Approximation Algorithms for the Arc Orienteering Problem ☆

D. Gavalas[a,f], C. Konstantopoulos[b,f], K. Mastakas[c,f,*], G. Pantziou[d,f], N. Vathis[e,f]

[a] *Department of Cultural Technology and Communication, University of Aegean, Greece*
[b] *Department of Informatics, University of Piraeus, Greece*
[c] *School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Greece*
[d] *Department of Informatics, Technological Educational Institution of Athens, Greece*
[e] *School of Electrical and Computer Engineering, National Technical University of Athens, Greece*
[f] *Computer Technology Institute & Press "Diophantus", Patras, Greece*

Abstract

In this article we present approximation algorithms for the Arc Orienteering Problem. Specifically, we give a $O(\log^2 m)-$approximation algorithm in directed graphs, where m is the number of arcs, while in undirected graphs, we obtain a $(6 + \epsilon + o(1))-$approximation algorithm for the general case and a $(4 + \epsilon)-$approximation algorithm for instances with unit profits. Moreover, we obtain approximation algorithms for the Mixed Orienteering Problem.

Keywords: Arc Orienteering Problem, Orienteering Problem, Mixed Orienteering Problem, Approximation Algorithms, NP-hardness

☆This work has been supported by the EU FP7/2007-2013 (DG CONNECT.H5-Smart Cities and Sustainability), under Grant Agreement no. 288094 (project eCOMPASS) and the EU FP7 Collaborative Project MOVESMART (Grant Agreement No 609026).

*Corresponding author

Email addresses: dgavalas@aegean.gr (D. Gavalas), konstant@unipi.gr (C. Konstantopoulos), kmast@math.ntua.gr (K. Mastakas), pantziou@teiath.gr (G. Pantziou), nvathis@softlab.ntua.gr (N. Vathis)

1. Introduction

The Arc Orienteering Problem (AOP) is a single route arc routing problem with profits defined as follows [1]. Given a quadruple $(G = (V, A), t, p, B)$ where $G = (V, A)$ is a directed graph with $V = \{s = u_1, u_2, \ldots, u_n = l\}$ its set of nodes and A its set of arcs, $t : A \rightarrow \mathbb{R}^+$ i.e., each arc $a \in A$ is associated with a nonnegative travel time t_a, $p : A \rightarrow \mathbb{R}^+$ i.e., each arc is associated with a nonnegative profit p_a, and a nonnegative time budget B, the goal is to find an $s - l$ walk with length at most B with the maximum collected profit from the traversed arcs. Note that while the travel cost associated with an arc is paid each time the arc is traversed by the walk, its profit is collected only once. The AOP is the arc routing version of the Orienteering Problem (OP) [2], where the nodes (instead of the arcs) are associated with profits. The OP is NP-hard [2] and APX-hard [3]. The algorithmic research relevant to OP and its extensions is very extensive [4, 5]. Considering approximability, the best known approximation algorithms are the $(2 + \epsilon)$-approximation and the $O(\log^2 n)$-approximation algorithms in undirected and directed graphs, respectively, proposed by Chekuri et al. [6] and the $O(\frac{\log^2 n}{\log \log n})$-approximation algorithm in asymmetric metric spaces proposed in [7].

Contrary to the OP, very limited body of literature deals with the AOP and its extensions. A metaheuristic algorithm for the AOP was proposed by Souffriau et al. [1] and a branch-and-cut and a matheuristic approach for the extension of the AOP to multiple tours were proposed in [8] and [9], respectively. In this article we study the AOP in directed and undirected graphs. In Section 2 we prove that AOP is NP-hard and propose a $O(\log^2 m)-$approximation algorithm for the AOP in directed graphs, where

m is the number of arcs of the graph, using the $O(\log^2 n)-$approximation algorithm for the OP in directed graphs proposed in [6]. In Section 3 we present a $(6+\epsilon+o(1))-$ approximation algorithm for the AOP in undirected graphs and a $(4+\epsilon)-$ approximation algorithm for the unweighted version of the problem, using the $(2+\epsilon)-$ approximation algorithm for the unweighted version of the OP, also proposed in [6]. Moreover, we give approximation algorithms for the Mixed Orienteering Problem (MOP), the combination of OP and AOP, by reducing it to the AOP.

2. The Arc Orienteering Problem

In this section we prove that AOP is NP-hard and propose an approximation algorithm for the problem by reducing it to the OP in directed graphs.

