Project Number 288094

eCOMPASS

eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

STREP

Funded by EC, INFSO-G4(ICT for Transport) under FP7

eCOMPASS – TR – 047

Approximation Algorithms for Arc Orienteering Problems

Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, Grammati Pantziou and Nikolaos Vathis

January 2014

Approximation Algorithms for Arc Orienteering Problems

Damianos Gavalas[1,6], Charalampos Konstantopoulos[2,6], Konstantinos Mastakas[3,6], Grammati Pantziou[4,6] and Nikolaos Vathis[5,6]

[1]Department of Cultural Technology and Communication, University of the Aegean, Mytilene, Greece, email: dgavalas@aegean.gr
[2]Department of Informatics, University of Piraeus, Piraeus, Greece, email: konstant@unipi.gr
[3]Department of Mathematics, University of Athens, Athens, Greece, email: kmast@math.uoa.gr
[4]Department of Informatics, Technological Educational Institution of Athens, Athens, Greece, email: pantziou@teiath.gr
[5]School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, email: nvathis@softlab.ntua.gr
[6]Computer Technology Institute & Press "Diophantus", Patras, Greece

Abstract

In this paper we study the Arc Orienteering Problem in directed and undirected graphs. To the best of our knowledge, we give the fist approximation algorithms for both the undirected and the directed versions of the problem. Our main results are the following: (i) We give an $O(\frac{\log^2(m)}{\log\log(m)})-$approximation algorithm for the AOP in directed graphs, where m is the number of arcs of the graph of the problem, using the $O(\frac{\log^2(n)}{\log\log(n)})-$approximation for the OP in directed graphs found in [13]. (ii) Using the $(2+\epsilon)-$approximation algorithm for the unweighted version of the OP in [6], we obtain 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 AOP in undirected graphs. Moreover, we prove that the Mixed Orienteering Problem (MOP) can be reduced to AOP and that any approximation algorithm for the AOP yields an approximation algorithm for the MOP.

Keywords: Arc Orienteering Problem, Orienteering Problem, Mixed Orienteering Problem, Approximation Algorithms, NP-completeness.

1 Introduction

The Arc Orienteering Problem (AOP) is a single route arc routing problem with profits introduced by Souffriau et al. in [15]. Given a directed graph $G = (V, A)$ whose arcs are associated with profits and travel times, two nodes $s, l \in V$, and a time budget B, the problem entails finding an $s - l$ walk of total length at most B so as to maximize the sum of the profits of the arcs visited by the walk. Note that the profit of each arc in the walk is collected only at the first time it is traversed. The AOP is the arc routing version of the Orienteering Problem (OP),

*This work was supported by the EU FP7/2007-2013 (DG CONNECT.H5-Smart Cities and Sustainability), under grant agreement no. 288094 (project eCOMPASS)

an NP-hard problem, named after a sport game called orienteering [9, 16]. In the OP the nodes (instead of the arcs) are associated with profits and the goal is to find a walk from s to l with length at most B such that the total profit of the visited nodes is maximized. The OP as well as its extension to multiple routes, the Team Orienteering Problem (TOP), and many other extensions and variants have been extensively studied in the literature. In the past decade a significant number of studies have been conducted wherein approximation approaches, metaheuristics and exact methods have been employed to tackle these problems (see [8] and [17]for a survey). One of the most common application of the OP and its extensions is to model different versions of the Tourist Trip Design Problem (TTDP) [18], a route-planning problem which deals with deriving near optimal routes for tourists visiting a destination with several points of interest (POIs) each associated with a profit.

The AOP is applicable to TTDP variants whose modeling requires profits to be associated with the arcs of the network as some links may be more beneficial to be traversed than others. As an example we may consider the derivation of personalized bicycle trips. Based on the biker's personal interests, starting and ending point and the available time budget, a personalized trip can be composed using arcs that better match the cyclist's profile. Similarly, AOP solvers may favor detours via riverside or pedestrian roads against shorter routes via high-traffic or unsafe zones for tourists moving among POIs. The extension of the AOP to multiple routes, introduced by Archetti et al. in [3] and named as Team Orienteering Arc Routing Problem (TOARP), may also find applications to TTDP variants. For example, consider the selection of paths of higher scenic value (among the many available between pairs of POIs) as well as the exclusion of paths including environmentally burdened road segments in favor of longer detours through pedestrian zones.

Although numerous research works concern the OP as well as many extensions and variants of the OP, there is very limited body of literature concerning AOP and TOARP. To the best of our knowledge, this literature includes the work of Souffriau et al. [15] which uses the AOP to model and provide a heuristic solution to the problem of planning cycle trips in the province of East Flanders, the work of Archetti et al. in [3] that proposes a formulation of the problem and a branch-and-cut algorithm and the work of Archetti et al. in [1] which introduces a matheuristic approach to TOARP.

The combination of the OP and the AOP is proposed in [17] under the name Mixed Orienteering Problem (MOP). In the MOP, profits are associated with the nodes as well as with the arcs of the graph. The problem is very interesting in the context of tourist trip planning as it can be used to formulate TTDP variants where certain routes may be of tourist interest, in addition to attractions. The only relevant research works concern the one-period Bus Touring Problem (BTP) [7], and the Outdoor Activity Tour Suggestion Problem (OATSP) [12].

