Information Processing Letters 83 (2002) 57–62

Information Processing Letters

www.elsevier.com/locate/ipl

Approximation algorithms for time-dependent orienteering

Fedor V. Fomin [a,1], Andrzej Lingas [b,*]

[a] *Graduiertenkolleg des PaSCo, Heinz Nixdorf Institut and University of Paderborn, Fürstenallee 11, D-33102, Paderborn, Germany*

[b] *Department of Computer Science, Lund University, Box 118, S-221 00, Lund, Sweden*

Received 1 December 2000; received in revised form 28 October 2001

Communicated by S. Albers

Abstract

The time-dependent orienteering problem is dual to the time-dependent traveling salesman problem. It consists of visiting a maximum number of sites within a given deadline. The traveling time between two sites is in general dependent on the starting time.

For any $\varepsilon > 0$, we provide a $(2+\varepsilon)$-approximation algorithm for the time-dependent orienteering problem which runs in polynomial time if the ratio between the maximum and minimum traveling time between any two sites is constant. No prior upper approximation bounds were known for this time-dependent problem.

Keywords: Time-depending orienteering; Traveling salesman; Approximation algorithms; Network design

1. Introduction

In the *orienteering* problem (see, e.g., [1,9]) a traveler wishes to visit a maximum number of sites (nodes) subject to given restrictions on the length of the tour. This problem can be regarded as the problem of traveling salesperson with restricted amount of resources (time, gasoline, etc.) wishing to maximize the number of visited sites. For this reason, the orienteering problem has been also called "the generalized traveling salesperson problem" or even as "the bank robber problem" [3,9]. Even if the ratio between the maximum and minimum amount of resources required for traveling between two sites is constantly bounded, the orienteering problem is MAX-SNP-hard simply because the correspondingly restricted traveling salesman problem is MAX-SNP-hard [5,14] (cf. [4,8,13]).

In this paper we consider a generalization of the orienteering problem which we term *time-dependent orienteering* (TDO, for short). In our generalization, the cost of traveling (time cost in our terminology) from any site to any other site in general depends on the start moment.

The orienteering problems considered in [1] are classified as the *path-orienteering*, *cycle-orienteering*, and even *tree-orienteering* problems depending on whether or not the network to be induced by the set of pairs of consecutive sites visited is supposed to have a form of a path, a cycle, or even a tree,

* Corresponding author.

E-mail addresses: fomin@upb.de (F.V. Fomin), Andrzej.Lingas@cs.lth.se (A. Lingas).

[1] Part of this work was done while the author was at the Centro de Modelamiento Matemático, Universidad de Chile and UMR 2071-CNRS, Chile. Supported by FONDAP. Also supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

0020-0190/01/$ – see front matter
PII: S0020-0190(01)00313-1

respectively. Additionally, one can wish to have one or more *roots*, which are 'essential' sites required to be visited. Following this classification we will refer to the cases of TDO problems without roots as *path* (*cycle, tree*)-TDO and to the cases with roots as *rooted-path* (*cycle, tree*)-TDO.

For illustration, consider the two following examples of possible applications of TDO.

Kinetic TSP [5,10]. There is given a set of targets and one robot (intercepter) with restricted amount of resources (e.g., fuel). The dynamic of targets is known, i.e., for each target one can specify its location at any discrete time moment. The problem is to find a program for the robot which allows it to hit as many targets as possible within a given time. This problem is an example of path-TDO. If there is a target specified to be intercepted then we have an example of rooted-path-TDO.

Time-Dependent Maximum Scheduling Problem (TDMS) [15]. There is given a set of tasks for a single machine. The execution of any task can start at any discrete moment and its execution time depends on the starting moment. The problem is to find a schedule for the machine maximizing the number of tasks completed within a given time period. This problem is equivalent to Job Interval Selection Problem studied by Spieksma [15]. It can be interpreted as a special case of TDO where the time of traveling from a site a to a site b does not depend on the site b and is interpreted as the execution time of the task corresponding to a. The Web Searching Problem studied by Czumaj et al. [7] yields the following motivation for TDMS. Assume that there is a central computer which is being used to collect all the information stored in a number of web documents, located in various sites. The information is gathered by scheduling a number of consecutive client/server connections with the required web sites, to collect the information page by page. The loading time of any particular page from any site can vary at different times, e.g.,, the access to the page is much slower in peak hours than in off-peak hours. We wish to download the maximum number of pages within a given period of time.

1.1. Main results

An algorithm is said to be a *c-approximation* algorithm for a maximization problem P if for any instance of P it yields a solution whose value is at least $1/c$ times the optimum.

