p^{OL}(1 + x)), d(B, p^{OL}(1 + x))\}$ and $g(x) = f(x) - x$. Clearly, g is continuous if server moves in a continuous way. Also, $g(0) = f(0) > 0$ and $g(1) = f(1) - 1 < 0$, since $p^{OL}(2) \in]-2, 2[$. Hence, there exists $\overline{x} \in [0, 1]$ such that $g(\overline{x}) = 0$, i.e., $f(\overline{x}) = \overline{x}$. From now on let x_0 be the minimum such $\overline{x}$.

Assume first that $f(x_0) = d(B, p^{OL}(1 + x_0)) = x_0$ (this means that $p^{OL}(1 + x_0) \in \{1-x_0, 1+x_0\}$). Actually, due to Lemma 1, only one of the two possibilities holds and hence: $p^{OL}(1 + x_0) = 1 - x_0$. In this case, if the third (last) request r_3 is released in $-1 + x_0$ at moment $t = 1 + x_0$, then the optimum is equal to 2 (see Figure 5: request r_+ is served at $t = 1$, r_3 at $1 + x_0$, and return to the origin is at time 2). On the other hand, since $p^{OL}(1+x_0) = 1-x_0$, then then the server can serve either r_+ and r_3, or r_- and r_3 in time equal to $(1+x_0)+x_0+1+2(1-x_0) = 4$. In both of the above cases, the OL server cannot guarantee a competitive ratio better than 2.

Assume now $f(x_0) = d(A, p^{OL}(1 + x_0)) = x_0$. In this case, considering a request r_3 released in $1 - x_0$ at moment $t = 1 + x_0$, arguments exactly analogous to the previous ones imply a competitive upper bound of 2. The proof of the theorem is now complete. □

Note that the above lower bound is achieved with $Q = 2$, the total number of requests released is 3 and two requests are released at time 0: only one request was released *on-line*. Obviously, the result of Theorem 1 does not work for the case $Q = 1$, neither for the case in which $Q = 2$ and only two request are released. These cases are addressed in Subsection 2.3

2.2 A 2-competitive algorithm

The following algorithm is 2-competitive for the OL-QTSP problem in any metric space:

Algorithm 1 (Wait and Go (WaG)) *For any sequence σ already presented to the server, with at least a quota Q available over all the requests in σ, the algorithm computes $Z^*(\sigma)$; at time $t = Z^*(\sigma)$ the algorithm starts an optimal tour that serves the quota and ends back in the origin.*

Since WaG starts its tour at time Z^*, it concludes it at time $t \leq 2 \cdot Z^*$. So we can state the following:

Theorem 2. *Algorithm WaG is 2-competitive for the OL-QTSP problem in general metric space.*

Clearly the upper bound provided by the algorithm matches the lower bound previously given for general metric spaces and hence no better competitive ratio can be established in the general case.

Note also that, if we deal with the non-weighted problem, where Q is the number of requests, if Q is fixed and it is not part of the input the optimal solution of the QTSP problem can be computed in polynomial time and, therefore, WaG is a polynomial time algorithm. If Q is part of the input, Tsitsiklis [19] proved that, for the off-line problem, the optimal solution can be computed in polynomial time if the underlying metric space is the line (or the halfline), while, for general metric spaces, we have to resort to an approximation algorithm for the OL-QTSP . Hence if such algorithm provides a solution with approximation ratio r we obtain a $2 \cdot r$ competitive algorithm.

2.3 Particular cases

In Theorem 1 we proved that, provided that $Q = 2$ and a minimum af 3 requests are released, no algorithm can achieve a competitive ratio better than 2. In this section we deal with two particular cases that we obtain when we release the constraint on the quota and the number of requests.

We start with the case when $Q = 1$, and we prove that, for such case, OL-QTSP admits a competitive ratio 3/2.

The algorithm guaranteeing such a ratio is the following:

Algorithm 2 (Eventually Replan (ER)) *The server waits for the first moment t_0 where a request r is released at distance at most t_0; it goes to serve it at full speed; if on the road, another request is released such that the algorithm can serve it and return to the origin sooner than for r, it decides to serve this new request.*

Theorem 3. *ER is 3/2-competitive for the OL-QTSP with $Q = 1$.*

Proof. Suppose that an optimal algorithm should serve a request r_1 released at distance x_1 from the origin at moment t_1; then, $Z^* = \max\{t_1, x_1\} + x_1$.