In this paper we study the AOP in directed and undirected graphs. We prove that the AOP is NP-hard and, to the best of our knowledge, we propose the first approximation algorithms for both the undirected and the directed version of the problem. Specifically, our main contributions are the following:

- Using the $O(\frac{\log^2(n)}{\log\log(n)})-$approximation for the OP in directed graphs by Nagarajan and Navi [13], we obtain a $O(\frac{\log^2(m)}{\log\log(m)})-$approximation for the AOP, where m is the number of arcs of the graph of the problem.
- Using the $(2+\epsilon)-$approximation algorithm for the unweighted version of the OP by Chekuri et al. [6], we obtain
  - a $(6+\epsilon+o(1))-$approximation algorithm for the AOP in undirected graphs.
  - a $(4+\epsilon)-$approximation algorithm for the unweighted version of the AOP in undirected graphs.

Also, we prove that the MOP can be reduced to AOP, and that any approximation algorithm for the AOP yields an approximation algorithm for the MOP. The paper is organized as follows: In Section 2 we present related work. In Section 3 we prove that the AOP is NP-hard and give an approximation algorithm for the AOP in directed graphs. In Section 4 we present approximation algorithms for the AOP in undirected graphs and the unweighted version of the AOP in undirected graphs. Finally, in Section 5 we give approximation algorithms for the MOP.

2 Related work

Souffriau et al. in [15] use the AOP to model and solve the problem of planning cycle trips in the province of East Flanders. Their solution approach is based on a Greedy Randomized Adaptive Search Procedure (GRASP), while experimental results are based on instances generated from the East Flanders network.

Archetti et al. in [3] propose a formulation for the AOP and study a relaxation of its associated polyhedron. Also, they develop a branch-and-cut algorithm for solving the problem. Archetti et al. in [1] propose a matheuristic approach for the AOP. Experimental results show that the algorithm gives an average percentage error with respect to the optimal solution which is lower than 1%.

The Undirected Capacitated Arc Routing Problem with Profits (UCARPP), the arc routing counterpart of the capacitated TOP, is considered in [2]. In this problem a profit and a non-negative demand is associated with each arc and the objective is to determine a path for each available vehicle in order to maximize the total collected profit, without violating the capacity and time limit constraints of each vehicle. The authors consider an application where carriers can select potential customers for transporting their goods. Another potential application is the creation of personalized bicycle trips. An exact approach for solving the UCARPP along with several heuristics were proposed in [2]. The problem was also studied by Zachariadis and Kiranoudis in [19] where a local search procedure was given.

To the best of our knowledge, the only research works relevant to the MOP, concern the one-period Bus Touring Problem (BTP) [7], and the Outdoor Activity Tour Suggestion Problem (OATSP) [12]. In the BTP the objective is to maximize the total profit of the tour by selecting a subset of nodes to be visited and arcs to be traveled both having associated profits, given a constraint on the total touring time. The profit of recurrently visited nodes and arcs is only counted once. In [7] a heuristic approach is employed to solve the BTP. The OATSP, introduced recently by Maervoet et al. [12], involves finding attractive closed paths in a transportation network, tailored for a specific outdoor activity mode such as hiking and mountain biking. Total path attractiveness is evaluated as the sum of the average arc attractiveness and the profits of the nodes along the path. The problem involves finding a closed path of maximal attractiveness given a target path length and tolerance. That is, the OATSP requires a target path length instead of a maximal travel time required by the BTP. This gives rise to a path length window constraint. In [12] an efficient heuristic solution to the OATSP is presented.

To the best of our knowledge, no approximation algorithms for the AOP or the MOP have been presented in the literature. On the other hand, there is a significant number of research works on the approximability of the OP. As mentioned in the introduction, OP is NP-hard ([9], [11]) and it also known to be APX-hard [4]. The basic idea for approximating the OP was presented by Blum et al. in [4], [5] where the min-excess $s-t$ path problem (given two nodes s, t and an integer k, find an $s-t$ path of minimum-excess [1] that visits at least k nodes) was defined. It was shown that an approximation for the min-excess path problem implies an approximation for the OP. Then, the min-excess path problem can be approximated using algorithms for the k-stroll problem (find a minimum length $s-t$ walk that visits at least k nodes). Blum et

[1]The excess of an $s-t$ path is the difference of the path length from the length of the shortest $s-t$ path.

al. obtained a 4-approximation algorithm for the OP in undirected graphs by using a $(2 + \epsilon)$-approximation for the k-stroll path problem. In fact, most subsequent approximation algorithms for OP follow the framework of [4], [5] which reduces the OP to the k-stroll problem via the min-excess path problem. The best known approximation algorithm for the OP in undirected graphs is due to Chekuri et al. [6] who obtained a $(2 + \epsilon)$-approximation algorithm with running time $n^{O(1/\epsilon^2)}$ by giving a bi-criteria approximation for k-stroll problem with respect to the path length and the number of nodes visited. Using the same approach, they also obtained an $O(\log^2 OPT)$ approximation algorithm for the OP in directed graphs, where OPT denotes the number of nodes in an optimal solution. The best known approximation algorithm for the OP in directed graphs is due to Nagarajan and Ravi [13]. They gave an $O(\frac{\log^2 n}{\log \log n})$-approximation algorithm for the OP in directed graphs employing a bi-criteria approximation solution for k-stroll based on an LP approach.