Let n be the number of input sites and let k be the ratio between the maximum and minimum time required for traveling between two sites.

We present $(2+\varepsilon)$-approximation algorithms for path-TDOs and cycle-TDOs running in time

$$\mathrm{O}\left(\left(2k^2\left\lceil\frac{2+\varepsilon}{\varepsilon}\right\rceil\right)!\,kn^{2k^2(\frac{2+\varepsilon}{\varepsilon})+1}\right).$$

In the corresponding rooted cases the time complexity increases by the multiplicative factor $\mathrm{O}(kn/\varepsilon)$. These bounds immediately carry over to the corresponding time-independent special cases, i.e., the unrooted and rooted, path-orienteering problems and cycle-orienteering problems. Our algorithm is the first constant-factor approximation algorithm for TDO with $k=\mathrm{O}(1)$ running in polynomial time. Although for large k, our algorithm can be hardly claimed to be practical because of its fairly high running time, it suggests that practical and efficient algorithms might be possible.

1.2. Related results

The authors are not familiar with any explicit prior approximation algorithms for time-dependent orienteering (TDO). Of course, if the ratio between the maximum and minimum distance is k then any approximation algorithm is a k-approximation one.

For the Time-Dependent Maximum Scheduling Problem (TDMS) which can be interpreted as a special case of TDO, a simple greedy 2-approximation algorithm running in time $\mathrm{O}(mt)$, where m is the number of available tasks and t is the deadline, follows from Spieksma's algorithm [15] for Job Interval Selection Problem. Also, it follows from the same work [15] that TDMS is MAX-SNP-hard.

As for the "classical", i.e., time-independent, orienteering problem, Awerbuch et al. proved that a c-approximation algorithm to the so-called k-traveling salesperson problem, asking for a shortest cycle visiting k sites (k-TSP), yields a $2c$-approximation algorithm for the orienteering problem [2]. This result combined with known approximation results for k-TSP yields a 6-approximation algorithm for the orienteering problem in metric spaces and a $(2+\varepsilon)$-approximation algorithm in the Euclidean plane. The latter result has been subsumed by Arkin et al. who presented 2-approximation algorithms for several variants of the orienteering problem in the plane [1]. More recently, Broden has designed a $4/3$-approximation algorithm for the very special case of the orienteering problem where the pairwise distances are constrained to $\{1,2\}$ [5]. Note here that the recent lower bounds on the constant approximation factor for the analogously restricted traveling salesperson problem [4] easily carry over to the aforementioned special case of the orienteering problem.

For an interesting review of results related to the orienteering problem, including several variants of the traveling salesperson problem, the reader is referred to [1].

The recent works by Hammar and Nilsson [10] and Broden [5] contain a number of inapproximability and approximability results on various restrictions of the problem dual to TDO, i.e., the time-dependent traveling salesperson problem.

2. Formal definition of TDO

For a given set S of n sites, a time-travel function $l: S \times S \times \mathbb{N} \cup \{0\} \to \mathbb{R}^+$ and a deadline t, the *salesperson's tour visiting a subset T of m sites* is a sequence of triples

$$(s_1, t_1^+, t_1^-), (s_2, t_2^+, t_2^-), \ldots, (s_m, t_m^+, t_m^-)$$

such that

(1) for $i \in \{1,2,\ldots,m\}$, $t_i^+, t_i^- \in \mathbb{N} \cup \{0\}$;

(2) $T = \{s_1, s_2, \ldots, s_m\}$;

(3) $0 = t_1^+ \leqslant t_1^- \leqslant t_2^+ \leqslant \cdots \leqslant t_m^+ \leqslant t_m^- = t$;

(4) for each $i \in \{1,2,\ldots,m-1\}$, $t_{i+1}^+ - t_i^- = l(s_i, s_{i+1}, t_i^-)$.

It is useful to interpret the moment t_i^- as the moment when salesperson leaves the site s_i and t_i^+ as the moment when salesperson enters s_i. So $t_{i+1}^+ - t_i^-$ is the time spent in travel from s_i to s_{i+1} and $t_i^- - t_i^+$ is the time the salesperson stays in s_i (importantly, the traveler is allowed to stay at any site any time). The *path* (*or cycle*) *time-dependent orienteering problem* is to find an open (closed, respectively) tour visiting maximum number of sites within the time t.

Note that the classical orienteering problem [1] is a special case of TDO where for any sites a, b, the travel time from a to b is time-independent, i.e., $l(s_a, s_b, t') = l(s_a, s_b, t'')$ for any $t', t'' \in [0,t]$.