Assume first $Z^* = 2x_1$ ($x_1 \geq t_1$). Then the server moved by EP starts before moment $t = x_1$ and returns to the origin (after serving a request) before $t = 3x_1$. So, the competitive ratio is $\rho \leq 3x_1/2x_1 = 3/2$.

Assume now $Z^* = t_1 + x_1$ ($x_1 \leq t_1$). If OL has not yet moved at moment $t = t_1$, then it moves in order to serve r_1 (since $2x_1 \leq 2t_1$). In this case the competitive ratio is $\rho = (t_1 + 2x_1)/(t_1 + x_1) \leq 3/2$ (recall that we deal with case $x_1 \leq t_1$). If the on-line algorithm has started to move before moment $t = x_1$, then $\rho \leq 3x_1/(x_1 + t_1) \leq 3/2$. Finally, assume that OL starts moving at the moment $t = t_0$, where $x_1 < t_0 < t_1$, in order to serve a request r_0 released at distance $x_0 \leq t_0$. At $t = t_1$, $d(O, OL(t_1)) \leq t_1 - t_0$. Independently on the request, the on-line algorithm has decided to serve at $t = t_1$ (this request may or may not be r_0), it decides to continue its way (guaranteeing $3t_0$), or to serve r_1. Hence, it can return in the origin before $t = \min\{3t_0, t_1 + (t_1 - t_0) + 2x_1\} = \min\{3t_0, 2Z^* - t_0\} \leq (3/2)Z^*$.

So, competitive ratio 3/2 is always guaranteed. □

Now we deal with the case when $Q = 2$ and only two requests are released. We show in the sequel that, also in this case, we can achieve competitive ratio $3/2$.

The algorithm achieving this ratio is the following:

Algorithm 3 (Serve and Wait (SaW)) *The server starts at the first moment t_0 such that it can serve a request r_0 before time $2 \cdot t_0$; once arrived in place r_0 is released, it waits until the second request is released; it serves it and it then returns in the origin.*

Theorem 4. *SaW is $3/2$-competitive for the OL-QTSP with $Q = 2$ and a total of two requests released.*

Proof. Denote by r_1 and r_2 the two requests released and assume that they are served in this order by an optimal algorithm; assume that this algorithm serves r_1 at instant τ_1, r_2 at instant τ_2 and that it returns in the origin at instant $\tau_3 = Z^*$. We distinguish the two following cases depending on the order followed by the server moved by SaW.

Assume first that OL serves r_1 before r_2. It starts before moment $t = \tau_1$ (this moment is the latest departure moment for the server) since it is possible to serve r_1 before moment $2 \cdot \tau_1$ and this request is already released at this moment. Since it aims at r_1, it serves it at moment $t = \tau_1 + d(0, r_1)$. If request r_2 is already released at this moment, then the server can serve both r_1 and r_2 and return to the origin before moment $\tau_1 + d(O, r_1) + (\tau_2 - \tau_1) + (\tau_3 - \tau_2) = \tau_3 + d(0, r_1) \leq 3/2\tau_3 = 3/2Z^*$. Otherwise, it ends its tour at most at moment $\tau_2 = d(r_1, r_2) + (\tau_3 - \tau_2) = \tau_3 + d(r_1, r_2) \leq 3/2\tau_3 = 3/2Z^*$.

Assume now that the server moved by SaW serves r_2 before r_1. Then it starts before $t = \tau_1$ (latest departure moment) and serves r_2 before $t = \tau_1 + (\tau_3 - \tau_2)$. At this moment, r_1 is already released; hence server serves it and returns to the origin before moment $\tau_1 + (\tau_3 - \tau_2) + (\tau_2 - \tau_1) + d(0, r_1) \leq 3/2\tau_3$.