3 The Arc Orienteering Problem

The AOP is defined as follows [15]: 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, $t : A \to \mathbb{R}^+$ i.e., each arc $a \in A$ is associated with a travel time t_a, $p : A \to \mathbb{R}^+$ i.e., each arc is associated with a profit p_a, and a non-negative time budget B, the goal is to find an $s - l$ walk with length at most B so as to maximize the sum of the profits of the arcs traversed by the walk. Note that an arc may be traversed multiple times by the walk. 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, independently of the number of times it is traversed.

We first prove that the AOP is NP-hard and then we give an approximation algorithm for the problem by reducing it to the OP for directed graphs.

Theorem 1. *The AOP is NP-hard*

Proof. We shall reduce the NP-complete decision knapsack problem [14] to the AOP. In the former we are given 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 we can pick, and a target profit P, and the question is whether there is a subset of the objects with total weight at most W and total profit at least P. We reduce the Knapsack problem to the AOP as follows: Given an instance of the Knapsack, we consider a directed star graph G with a central (start and terminal) node s connected to each node o_i representing an object, and vice versa. Also we consider each arc having a travel time and a profit equal to half of the o_i's weight and profit, respectively, the time budget equal to W and the total profit that should be collected by a walk greater than or equal to P. Formally, the graph G is defined as follows: $G = (V, A)$ where $V = \{s\} \cup O, A = \{(s, o_i), (o_i, s) : i = 1, 2, \ldots, n\}$ and for each $o_i \in V$ the arcs (s, o_i) and (o_i, s) have travel time equal to $\frac{w_i}{2}$ and profit equal to $\frac{p_i}{2}$. It is easy to notice that a subset $S = \{o_{k_1}, o_{k_2}, \ldots, o_{k_l}\}$ of O has total weight at most W and total profit greater than or equal to P if and only if the closed walk $(s, o_{k_1}, s, o_{k_2}, s, \ldots, s, o_{k_l}, s)$ is of length at most W and has total profit at least P. $\square$

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

Proof. Given an instance of the AOP $(N = (V, A), t, p, B)$, $|A| = m$ we construct an instance of the OP in the directed asymmetric metric network N_m in two phases as follows: we first define $N' = (V', A')$ such that $V' = \{s, l\} \cup \{(u, v) : (u, v) \in A\}$ $(|V'| = n' = m + 2)$ i.e., the set of nodes of N' consist of the starting node s, the terminal node l and a node a for each arc a in A

representing that traversing the arc a in AOP corresponds to visiting the node a in the OP. The set of arcs A' is defined as follows: s is connected only to the nodes representing the outgoing arcs from s, i.e. of the form (s,u) with the outgoing arc $(s,(s,u))$ and l is connected only to the nodes representing the ingoing arcs to l, i.e. (u,l) with the ingoing arc $((u,l),l)$ and all the other connections will be of the form $((u,v),(v,w))$ where (u,v) and (v,w) are arcs of N. Thus, $A' = \{(s,(s,u)),((u,l),l) : (s,u) \text{ and } (u,l) \in A\} \cup \{((u,v),(v,w)) : (u,v),(v,w) \in A\}$. The travel time of an arc $((u,v),(v,w))$ equals to half of the sum of the travel times of (u,v) and (v,w) in the AOP instance, hence $t'_{((u,v),(v,w))} = \frac{t_{(u,v)}+t_{(v,w)}}{2}$ and $t'_{(s,(s,u))} = \frac{t_{(s,u)}}{2}$ and $t'_{((u,l),l)} = \frac{t_{(u,l)}}{2}$. The profits of the nodes of the OP instance represent the profits of the arcs of the AOP instance, so $p'_{(u,v)} = p_{(u,v)}$ and the allowed time budget is the same in both instances, hence $B' = B$. Then, we obtain the network N_m, the metric closure of the network N'. Note, that each arc in N' will retain the same travel time in network N_m, i.e. for the arc $((u,v),(v,w)) \in N_m$, $t'_{((u,v),(v,w))} = \frac{t_{(u,v)}+t_{(v,w)}}{2}$, since any other path connecting (u,v) with (v,w) will be of the form $((u,v),(v,x_1),(x_1,x_2),\ldots,(x_k,v),(v,w))$ and hence have travel time greater than or equal to $t'_{((u,v),(v,x_1))} + t'_{((x_k,v),(v,w))} = \frac{t_{(u,v)}+t_{(v,x_1)}+t_{(x_k,v)}+t_{(v,w)}}{2} \geq \frac{t_{(u,v)}+t_{(v,w)}}{2}$.