3. Main procedure and algorithms

We may assume without loss of generality that the minimum travel time between two sites, $\min_{s,s' \in S, t' \in [0,t]} l(s, s', t')$, is 1.

For a nonnegative integer q and a positive integer $i \leqslant \lceil t/q \rceil$, we shall denote by $I_i(q)$ the subinterval $[q(i-1), \min\{qi-1, t\}]$ of $[0,t]$. For a given set S of sites, a *q-partial salesperson's tour* is a sequence Q of triples (s_l, t_l^+, t_l^-), $s_l \in S$, $t_l^+, t_l^- \in [0,t]$, such that for every time interval $I_i(q)$, $1 \leqslant i \leqslant \lceil t/q \rceil$, the subsequence $(s_p, t_p^+, t_p^-)_{t_p^+, t_p^- \in I_i(q)}$ of Q is a (*salesperson's*) *path-tour*, i.e., an open tour visiting all sites in the interval. In other words, q-partial tour induces a path-tour for each time interval $I_i(q)$, $i \in \{1,2,\ldots,\lceil t/q \rceil\}$, but in general it is not a tour.

The following simple procedure is the heart of our algorithms for TDO.

Procedure *Greedy*(S,q,t)
INPUT: Set S of n sites, integer q, deadline t;
OUTPUT: q-partial tour.
(1) $T \leftarrow S$;
(2) **for** $i = 1, 2, \ldots, \lceil t/q \rceil$ **do**
let T_i be a maximum cardinality subset of T that can be visited in the time interval $I_i(q)$ by a path-tour;
compute a path-tour visiting T_i in the time interval $I_i(q)$;
$T \leftarrow T \setminus T_i$

Obviously, the set T_i can be found in $O(q!n^q q)$ time by considering all possible choices of a subset of T containing at most q elements and then applying the straightforward $O(q!q)$-time brute force method for the time-dependent traveling salesperson problem on the subset. Hence the overall time complexity of *Greedy*(S,q,t) is $O(q!tn^q)$.

Let k be the maximum time needed to travel between two sites, i.e., $k = \max_{s,s' \in S, t' \in [0,t]} l(s, s', t')$.

Note that by our initial assumption k is also the ratio between the maximum and minimum time required for traveling between two sites.

Algorithm *GreedyPath*(S, q, t)
INPUT: Set S of n sites, integer q, deadline t;
OUTPUT: path-tour.
(1) Run procedure *Greedy*(S, q, t);
(2) **for** $i = 1, \dots, \lceil t/q \rceil - 1$ **do**
Remove from the set of visited sites all sites visited in the time interval $[qi - k/2, qi + k/2]$;
Glue the obtained subtours by forcing the salesperson to go from the last visited site in the time interval $[q(i-1) + k/2, qi - k/2]$ to the first visited site in the time interval $[qi + k/2, q(i+1) - k/2]$

Algorithm *GreedyCycle*(S, q, t) is obtained from *GreedyPath*(S, q, t) by closing the path-route near its endpoints, i.e., by forcing the salesperson to go from the last visited site in the time interval $s_{\lceil t/q \rceil - 1}$ to the first visited site from the time interval $[k, q - k/2]$.

4. Approximation analysis

Lemma 1. *Let p be the maximum number of different sites that can be visited by a q-partial tour. Then the number of sites participating in the q-partial tour produced by Greedy(S, q, t) is at least $\lceil p/2 \rceil$.*

Proof. Let $W \subseteq S$ be a set of p sites that can be visited by a q-partial tour Q_{opt} and let V be the set of sites returned by *Greedy*(S, q, t). Denote by V_i the set of sites visited by *Greedy*(S, q, t) in the time interval $[0, iq)$ and by W_i the subset of W visited by Q_{opt} in the time interval $[0, iq)$. For $1 \leqslant i \leqslant \lceil t/q \rceil$, let $X_i := W \setminus (V_i \cup W_i)$, i.e., X_i consists of the sites in W that have been visited by neither *Greedy*(S, q, t) nor Q_{opt} in the time interval $[0, qi)$.

We claim that for each $1 \leqslant i \leqslant \lceil t/q \rceil$,

$$|X_i| \geqslant p - 2|V_i|. \tag{1}$$

Since $X_{\lceil t/q \rceil} = \emptyset$, this claim implies the lemma. We prove it by induction on i.

The procedure *Greedy*(S, q, t) finds the maximum number of sites than can be visited within time interval $I_1(q)$. Hence $|V_1| \geqslant |W_1|$ and $|V_1 \cup W_1| \leqslant 2|V_1|$ hold. By definition, $|X_1| = p - |V_1 \cup W_1| \geqslant p - 2|V_1|$ holds. Thus, for $i = 1$ the inequality (1) is true. Suppose that (1) is true for all $r \in \{1, \dots, i-1\}$.