Therefore, in both of the above cases, competitive ratio achieved is bounded above by $3/2$ and the proof of the proposition is complete. □

3 OL-QTSP on the halfline

In this section we consider the case in which the metric space is the half-line. For such case we are able to show a competitiveness lower bound and a matching upper bound. As we said before the off-line version of this problem is solvabe in polynomial time [19].

Theorem 5. *For the OL-QTSP problem, when the metric space is the halfline, for any $Q \geq 1$, there are no ρ-competitive algorithms where $\rho < 3/2$.*

Proof. We start the proof for the case in which $Q = 1$.

At time 1 one request in point 1 is released. Due to the speed constraint (Lemma 1), the server moved by the on-line algorithm cannot reach the request before time $t = 2$, otherwise the adversary might release immediately a request in the origin. So the on-line server cannot be back in the origin before time $t = 3$, while optimal completion time is 2.

For the case in which $Q > 1$, it is sufficient to use the above sequence in which we release Q requests at the same time in one point. □

We now present a 3/2-competitive algorithm that matches the above lower bound for the problem.

Algorithm 4 (SlowWalk (SW)) *The on-line server OL moves from the origin at half speed until the first time t_0 in which it is possible i) to serve a quota Q over the weights of the requests and ii) come back to the origin in no more than $t_0/2$; then it goes back to the origin at full speed.*

Theorem 6. *For the weighted OL-QTSP problem in the halfline, and for any value of the quota Q, the algorithm SW is ρ-competitive with $\rho = 3/2$.*

Proof. Basically, we have to prove that the on-line algorithm proceeds in such a way to conclude its tour not later than time $t_0 + t_0/2$ while the adversary uses at least time t_0. Let us first observe that if we consider $t_0 = Z^*$ all requests served by the adversary in an optimal solution have to lie between O and $t_0/2$ and have been released before time t_0. Therefore, at time $t_0 = Z^*$, SW can turn back because both requirements are met and it can conclude its tour by serving all requests on its way to the origin. The overall time it uses is $t_0 + t_0/2 = 3/2Z^*$ □

4 Application to the general OL-TSP problem

In the previous section we have observed that the server can gain in competitiveness by not moving too fast. Other results based on the same approach can be found in [17], where the notion of "non zealous" algorithm is defined, as opposed to the notion of "zealous" algorithm [7] that is, intuitevely, an algorithm that never remains idle when there is work to do. In this section we consider the different problem of the general On-Line TSP, in the *homing* version (H-OLTSP [3]) that is the version in which the server has to conclude its tour in the origin. By applying the same "slow is good" paradigm we show that, also in the H-OLTSP case, the "non-zealousness" of the server can help him in achieving a better competitiveness ratio (see also [17]).

We consider a metric space M and denote by Δ-TSP the metric TSP. In H-OLTSP each istance of requests denotes the on-line version; each instance of requests is a sequence $(p_i, t_i)_{i \geq 0}$ and the sequence (t_i) increases. Given such an instance, the traveler has to visit every point p_i (after t_i) in the least possible time, given the fact that, at each time, he only knows the already revealed points. The following on-line algorithm, named **Wait and Slowly Move (WaSM)**, uses a polynomial time r-approximation algorithm A for Δ-TSP.

Without loss of generality one can suppose that, for every Δ-TSP instance $(p_1, \dots, p_n)$ we have:

$$\forall t, d(0, p^*(t)) \leq \max_i (d(0, p_i))$$

Algorithm 5 (Wait and Slowly Move (WaSM))
1)$B \leftarrow (3+\sqrt{17})/2$
2) *as a new request is presented at time* t_0*, then:*
the on-line traveler comes back to the origin;
he waits until time $t_0(1+1/B)$*;*
he then follows the solution performed by A *with speed* $v(t)$ *such that:*
if $d(p^{OL}(t),0) \geq t/B$ ***then*** $v(t)=0$ *(ball-constraint)*
else $v(t)=1$*;*

Note that the ball-constraint means that, at every time t, the traveler is not allowed to be further than t/B from the origin.