Now, we will prove that a solution to the AOP instance yields a solution to the OP instance with the same total profit and length and hence

$$\mathrm{OPT}_{\mathrm{AOP}} \leq \mathrm{OPT}_{\mathrm{OP}} \tag{1}$$

For this, consider that a solution to the AOP $P = (s = w_0, w_1, w_2, \ldots, w_{k-1}, w_k = l)$, is transformed into a solution to the OP as follows: $P' = (s,(s,w_1),(w_1,w_2),\ldots,(w_{k-1},l),l)$ with total length

$$t'_{(s,(s,w_1))} + \sum_{i=0}^{k-2} t'_{((w_i,w_{i+1}),(w_{i+1},w_{i+2}))} + t'_{((w_{k-1},l),l)} = \frac{t_{(s,w_1)}}{2} + \sum_{i=0}^{k-2} \frac{t_{(w_i,w_{i+1})} + t_{(w_{i+1},w_{i+2})}}{2} + \frac{t_{(w_{k-1},l)}}{2}$$

which is equal to the length of P and collects the same profit with P.

On the other hand, given a feasible solution W_m of the OP instance in N_m (i.e. a $s-l$ walk of total travel time at most B) we first decompose each edge in the walk into its corresponding shortest path in N' obtaining the walk W'. Namely, if the node (u,v) is followed by the node (w,x) in the walk and its shortest path in N' is the sequence $((u,v),(v,z_1),(z_1,z_2),\ldots,(z_k,w),(w,x))$, then since N_m is the metric closure of N', the travel time of the edge $((u,v),(w,x))$ will be equal to the travel time of the segment $((u,v),(v,z_1),(z_1,z_2),\ldots,(z_k,w),(w,x))$. Also, the profit collected by visiting the nodes of the segment will be greater than or equal to the profit obtained by just visiting (w,x) after (u,v) in N_m. Hence, we obtain a walk W' of the same total travel time and profit greater than or equal to the profit W_m. Then, W' will be of the form $W' = (s,(s,w_1),(w_1,w_2),\ldots,(w_{k-1},l),l)$. The latter is re-transformed into the walk $W = (s,w_1,w_2,w_3,\ldots,w_{k-1},l)$ in the AOP instance with the same travel time and profit as with W' in the OP instance. Under the previous result, getting a solution W_m of the N_m using an $f(n')-$approximation algorithm for the OP in directed asymmetric metric graphs, we obtain that $\mathrm{profit}(W_m) \geq \frac{\mathrm{OPT}_{\mathrm{OP}}}{f(n')}$. Then, retransforming W_m, first into W', a walk in N', and then into W, a walk in the AOP instance, we get a solution with profit

$$\mathrm{profit}(W) \geq \mathrm{profit}(W_m) \geq \frac{\mathrm{OPT}_{\mathrm{OP}}}{f(n')}$$

Using (1), we get that

$$\mathrm{profit}(W) \geq \frac{\mathrm{OPT}_{\mathrm{AOP}}}{f(n')} = \frac{\mathrm{OPT}_{\mathrm{AOP}}}{f(m+2)}$$

□

Corollary 3. *Using the $O(\frac{\log^2(n)}{\log\log(n)})-$approximation for the OP in directed asymmetric metric networks by Nagarajan and Navi [13], we obtain a $O(\frac{\log^2(m)}{\log\log(m)})-$approximation for the AOP, where m is the number of arcs of the graph of the problem.*

4 Approximation Algorithms for the AOP in Undirected Graphs

In this section we study the AOP in undirected graphs i.e. the graph in the definition of the problem $(G=(V,E),t,p,B)$ consists of edges instead of arcs. Notice that a similar reduction to the one for the AOP given in Theorem 1 (i.e., define the set of edges of the graph G of the AOP instance as the set $\{\{s,o_1\},\{s,o_2\},\cdots,\{s,o_n\}\}$ and assign profit p_i to each edge $\{s,o_i\}$ equal to the profit of the object o_i, $i=1,2,\cdots,n$), shows that the AOP in undirected graphs is NP-hard. Therefore, we have the following Lemma.

Lemma 4. *The AOP in undirected graphs is NP-hard.*

In the sequel, we prove that there exists a constant factor approximation algorithm for the AOP in undirected graphs by reducing it to the Unweighted OP (UOP) in undirected graphs. The UOP is the restriction of the OP where all the nodes of the graph G have profit equal to 1. The problem can be also stated as follows: Given a graph G with travel times associated with its edges, two nodes s and l, and a time budget B, the goal is to find an $s-l$ walk of total length at most B so as to maximize the number of distinct nodes visited by the walk. First, using scaling and rounding techniques in a similar way with the work of [10, 6] for the OP, the AOP with general profits over the edges can be reduced to the AOP with profits over the edges integers polynomially bounded by the size of the instance.

Lemma 5. *A $\rho-$approximation algorithm for the AOP in undirected graphs with profits over the edges integers polynomially bounded by the size of instance i.e. there is an integer k such that for each edge $e=\{u,v\}$, the profit p_e is $1\le p_e\le n^k$, yields a $(\rho+o(1))-$approximation algorithm for the AOP in undirected graphs.*