Theorem 1. *The AOP is NP-hard*

Proof. We reduce the Knapsack problem [10] to the AOP. A Knapsack instance contains a set of objects $O = \{o_1, o_2, \ldots, o_n\}$ such that each o_i has a weight w_i and a profit p_i, a limit W in the total weight of objects that can be picked, and a target profit P. The Knapsack instance can be reduced to an AOP instance containing a star graph G with a central node s connected to each node o_i representing an object, and vice versa. Both (s, o_i) and (o_i, s) have travel time (profit) equal to $\frac{w_i}{2}$ ($\frac{p_i}{2}$) and the time budget is equal to W. Then, the Knapsack instance is a "yes" instance if and only if the solution of the AOP instance has profit at least P. □

Theorem 2. *An $f(n)-$approximation algorithm for the OP in directed graphs, where n is the number of nodes, yields an $f(m+2)-$approximation algorithm for the AOP, where m is the number of arcs.*

Proof. Given an instance of the AOP $(G = (V, A), t, p, B)$, $|A| = m$, we construct an instance of the OP in the directed network N as follows: We first define $N = (V', A')$ such that $V' = \{s, l\} \cup \{(u, v) : (u, v) \in A\}$, $|V'| = n' = m + 2$, and $A' = \{(s, (s, u)), ((u, l), l) : (s, u) \text{ and } (u, l) \in A\} \cup \{((u, v), (v, w)) : (u, v), (v, w) \in A\}$. The travel times of the arcs in N are defined as follows: $t'_{((u,v),(v,w))} = \frac{t_{(u,v)}+t_{(v,w)}}{2}$, $t'_{(s,(s,u))} = \frac{t_{(s,u)}}{2}$ and $t'_{((u,l),l)} = \frac{t_{(u,l)}}{2}$. We also define the profit of each node (u, v) to be equal to $p'_{(u,v)} = p_{(u,v)}$ and set the time budget of the instance as $B' = B$. It is easy to see that a solution of the AOP instance yields a solution of the OP instance of equal total profit and length and vice versa. □

The theorem implies an approximation algorithm for the AOP. Using the $O(\log^2 n)-$approximation algorithm for the OP in directed graphs by Chekuri et al. [6] we obtain a $O(\log^2 m)-$approximation algorithm for the AOP. If all the travel costs of the arcs are greater than zero, we obtain a $O(\frac{\log^2 m}{\log \log m})-$approximation, by applying the metric closure in the constructed OP instance and use the algorithm by Nagarajan and Ravi [7].

3. Approximation Algorithms for the AOP in Undirected Graphs

In this section we study the AOP in undirected graphs. A similar reduction to the one in Theorem 1 shows that the problem is NP-hard. We obtain a constant factor approximation algorithm for the problem by reducing it to the Unweighted OP (UOP), the restriction of the OP with unit profits over the nodes. First, we reduce the AOP to the special case with polynomially bounded positive integer profits using a similar technique with [6].

Lemma 3. *A $\rho-$approximation algorithm for the AOP in undirected graphs with polynomially bounded positive integer profits yields a $(\rho + o(1))-$ approximation algorithm for the AOP in undirected graphs.*

Proof. Given an instance $I = (G = (V, E), t, p, B)$ of the AOP, we shall construct an instance $I' = (G' = (V', E'), t, p', B)$ with polynomially bounded positive integer profits over its edges. First, we guess the edge of highest profit ($p_{\max}$) in the optimal walk and remove all higher profit edges. Then, we set $p'_e = \lfloor \frac{n^3 p_e}{p_{\max}} \rfloor + 1$ for each edge e. Then a feasible walk W, consisting of the distinct edges $e_1, e_2, \ldots, e_k$ has profit equal to $\text{profit}(W) = \sum_{j=1}^{k} p_{e_j}$ in I and $\text{profit}'(W) = \sum_{j=1}^{k} p'_{e_j} > \frac{n^3 \text{profit(W)}}{p_{\max}}$ in I'. Hence, $\text{OPT}' > \frac{n^3}{p_{\max}}\text{OPT}$, where OPT(OPT$'$) is the optimum in $I(I')$. On the other hand, $\text{profit}'(W) = \sum_{j=1}^{k} p'_{e_j} \le \sum_{j=1}^{k} (\frac{n^3 p_{e_j}}{p_{\max}} + 1) \implies \frac{n^3}{p_{\max}}\text{profit}(W) \ge \text{profit}'(W) - m \ge \text{profit}'(W) - m\frac{\text{OPT}}{p_{\max}}$, so $\text{profit}(W) \ge \frac{p_{\max}}{n^3}\text{profit}'(W) - \frac{m}{n^3}\text{OPT}$. Using a $\rho-$approximation algorithm for AOP in undirected graphs with polynomially bounded positive integer profits, we obtain a walk W with profit $\text{profit}'(W) \ge \frac{\text{OPT}'}{\rho}$, so $\text{profit}(W) \ge \frac{1}{\rho}\frac{p_{\max}}{n^3}\text{OPT}' - \frac{m}{n^3}\text{OPT} > (\frac{1}{\rho} - \frac{m}{n^3})\text{OPT}$. □