Theorem 7. *If* $\Delta - TSP$ *admits a polynomial time* r*-approximation algorithm, then WaSM is a* $(r+c)$*-competitive algorithm for the* $H-OLTSP$*, where* $c = (1+\sqrt{17})/4 \simeq 1.2808$*.*

Proof. Observe that the value $B = (3+\sqrt{17})/2$ satisfies $(B-1)/2 = 1+1/B$. Let us first note that, as the on-line traveler is constrained always to remain at a distance at most t/B from the origin (ball-constraint), then, whenever a new request is presented at time t_0, he can be guaranteed to be at the origin at time $t_0(1+1/B)$.

Let then t be the time the last request is presented. At time $t(1+1/B)$ the on-line traveler is in the origin and begins to follow the solution provided by A. Let us consider the two cases in which the on-line traveler has to stop after time $t(1+1/B)$ (because of the ball-constraint) or not.

Case 1: $\exists t_1 > 0, p^{OL}(t(1+1/B)+t_1) = [t(1+1/B)+t_1]/B$. Let us denote by t' be the maximum such time, which means that $p^{OL}(t(1+1/B)+t') = [t(1+1/B)+t']/B$ and that $\forall \tau > t', p^{OL}(t(1+1/B)+\tau) = [t(1+1/B)+\tau]/B$ (from then on, the on-line traveler does not stop until the end of the tour).

Note that the distance covered between $t(1+1/B)$ and $t(1+1/B)+t'$ is at least $[t(1+1/B)+t']/B$, consequently

$$Z^{OL} \leq t(1+1/B) + t' + r \cdot Z^* - [t(1+1/B)+t']/B \quad (1)$$

On the other side the furthest request being at distance at least $[t(1+1/B)+t']/B$ from the origin, we have:

$$Z^* \geq 2[t(1+1/B)+t']/B \quad (2)$$

From relations 1 and 2 we deduce that, in this case, the related competitive ratio is no more than $r + (B-1)/2$.

Case 2: $\forall t_1 > 0, p^{OL}(t(1+1/B)+t_1) < [t(1+1/B)+t_1]/B$. Then,

$$Z^{OL} \leq t(1+1/B) + rZ^*$$

and

$$Z^* \geq t$$

Consequently a competitive ratio of $(1+1/B+r) = (B-1)/2+r$ is guaranteed, which concludes the proof. □

As a consequence of this theorem, if we use Christofides heuristic [12], that provides an approximation factor of $r = 1.5$, we can easily state the following result.

Corollary 1. *H-OLTSP admits a* 2.78*-competitive polynomial time algorithm.*

This result improves the 3-competitive result for this problem proved in [3].

5 Conclusions

In this paper the Quota Traveling Salesman Problem has been tackled from the on-line point of view. First we have shown that for general metric spaces the problem can be solved by means of a 2-competitive algorithm while no better competitive algorithm is possible. In particular since the lower bound result is proved on the real line no better on-line algorithm can be expected even for such particular case. On the contrary, if we consider the half-line metric space we showed a 3/2 competitive algorithm and a matching lower bound. A peculiarity of the OL-QTSP problem is that no zealous algorithm can achieve the best competitive performance.

Although for the classical Homing OL-TSP exponential time zealous algorithms have been shown to reach the best possible competitive ratio, we show a non zealous polynomial time algorithm wich achieves a 2.78 competitive ratio, outperforming the best known polynomial time algorithm for the same problem.

The problem to find a polynomial time algorithm for OL-QTSP with a good competitive ratio is still open.

References

1. S. Arya and H. Ramesh. A 2.5-factor approximation algorithm for the k-mst problem. *Information Processing Letters*, (65):117–118, 1998.
2. N. Ascheuer, S.O. Krumke, and J. Rambau. On-line dial-a-ride problems: Minimizing the completion time. In *Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science*, volume 1770 of *LNCS*, pages 639–650, 2000.
3. G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithms for the on-line travelling salesman. *Algorithmica*, (29):560–581, 1998.
4. B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 277–283, May 1995.
5. B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 1999.
6. E. Balas. The prize collecting traveling salesman problem. *Networks*, 19:621–636, 1989.
7. M. Blom, S.O. Krumke, W.E. de Paepe, and L. Stougie. The online-tsp against fair adversaries. *INFORMS Journal on Computing*, 13:138–148, 2001.