Proof. Given an instance $(N=(V,E),t,p,B)$ of the AOP with general profits over its edges, we shall construct an instance $(N'=(V',E'),t,p',B)$ with polynomially bounded integer profits over its edges. First, we guess the edge of highest profit in the optimal walk of the initial instance. Let this profit be equal to $p_{\max}$. Then, we remove from E all the edges with profit greater than $p_{\max}$. Notice that if $N'=(V',E')$ is the new graph, the optimal solution in N' is the same with the optimal solution in N. Then, we apply a scaling technique for each edge $e\in A'$ by setting its new profit as

$$p'_e=\lfloor\frac{n^3p_e}{p_{\max}}\rfloor+1$$

Consider a feasible walk W, consisting of the distinct edges $e_1,e_2,\ldots,e_k$. Then, in the initial instance (in N) W will have profit equal to $\text{profit}(W)=\sum_{j=1}^{k}p_{e_j}$ while in the latter instance (in N'), W will have profit $\text{profit}'(W)=\sum_{j=1}^{k}p'_{e_j}>\sum_{j=1}^{k}\frac{n^3p_{e_j}}{p_{\max}}=\frac{n^3\text{profit(W)}}{p_{\max}}$. Hence, if OPT is the value of an optimal solution in N and OPT' its value in N', then

$$\text{OPT}'>\frac{n^3}{p_{\max}}\text{OPT} \tag{2}$$

On the other hand, $\text{profit}'(W) = \sum_{j=1}^{k} p'_{e_j} \leq \sum_{j=1}^{k}(\frac{n^3 p_{e_j}}{p_{\max}}+1) \implies \frac{n^3}{p_{\max}}\text{profit}(W) \geq \text{profit}'(W) - k \geq \text{profit}'(W) - m \geq \text{profit}'(W) - m\frac{\text{OPT}}{p_{\max}}$, so

$$\text{profit}(W) \geq \frac{p_{\max}}{n^3}\text{profit}'(W) - \frac{m}{n^3}\text{OPT} \tag{3}$$

Using a $\rho-$approximation algorithm for the latter instance we obtain a walk W with profit $\text{profit}'(W) \geq \frac{\text{OPT}'}{\rho}$ and from (3), the profit of walk W in the initial instance will be

$$\text{profit}(W) \geq \frac{1}{\rho}\frac{p_{\max}}{n^3}\text{OPT}' - \frac{m}{n^3}\text{OPT}$$

Finally, using (2) we get that

$$\text{profit}(W) > (\frac{1}{\rho} - \frac{m}{n^3})\text{OPT}$$

□

Lemma 6. *A $\rho-$approximation algorithm for the UOP in undirected graphs yields a $3\rho-$approximation algorithm for the AOP in undirected graphs when the profits of the edges are integers polynomially bounded by the size of instance, i.e. there is an integer k such that for each edge $e = \{u, v\}$, the profit p_e is $1 \leq p_e \leq n^k$.*

Proof. Given an instance of the AOP, first we consider that for each edge $e = \{u, v\}$ of the AOP instance the shortest path from s to l passing through e has length at most B (this can be tested in polynomial time by checking whether $\min\{l(s, u)+t_{\{u,v\}}+l(v, t), l(s, v)+t_{\{u,v\}}+l(u, t)\} \leq B$, where $l(x, y)$ is the length of the shortest path from node x to node y), otherwise e cannot be part of the solution and it is removed from the graph. Then we construct an instance of UOP by splitting each edge $\{u, v\}$ of the AOP into $p_{uv} + 1$ edges (subsegments) as follows: For each edge $\{u, v\}$ of the AOP instance we create the nodes $u, \{u, v\}_1, \{u, v\}_2, \ldots, \{u, v\}_{p_{uv}}, v$ and the edges $\{u, \{u, v\}_1\}$, $\{\{u, v\}_1, \{u, v\}_2\}$, $\cdots$, $\{\{u, v\}_{p_{uv}-1}, \{u, v\}_{p_{uv}}\}$, $\{\{u, v\}_{p_{uv}}, v\}$ (see Figure 1). We set the travel times of the edges $\{u, \{u, v\}_1\}$ and $\{\{u, v\}_{p_{uv}}, v\}$ equal to half of the travel time of the initial edge $\{u, v\}$ i.e. equal to $\frac{t_{\{u,v\}}}{2}$. We also set the travel times of the edges $\{\{u, v\}_i, \{u, v\}_{i+1}\}, i = 1, 2, \ldots, p_{uv-1}$ equal to zero. Now, a solution W_{AOP} to the AOP instance is easily transformed into a solution W_{UOP} to the UOP instance by just replacing any edge $\{u, v\}$ of the W_{AOP} with the sequence of nodes $u, \{u, v\}_1, \{u, v\}_2, \ldots, \{u, v\}_{p_{uv}}, v$. It is easy to verify that the number of the distinct nodes in W_{UOP} is greater than or equal to the total profit of the W_{AOP}, and the total travel time of W_{UOP} is equal to the travel time of the W_{AOP} and hence $\text{OPT}_{\text{AOP}} \leq \text{OPT}_{\text{UOP}}$.