We claim that

$$X_{i-1} \setminus X_i = (V_i \cup W_i) \setminus (V_{i-1} \cup W_{i-1}). \tag{2}$$

In fact, for every $u \in W$, we have

$$\begin{aligned} u \in X_{i-1} \setminus X_i &\Leftrightarrow u \in X_{i-1} \wedge u \notin X_i \\ &\Leftrightarrow u \in W \setminus (V_{i-1} \cup W_{i-1}) \wedge \\ &\quad\; u \notin W \setminus (V_i \cup W_i) \\ &\Leftrightarrow u \in (V_i \cup W_i) \setminus (V_{i-1} \cup W_{i-1}). \end{aligned}$$

By (2),

$$(X_{i-1} \setminus X_i) \cap V_i \subseteq V_i \setminus V_{i-1}. \tag{3}$$

The set of sites $V_i \setminus V_{i-1}$ is chosen by *Greedy*(S, q, t) at the ith step. Therefore, $|V_i \setminus V_{i-1}|$ is the maximum number of sites from $S \setminus V_{i-1}$ that can be visited within the time interval $I_i(q)$. Eq. (2) implies

$$\big((X_{i-1} \setminus X_i) \cap W_i\big) \cap V_{i-1} = \emptyset$$

and we conclude that

$$\big|(X_{i-1} \setminus X_i) \cap W_i\big| \leqslant |V_i \setminus V_{i-1}|. \tag{4}$$

It follows from (2), (3) and (4) that

$$\begin{aligned} &|X_{i-1} \setminus X_i| \\ &\quad \leqslant \big|(X_{i-1} \setminus X_i) \cap V_i\big| + \big|(X_{i-1} \setminus X_i) \cap W_i\big| \\ &\quad \leqslant 2|V_i \setminus V_{i-1}|. \end{aligned} \tag{5}$$

Since $X_i \subseteq X_{i-1}$ and $V_i \supseteq V_{i-1}$, we have that

$$|X_i| = |X_{i-1}| - |X_{i-1} \setminus X_i| \tag{6}$$

and

$$|V_i| = |V_{i-1}| + |V_i \setminus V_{i-1}|. \tag{7}$$

Combining (5), (6), (7) with the induction assumption $|X_{i-1}| \geqslant p - 2|V_{i-1}|$, we obtain

$$\begin{aligned} |X_i| &= |X_{i-1}| - |X_{i-1} \setminus X_i| \\ &\geqslant p - 2|V_{i-1}| - 2|V_i \setminus V_{i-1}| \\ &= p - 2|V_i|, \end{aligned}$$

and (1) follows. □

Theorem 2. *For any $\varepsilon > 0$, if the path- and cycle-TDO for n sites admit $(2+\varepsilon)$-approximation algorithms running in time* $O((2k^2\lceil \frac{2+\varepsilon}{\varepsilon} \rceil)! k n^{2k^2(\frac{2+\varepsilon}{\varepsilon})+1})$.

Proof. Let S be a set of n sites. Suppose that an optimal solution to the path- (cycle-) TDO visits m sites. We may assume without loss of generality. that $m \geqslant t/k$ (otherwise there is a trivial algorithm finding the exact solution).

Take $q = 2k^2((2+\varepsilon)/\varepsilon)$. By Lemma 1, the procedure *Greedy*(S, q, t) outputs a q-partial tour visiting at least $m/2$ sites. Consequently, each of the algorithms *GreedyPath*(S, q, t), *GreedyCycle*(S, q, t) outputs a tour visiting at least

$$\frac{m}{2} - \frac{kt}{q} = \frac{1}{2}\left(m - \frac{2kt}{q}\right) \geqslant \frac{m}{2}\left(1 - \frac{2k^2}{q}\right) = \frac{m}{2+\varepsilon}$$

sites. Hence, *GreedyPath*(S, q, t) and *GreedyCycle*(S, q, t) are $(2+\varepsilon)$-approximation algorithms.

By the remark on the time complexity of procedure *Greedy*(S, q, t), both algorithms run in time $\mathrm{O}((2k^2\lceil\frac{2+\varepsilon}{\varepsilon}\rceil)!\,tn^{2k^2(\frac{2+\varepsilon}{\varepsilon})})$ and the assumption $n \geqslant m \geqslant t/k$ implies the complexity bound in the theorem thesis. □

We can trivially model the time-independent path- and cycle-orienteering problems as special cases of TDO by setting the traveling time between two sites to the distance between them.