8. A. Blum, S. Chawla, D. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS-2003)*, Cambridge, MA, 2003. IEEE.
9. A. Blum, R. Ravi, and S. Vempala. A constant-factor approximation algorithm for the k-mst problem. In *ACM Symposium on Theory of Computing*, pages 442–448, 1996.
10. A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.
11. S.Y. Cheung and A. Kumar. Efficient quorumcast routing algorithms. In *Proceedings of INFOCOM '94*, volume 2, pages 840–855, Toronto, 1994.
12. N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, G.S.I.A. Carnegie Mellon University, 1976.
13. E. Feuerstein and L. Stougie. On-line, single server dial-a-ride problems. *Theoretical Computer Science*, (269):91–105, 2001.
14. A. Fiat and G. Woeginger, editors. *Online Algorithms: The State of the Art*, volume 1442 of *LNCS*. Springer, 1998.
15. N. Garg. A 3 factor approximation algorithm for the minimum tree spanning k vertices. In *Proceedings IEEE Foundations of Computer Science*, pages 302–309, 1996.
16. 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*, volume 2136 of *LNCS*, pages 487–499. Springer, 2001.
17. M. Lipmann. *On-Line Routing Problems*. PhD thesis, Technische Universiteit Eindhoven, 2003.
18. A. Regan S. Irani, X. Lu. On-line algorithms for the dynamic traveling repair problem. In *Proceedings of the 13th Symposium on Discrete Algorithms*, pages 517–524, 2002.
19. J.N. Tsitsiklis. Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22:263–282, 1992.

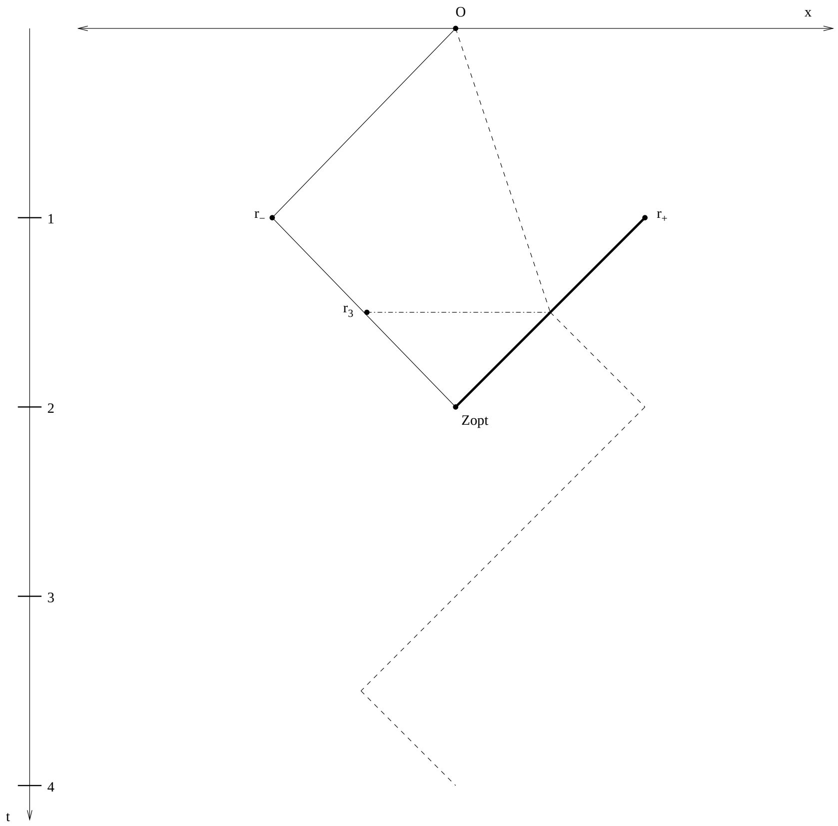

Fig. 1. A graphical example of the lower bound for general metric spaces. The plain line on the left is the tour of the optimal server, while the dashed line is the tour of the on-line server. As soon as the on-line server crosses the (virtual) bold line, a new request is released in the symmetric point (with respect to the origin). Any on-line tour that serves two requests can not end before time 4, while $Z^* = 2$