Now, we shall prove that any solution of the UOP instance can be re-transformed into a solution to the AOP with at least one third of the profit of the former solution. We shall consider a sequence of nodes of the form $u, \{u, v\}_1, \{u, v\}_2, \ldots, \{u, v\}_{p_{uv}}, v$ as an *appropriate* segment, i.e. a segment that represents an edge of the original instance. Likewise, a sequence of the form $u, \{u, v\}_1, \{u, v\}_2, \ldots, \{u, v\}_{i-1}, \{u, v\}_i, \{u, v\}_{i-1} \ldots, \{u, v\}_2, \{u, v\}_1, u$ is considered as an *inappropriate* segment i.e. a segment that does not represent an edge of the AOP instance. For example in Figure 1, the segment $v_1, \{v_1, v_2\}_1, \{v_1, v_2\}_2, v_2$ is an appropriate segment, while the segment $v_5, \{v_5, v_6\}_1, \{v_5, v_6\}_2, \{v_5, v_6\}_3, \{v_5, v_6\}_2, \{v_5, v_6\}_1, v_5$ is an inappropriate one. Then, a solution to the UOP instance consists of a sequence of segments, where each segment is either appropriate or inappropriate. Notice that for each inappropriate segment $u, \{u, v\}_1, \{u, v\}_2, \ldots, \{u, v\}_{i-1}, \{u, v\}_i, \{u, v\}_{i-1} \ldots, \{u, v\}_2, \{u, v\}_1, u$ we may consider that $i = p_{uv}$, because otherwise, it can be replaced by the equal length and higher profit segment with $i = p_{uv}$.

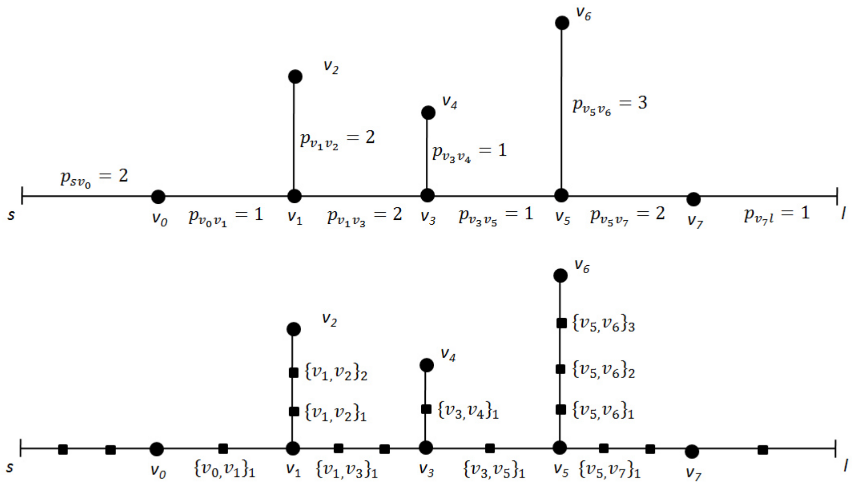

Figure 1: Illustration of representing the profit of an edge with nodes

In a UOP solution, if p_{AS} is the profit gained by the appropriate segments and p_{IS} is the profit gained by the inappropriate segments then the total profit of the solution p_{TOT} is equal to $p_{\text{AS}} + p_{\text{IS}}$. In the case that all segments participating in a UOP solution W_{UOP} are appropriate, the re-transformation of the W_{UOP} to an AOP solution W_{AOP} is obviously done by adding to the W_{AOP} the edges of the AOP instance represented by the segments of the UOP instance. This yields a feasible AOP solution with profit equal to the number of distinct nodes of the form $\{u, v\}_i$ in the walk W_{UOP}. Notice that the profit of W_{AOP} will be at least half of the W_{UOP}'s profit, since each new appropriate segment in the UOP instance will contribute to the solution's profit at most the profit of the represented edge plus 1, counting the endpoint of the edge, hence at most twice the profit of the edge.

If however, the set of inappropriate segments is not the empty set, then the re-transformation of an UOP solution to a AOP one is not that easy. Given a UOP solution W_{UOP}, we say that we *extend* an inappropriate segment $u, \{u, v\}_1, \ldots, \{u, v\}_{p_{uv}-1}, \{u, v\}_{p_{uv}}, \{u, v\}_{p_{uv}-1}, \ldots, \{u, v\}_1, u$ in W_{UOP} when replacing it with the appropriate segments $u, \{u, v\}_1, \ldots, \{u, v\}_{p_{uv}}, v$ and $v, \{u, v\}_{p_{uv}}, \ldots, \{u, v\}_1, u$. Then the idea is to construct an AOP solution as follows: first add all the edges corresponding to the appropriate segments and then add a number of edges corresponding to inappropriate segments (i.e. extend these segments to make them appropriate) and ignore a number of inappropriate segments (in order not to violate the time budget). This can be done as long as the total profit of the constructed AOP solution is at least a constant factor of the W_{UOP} profit.