Corollary 3. *For any $\varepsilon > 0$, path- and cycle-orienteering time-independent problems for n sites admit $(2+\varepsilon)$-approximation algorithms running in time $\mathrm{O}((2k^2\lceil\frac{2+\varepsilon}{\varepsilon}\rceil)!\,kn^{2k^2(\frac{2+\varepsilon}{\varepsilon})+1})$, where the distance between the furthest site pair is at most k times greater than that between the closest pair.*

5. Extensions

We can easily extend our technique to include the rooted case of the path- and cycle-TDO (cases where 'essential' sites are required to be visited).

Analogously to Theorem 2, we can obtain the following result.

Theorem 4. *For any $\varepsilon > 0$, if the distance between the furthest site pair is $\mathrm{O}(1)$ times greater than that between the closest pair, the rooted path- and cycle-TDO for n sites admit $(2+\varepsilon)$-approximation algorithms running in polynomial time.*

Our technique also can be extended to include orienteering variants of many other optimizations problems (e.g., tree-orienteering) as well variants with parallel travelers. It can be also used in the design of efficient approximation algorithms for time-dependent bicriteria network optimization problems (see [12] and the last chapter in [6]). Finally, it can be applied to derive approximative solutions to the Budget Prize Collecting Steiner Tree problem (see [11]) and its time dependent variant.

Acknowledgements

We are grateful to unknown referees for valuable comments.

References

[1] E.M. Arkin, J.S.B. Mitchell, G. Narasimhan, Resource-constrained geometric network optimization, in: Proceedings 14th ACM Symposium on Computational Geometry, June 6–10, 1998, pp. 307–316.

[2] B. Awerbuch, Y. Azar, A. Blum, S. Vempala, Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesman, in: Proceedings 27th Annual ACM Symp. Theory Comput. (STOC'95), 1995, pp. 277–283.

[3] E. Balas, The prize collecting traveling salesperson problem, Networks 19 (1989) 621–636.

[4] H.J. Bockenhauer, J. Hromkovic, R. Klasing, S. Seibert, W. Unger, An improved lower bound on the approximability of metric TSP and approximation algorithms for the TSP with sharpened triangle inequality, Inform. Process. Lett. 75 (3) (2000) 133–138.

[5] B. Broden, Time Dependent Traveling Salesman Problem, M.Sc. thesis, Department of Computer Science, Lund University, Sweden, 2000.

[6] J. Cheriyan, R. Ravi, Approximation algorithms for network problems, Manuscript, 1998; http://www.gsia.cmu.edu/andrew/ravi/home.html.

[7] A. Czumaj, I. Finch, L. Gasieniec, A. Gibbons, P. Leng, W. Rytter, M. Zito, Efficient web searching using temporal factors, in: F. Dehne, A. Gupta, J.-R. Sack, R. Tamassia (Eds.), Proceedings of the 6th Workshop on Algorithms and Data Structures (WADS), Lecture Notes in Comput. Sci., Vol. 1663, Springer, Berlin, 1999, pp. 294–305.

[8] L. Engebretsen, An explicit lower bound for TSP with distances one and two, in: Proceedings 16th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci., Springer, Berlin, 1999, pp. 373–382.

[9] B.G. Golden, L. Levy, R. Vohra, The orienteering problem, Naval Res. Logist 34 (1991) 307–318.

[10] M. Hammar, B. Nilsson, Approximation results for kinetic variants of TSP, in: Proceedings of the 26th Interna-

tional Colloquium on Automata, Languages, and Programming (ICALP'99), Lecture Notes in Comput. Sci., Vol. 1644, Springer, Berlin, 1999, pp. 392–401.

[11] D.S. Johnson, M. Minkoff, S. Phillips, The Prize Collecting Steiner Tree Problem: Theory and practice, in: Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms, 2000, pp. 760–769.

[12] M.V. Marathe, R. Ravi, R. Sundaram, S.S. Ravi, D.J. Rosenkrantz, H.B. Hunt III, Bicriteria network design problems, J. Algorithms 28 (1998) 142–171.

[13] C.H. Papadimitriou, S. Vempala, On the approximability of the traveling salesman problem, in: Proceedings of the 32nd ACM STOC, 2000, pp. 126–133.

[14] C.H. Papadimitriou, M. Yannakakis, The traveling salesman problem with distances one and two, Math. Oper. Res. 18 (1) (1993) 1–11.

[15] F.C.R. Spieksma, On the approximability of an interval scheduling problem, J. Scheduling 2 (1999) 215–227.