Theorem 4. *A $\rho-$approximation algorithm for the UOP in undirected graphs yields a $3\rho-$approximation algorithm for the AOP in undirected graphs with polynomially bounded positive integer profits.*

Proof. Given an instance of the AOP, each edge e for which the shortest path from s to l passing through e exceeds the time budget is removed from the graph. Then we construct an instance of UOP splitting each edge $\{u, v\}$ into $p_{uv} + 1$ edges as follows: Each node of the AOP instance is a node of the UOP instance and for each edge $\{u, v\}$ of the AOP instance, the UOP instance includes the auxiliary nodes $\{u,v\}_1, \{u,v\}_2, \cdots, \{u,v\}_{p_{uv}}$ and the edges $\{u, \{u,v\}_1\}$, $\{\{u,v\}_1, \{u,v\}_2\}$, $\cdots$, $\{\{u,v\}_{p_{uv}-1}, \{u,v\}_{p_{uv}}\}$, $\{\{u,v\}_{p_{uv}}, v\}$. The travel times of $\{u, \{u,v\}_1\}$ and $\{\{u,v\}_{p_{uv}}, v\}$ in the UOP instance are set $\frac{t_{\{u,v\}}}{2}$, while the length of each edge $\{\{u,v\}_i, \{u,v\}_{i+1}\}, i = 1, 2, \ldots, p_{uv-1}$ is set zero. The time budget of the UOP instance is the same with the AOP instance.

A solution of the AOP instance yields a solution of the UOP instance of the same length and at least the same profit, replacing any edge $\{u,v\}$ by the sequence $(u, \{u,v\}_1, \{u,v\}_2, \cdots, \{u,v\}_{p_{uv}}, v)$. Hence $\mathrm{OPT}_{\mathrm{AOP}} \leq \mathrm{OPT}_{\mathrm{UOP}}$.

On the other hand, any solution of the constructed UOP instance yields a solution of the AOP instance with at least one third of its profit. We consider a sequence of nodes of the form $(u, \{u,v\}_1, \{u,v\}_2, \cdots, \{u,v\}_{p_{uv}}, v)$ as an *appropriate* segment, i.e. a segment that represents the traversal of the edge $\{u,v\}$ of the AOP instance, while we consider a sequence of the form $(u, \{u,v\}_1, \cdots, \{u,v\}_{i-1}, \{u,v\}_i, \{u,v\}_{i-1} \cdots, \{u,v\}_1, u)$ as an *inappropriate* segment, i.e. a segment that represents the partial traversal of the edge $\{u,v\}$ of the AOP instance. For each inappropriate segment $(u, \{u,v\}_1, \ldots, \{u,v\}_{i-1}, \{u,v\}_i, \{u,v\}_{i-1} \cdots, \{u,v\}_1, u)$ we may consider that $i = p_{uv}$, otherwise the segment can be extended to the equal length and higher profit segment with $i = p_{uv}$.

In a UOP solution, let p_{AS} the profit gained by the appropriate segments and p_{IS} the profit gained by the inappropriate segments (the number of auxiliary nodes visited in them), then the total profit of the solution p_{TOT} equals to $p_{\mathrm{AS}} + p_{\mathrm{IS}}$. If all segments are appropriate, the re-transformation to an AOP solution is done by replacing the segments by their representing edges, yielding a solution of at least half the profit of the UOP solution. If however inappropriate segments exist, we may replace some of them with their representing edge traversed in both directions. We will replace the inappropriate segment $(u, \{u,v\}_1, \cdots, \{u,v\}_{p_{uv}-1}, \{u,v\}_{p_{uv}}, \{u,v\}_{p_{uv}-1}, \cdots, \{u,v\}_1, u)$ with the sequence of nodes (u, v, u) in the AOP instance.