Let IS= $\{s_1, s_2, \ldots, s_k\}$ be the set of inappropriate segments participating in W_{UOP}. Let also $t_1, t_2, \ldots, t_k$ be the travel times spent on these segments and $p_1, p_2, \ldots, p_k$ be the profits collected by traversing them ($\sum_{i=1}^{k} p_i = p_{\text{IS}}$). A subset of IS, FS= $\{s_{a_1}, s_{a_2}, \ldots, s_{a_m}\}, m \leq k$, is a *feasible subset* of segments in the case that $\sum_{j=1}^{m} 2t_{a_j} \leq \sum_{i=1}^{k} t_i$. Then FS is a *maximal subset* for the given time constraint, if the insertion of another segment (a_{m+1}) would violate the time budget i.e. $\sum_{j=1}^{m+1} 2t_{a_j} > \sum_{i=1}^{k} t_i$. Now, we consider a maximal feasible subset MFS of segments (this can be found in polynomial time) and distinguish between the following cases:

- If MFS has total profit greater than or equal to one third of the total profit of the IS

($p_{\text{MFS}} \geq \frac{p_{\text{IS}}}{3}$), then we append the segments of MFS (called the appended segments) and remove all the other segments of IS. This creates a walk of only appropriate segments with total profit at least $p_{\text{MFS}} + \frac{p_{\text{AS}}}{2} \geq \frac{p_{\text{IS}}+p_{\text{AS}}}{3} = \frac{p_{\text{TOT}}}{3}$, a third of the profit of the initial solution, since from the appended segments we add their represented edges of AOP getting a third of their total profit and adding the edges represented from the previous appropriate segments (not the appended ones) we get at least half of the profit of each segment.

- If MFS has total profit less than a third of the total profit of the IS, then the relative complement of MFS in IS, IS\MFS = MFS^c, has profit at least two thirds of the total profit of the IS. Then we distinguish between the following two cases:
  - if MFS^c has at least two elements then we remove the segment with the lowest profit call it s_b and this creates a feasible subset since MFS was maximal for the time constraint and MFS$^c\backslash\{s_b\}$ has profit at least half the profit of MFS^c, hence at least a third of IS and with the same procedure as previous we get a solution to the AOP with at least a third of the solution of the UOP by extending the segments in MFS^c.
  - if MFS^c has exactly one segment consider it to be s_1, then if $p_1 \geq \frac{p_{\text{TOT}}}{3}$ considering the shortest path from s to l passing through the edge representing the extended segment s_1, we obtain a walk with total profit of at least $\frac{p_{\text{TOT}}}{3}$. Otherwise, if $p_1 < \frac{p_{\text{TOT}}}{3}$ (i.e. $p_{\text{AS}} + p_{\text{MFS}} \geq \frac{2p_{\text{TOT}}}{3}$) it is enough to extend the segments in MFS and remove s_1, obtaining in this way, a walk with profit at least $\frac{p_{\text{AS}}}{2} + p_{\text{MFS}} \geq \frac{p_{\text{AS}}+p_{\text{MFS}}}{2} \geq \frac{p_{\text{TOT}}}{3}$.

So, any solution of the UOP instance is re-transformed into a solution of the AOP instance with total profit at least a third of the former. Hence, obtaining a $\rho-$approximation solution to the UOP it is re-transformed into a solution to AOP with total profit at least $\frac{\text{OPT}_{\text{UOP}}}{3\rho}$ and since as proven $\text{OPT}_{\text{AOP}} \leq \text{OPT}_{\text{UOP}}$, this yields a $3\rho-$approximation algorithm for the AOP. ☐

Theorem 7. *There exists a $(6 + \epsilon + o(1))-$approximation algorithm for the AOP in undirected graphs with execution time $n^{O(\frac{1}{\epsilon^2})}$.*

Proof. We first apply Lemma 5 to creat a graph with polynomially bounded profits over the edges. Then, the Lemma 6 combined with the $(2 + \epsilon)-$approximation algorithm for the UOP by Chekuri et al. [6] produces the required solution. ☐

The unweighted version of AOP (**UAOP**) in undirected graphs is the restriction of the problem where all edges have profit equal to 1. Thus, it can be alternatively defined as follows: Given a graph G with travel times associated with its edges, two nodes s and l, and a time budget B, the goal is to find an $s - l$ walk of total length at most B so as to maximize the number of distinct edges traversed by the walk.

Lemma 8. *A $\rho-$approximation algorithm for the UOP in undirected graphs yields a solution with profit at least $\lfloor \frac{OPT}{2\rho} \rfloor$ to the UAOP in undirected graphs.*

Proof. The proof is similar to the Lemma 6 with the only difference that instead of choosing a maximal feasible set of segments with one third of the total profit, we pick the $\lfloor \frac{k}{2} \rfloor$ segments with the least travel time, hence obtaining a solution with profit at least $\lfloor \frac{\text{profit of solution of UOP}}{2} \rfloor$. ☐

Similarly to Theorem 7 we obtain the following theorem.

Theorem 9. *There exists a $(4+\epsilon)-$approximation algorithm for the UAOP in undirected graphs running in $n^{O(\frac{1}{\epsilon^2})}$ time.*

5 The Mixed Orienteering Problem

The Mixed Orienteering Problem **MOP** first mentioned in [17], also seen as bus touring problem [7], is the combination of the OP and the AOP. In the MOP, profits are associated to the nodes as well as to the arcs of the graph. The problem can be formally defined as follows: Given an instance $(N = (V, A), t, p, B)$ where $N = (V, A)$ 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, $t : A \to \mathbb{R}^+$ i.e., each arc $a \in A$ is associated with a travel time t_a, $p : V \cup A \to \mathbb{R}^+$ i.e., each node and arc is associated with a profit, and a non-negative time budget B, the goal is to find an $s - l$ walk with length at most B so as to maximize the sum of the profits of the nodes visited and the arcs traversed by the walk. The profit from a node or an arc is collected only once, independently of the number of times a node is visited or an arc is traversed.

Theorem 10. *The Mixed Orienteering Problem can be reduced to AOP.*

Proof. Given an instance of the Mixed Orienteering Problem $(N = (V, A), t, p, B)$ we construct an instance of the AOP $(N' = (V', A'), t', p', B')$ as follows: for each node $u \in V$ we create a node $u' \in V'$ connected only with u, such than both arcs (u, u') and (u', u) have zero length and profit equal to half of the profit of node u in the instance of Mixed Orienteering problem. Formally, $V' = \{u, u' : u \in V\}$ and $A' = A \cup \{(u, u'), (u', u) : u \in V\}$, $t'_{(u,v)} = t_{(u,v)}, (u, v) \in A$ and $t'_{(u,u')} = t'_{(u',u)} = 0$ and $p'_{(u,v)} = p_{(u,v)}, (u, v) \in A$ and $p'_{(u,u')} = p'_{(u',u)} = \frac{p_u}{2}$ and $B' = B$.

Every walk $W = (s, w_1, w_2, \ldots, w_k, t)$ in the instance of the Mixed Orienteering Problem can be transformed into a walk $W' = (s, w_1, w'_1, w_1, w_2, w'_2, w_2, \ldots, w_k, w'_k, w_k, t)$ with the same length and profit. So the optimal value of the MOP OPT_{MOP} is less than or equal to the optimal value of the AOP OPT_{AOP}. Similarly, any walk of the AOP instance is re-transformed in a walk of the MOP instance by removing all the $u' \in V'$ from the path and has the same length with the former and at least the same profit. So, $\text{OPT}_{MOP} = \text{OPT}_{AOP}$. $\square$

Note that from the previous proof the re-transformation of the AOP to the MOP yields at least a same profit solution. Therefore, the following corollary holds.

Corollary 11. *Any approximation algorithm for the AOP yields an approximation algorithm for the MOP and hence based on Corollary 3 we have obtained an* $O(\frac{\log^2 n}{\log\log n})-$*approximation algorithm for the MOP.*

The MOP in undirected graphs is the MOP defined over an undirected graph. In a similar reduction to the one given in Theorem 10, with the only difference of inserting for each node $u \in V$ an edge $\{u, u'\}$ with profit equal to the profit of u instead of two arcs (u, u') and (u', u) with half of the profit, we obtain the following Lemma

Lemma 12. *MOP in undirected graphs can be reduced to AOP in undirected graphs and any approximation algorithm for the AOP in undirected graphs yields an approximation algorithm for the MOP in undirected graphs with the same approximation ratio.*

Hence, using Theorem 7 we obtain the following Corolary

Corollary 13. *There exists a* $(6+\epsilon+o(1))-$*approximation algorithm for the MOP in undirected graphs running in* $n^{O(\frac{1}{\epsilon^2})}$.

References

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

[2] C. Archetti, D. Feillet, A. Hertz, and M. G. Speranza. The undirected capacitated arc routing problem with profits. *Computers & Operations Research*, 37(11):1860 – 1869, 2010.

[3] C. Archetti, M. G. Speranza, A. Corberan, J. M. Sanchis, and I. Plana. The team orienteering arc routing problem. *Transportation Science*, 0(0):null, 0.

[4] A. Blum, S. Chawla, D. R. 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 the Foundations of Computer Science*, pages 46 –55, 2003.

[5] A. Blum, S. Chawla, D. R. Karger, Lane T., A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. *SIAM J. Comput.*, 37(2):653–670, 2007.

[6] C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. In *Proceedings of the 19th Annual ACM-SIAM symposium on Discrete Algorithms*, SODA '08, pages 661–670, 2008.

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

[8] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, and Y. Tasoulas. A survey on algorithmic approaches for solving tourist trip design problems. http://www.ecompass-project.eu/sites/default/files/ECOMPASS-TR-010.pdf, October 2012. Technical Report of European Commission FP7 Project 'eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers (eCOMPASS)'.

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

[10] N. J. Korula. *Approximation algorithms for network design and orienteering*. PhD thesis, University of Illinois at Urbana-Champaign, 2010.

[11] G. Laporte and S. Martello. The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2-3):193 – 207, 1990.

[12] J. Maervoet, P. Brackman, K. Verbeeck, P. De Causmaecker, and G. Vanden Berghe. Tour suggestion for outdoor activities. In *Proceedings of the 12th International Symposium on Web and Wireless Geographical Information Systems (W2GIS'13)*, volume 7820 of *LNCS*, pages 54–63, 2013.

[13] V. Nagarajan and R. Ravi. The directed orienteering problem. *Algorithmica*, 60:1017–1030, August 2011.

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

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

[16] T. Tsiligirides. Heuristic methods applied to orienteering. *The Journal of the Operational Research Society*, 35(9):797–809, 1984.

[17] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1 – 10, 2011.

[18] P. Vansteenwegen and D. Van Oudheusden. The mobile tourist guide: An or opportunity. *Operational Research Insight*, 20(3):21–27, 2007.

[19] E.E. Zachariadis and C.T. Kiranoudis. Local search for the undirected capacitated arc routing problem with profits. *European Journal of Operational Research*, 210(2):358 – 367, 2011.