Let IS$= \{s_1, s_2, \cdots, s_k\}$ be the set of inappropriate segments of the UOP solution. Let also $t_1, t_2, \ldots, t_k$ be the travel times spent on these segments and $p_1, p_2, \cdots, p_k$ be the profits collected by traversing them ($\sum_{i=1}^{k} p_i = p_{\text{IS}}$). A subset of IS, RS$= \{s_{a_1}, s_{a_2}, \ldots, s_{a_m}\}, m \leq k$, with $\sum_{j=1}^{m} 2t_{a_j} \leq \sum_{i=1}^{k} t_i$ will be called a *replacable subset*. Then RS is a *maximal replacable subset* for the given time constraint, if the insertion of any other segment (a_{m+1}) into the set would violate the time budget i.e. $\sum_{j=1}^{m+1} 2t_{a_j} > \sum_{i=1}^{k} t_i$. Consider a maximal replacable subset MRS of segments. We distinguish between the following cases: (i) $p_{\text{MRS}} \geq \frac{p_{\text{IS}}}{3}$, where p_{MRS} is the total profit of MRS. Then a solution for the AOP is obtained consisting of the edges represented by the appropriate segments (contributing at least $\frac{p_{\text{AS}}}{2}$ profit) and the sequences that replace the segments in MRS (contributing p_{MRS} profit), hence at least a third of UOP solution's profit. (ii) $p_{\text{MRS}} < \frac{p_{\text{IS}}}{3}$. In this case the set $\text{IS}\backslash\text{MRS} = \text{MRS}^c$, has profit at least two thirds of the total profit of the IS. Then if MRS^c has at least two elements, we remove the segment with the lowest profit, creating a replacable subset with profit at least a third of IS and we apply the procedure discussed previously. Otherwise, if MRS^c has only one segment $s_1 = (u, \{u,v\}_1, \ldots, \{u,v\}_{p_{uv}}, \ldots, \{u,v\}_1, u)$, then if $p_1 < \frac{p_{\text{TOT}}}{3}$ we apply the same technique, since $p_{\text{AS}} + p_{\text{MRS}} \geq \frac{2p_{\text{TOT}}}{3}$. Otherwise ($p_1 \geq \frac{p_{\text{TOT}}}{3}$), obtaining the shortest path from s to l through edge $\{u,v\}$ yields an AOP solution of at least a third of UOP solution's profit. Hence, the theorem is proved. □

Using Lemma 3, Theorem 4 and the $(2+\epsilon)-$approximation algorithm for the UOP by Chekuri et al. [6] we obtain a $(6+\epsilon+o(1))-$approximation algorithm for the AOP in undirected graphs with execution time $n^{O(\frac{1}{\epsilon^2})}$.

The unweighted version of AOP (**UAOP**) in undirected graphs is the restriction of the problem where all edges have profit equal to 1. Similarly to Theorem 4, but picking the half shortest inappropriate segments to replace, a $\rho-$approximation algorithm for the UOP in undirected graphs yields a $2\rho-$approximation algorithm for the UAOP in undirected graphs. Using Chekuri et al.'s algorithm we obtain a $(4+\epsilon)-$approximation algorithm.

The Mixed Orienteering Problem (**MOP**) [5, 11], is the combination of the OP and the AOP, where both nodes and arcs are associated with profits. MOP can be reduced to AOP as follows: For each node u, add a dummy node u' and the arcs $(u,u'),(u',u)$ with zero travel cost and profit equal to $\frac{p_u}{2}$ and then remove the profit from u. It is easy to see that any approximation algorithm for the AOP yields an approximation algorithm for the MOP.

[1] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, D. Van Oudheusden, The planning of cycle trips in the province of east flanders, Omega 39 (2) (2011) 209 – 213.

[2] B. L. Golden, L. Levy, R. Vohra, The orienteering problem, Naval Research Logistics (NRL) 34 (3) (1987) 307–318.

[3] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, M. Minkoff, Approximation algorithms for orienteering and discounted-reward tsp, in: Proc. of the 44th IEEE FOCS, 2003, pp. 46 –55.

[4] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, A survey on algorithmic approaches for solving tourist trip design problems, J. of Heuristics 20 (3) (2014) 291–328.

[5] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey, EJOR 209 (1) (2011) 1 – 10.

[6] C. Chekuri, N. Korula, M. Pál, Improved algorithms for orienteering and related problems, in: Proc. of ACM-SIAM SODA, 2008, pp. 661–670.

[7] V. Nagarajan, R. Ravi, The directed orienteering problem, Algorithmica 60 (2011) 1017–1030.

[8] C. Archetti, M. G. Speranza, A. Corberan, J. M. Sanchis, I. Plana, The team orienteering arc routing problem, Transportation Science 0 (0) (2013) 1–16.

[9] C. Archetti, A. Corberan, I. Plana, J. M. Sanchis, M. G. Speranza, A matheuristic for the team orienteering arc routing problem, Working paper, Department of Economics and Management, University of Brescia (2013).

[10] C. H. Papadimitriou, Computational complexity, Addison-Wesley, 1994.

[11] R. Deitch, S. P. Ladany, The one-period bus touring problem: Solved by an effective heuristic for the orienteering tour problem and improvement algorithm, EJOR 127 (1) (2000) 69 – 77.