Constant-Factor Approximation to Deadline TSP and Related Problems in (Almost) Quasi-Polytime

Zachary Friggstad
Department of Computer Science, University of Alberta, Edmonton, Canada

Chaitanya Swamy
Department of Combinatorics and Optimization, University of Waterloo, Canada

Abstract

We investigate a genre of vehicle-routing problems (VRPs), that we call *max-reward* VRP*s*, wherein nodes located in a metric space have associated rewards that depend on their visiting times, and we seek a path that earns maximum reward. A prominent problem in this genre is *deadline TSP*, where nodes have *deadlines* and we seek a path that visits all nodes by their deadlines and earns maximum reward. Our main result is a *constant-factor approximation* for deadline TSP running in time $O\left(n^{O(\log(n\Delta))}\right)$ in metric spaces with integer distances at most Δ. This is the *first* improvement over the approximation factor of $O(\log n)$ due to Bansal et al. [2] in over 15 years (but is achieved in super-polynomial time). Our result provides the first concrete indication that $\log n$ is unlikely to be a real inapproximability barrier for deadline TSP, and raises the exciting possibility that deadline TSP might admit a polytime constant-factor approximation.

At a high level, we obtain our result by carefully guessing an appropriate sequence of $O(\log(n\Delta))$ nodes appearing on the optimal path, and finding suitable paths between any two consecutive guessed nodes. We argue that the problem of finding a path between two consecutive guessed nodes can be relaxed to an instance of a special case of deadline TSP called *point-to-point (*P2P*) orienteering*. *Any* approximation algorithm for P2P orienteering can then be utilized in conjunction with either a greedy approach, or an LP-rounding approach, to find a good set of paths overall between every pair of guessed nodes. While concatenating these paths does not immediately yield a feasible solution, we argue that it can be covered by a constant number of feasible solutions. Overall our result therefore provides a *novel reduction* showing that any α-approximation for P2P orienteering can be leveraged to obtain an $O(\alpha)$-approximation for deadline TSP in $O\left(n^{O(\log n\Delta)}\right)$ time.

Our results extend to yield the same guarantees (in approximation ratio and running time) for a substantial generalization of deadline TSP, where the reward obtained by a client is given by an arbitrary non-increasing function (specified by a value oracle) of its visiting time. Finally, we discuss applications of our results to variants of deadline TSP, including settings where both end-nodes are specified, nodes have release dates, and orienteering with time windows.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Mathematics of computing → Discrete optimization

Keywords and phrases Approximation algorithms, Vehicle routing problems, Deadline TSP, Orienteering

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.67

Category Track A: Algorithms, Complexity and Games

Funding *Zachary Friggstad*: Supported by the Canada Research Chairs program and an NSERC Discovery grant.
Chaitanya Swamy: Supported in part by NSERC grant 327620-09 and an NSERC Discovery Accelerator Supplement Award.

1 Introduction

Vehicle-routing problems (VRPs) constitute a rich class of optimization problems that find a variety of applications and have been extensively studied in the Operations Research and Computer Science literature (see, e.g., [27]) Broadly speaking, vehicle-routing problems can be divided into two categories: one, where we have a *fixed* set of nodes or clients that need to be visited, and we seek the most effective route(s) for visiting these clients (e.g., TSP-style problems [9, 25, 29, 26], minimum-latency problems [5, 23], VRPs with distance bounds [19, 22] and regret bounds [13]); and the other, where, due to resource constraints, we need to *select* which set of clients to visit *and* plan suitable routes for these clients. We investigate a prominent class of VRPs that fall into the second category, wherein nodes have associated rewards that depend on their visiting times, and we seek a path that earns maximum reward. We call this genre of problems *max-reward* VRP*s*, and they constitute a well-studied class of VRPs (see, e.g., [15, 3, 2, 8, 6, 14]).

We consider a fundamental problem in this genre, called the *deadline TSP* problem. In deadline TSP, we are given a (symmetric) metric space $(\{r\} \cup V, c)$ with r being a distinguished starting root node, and each node $v \in V$ has a certain deadline $D_v \geq 0$ and reward $\pi_v \geq 0$. We seek a path starting at r that maximizes the total reward of the nodes on the path that are visited by their deadlines. Since we are in a metric space, by shortcutting, we may assume that all nodes on the path are visited by their deadlines. So equivalently, we seek a maximum-reward r-rooted path that visits all nodes by their deadlines. A simpler problem in the max-reward VRP genre is the *point-to-point (*P2P*) orienteering* problem, wherein we are also given an end-node t, and we seek an r-t path of at most a given length B that collects maximum reward. This is a special case of deadline TSP, which can be seen by setting the deadline of each node v to $B - c_{vt}$.

Max-reward VRPs tend to be more complicated problems than the first category of "fixed-node-set" VRPs mentioned above because of the added combinatorial aspect of selecting which nodes to visit, which is interlinked with the routing decisions, and we have much less of an understanding of max-reward VRPs compared to fixed-node-set VRPs. A constant-factor approximation is known only for P2P orienteering in undirected graphs, which is one of the most rudimentary max-reward VRPs. For other, more-sophisticated, max-reward VRPs – deadline TSP, submodular orienteering, directed orienteering – only logarithmic or polylogarithmic (or worse) approximation factors are known. Furthermore, even for undirected orienteering, the current-best approximation ratio has remained stagnant at $(2 + \epsilon)$ [6] (and fresh LP-based insights were obtained only recently [14]), whereas for s-t path TSP (the corresponding fixed-node-set problem), a steady stream of work [1, 24, 28, 29] has exploited LP-based insights (and other ideas) to improve the approximation ratio to 1.5 [29]. The contrast is even more evident in asymmetric metrics: while $O(1)$-approximation algorithms are now known for asymmetric TSP (ATSP) [26] and s-t path ATSP [17], the best-known guarantee for directed orienteering is an $O(\alpha \log |V|)$-approximation using an LP-relative α-approximation for ATSP [21], which explicitly shows the degradation when moving to the max-reward-VRP version.

Our results. Our main contribution is to provide the *first constant-factor approximation* guarantee for deadline TSP (Theorem 3.1); notably, we obtain a relatively small approximation ratio of $(7.63 + \epsilon)$. We may assume by scaling that all distances in our metric space are integers. Our algorithm runs in time $O\big(n^{O(\log n\Delta)}\big)$, where n is the number of points, and Δ is the diameter of the (scaled) metric space. In particular, for graphical metrics, we obtain a quasi-polytime (i.e., $O\big(n^{O(\log n)}\big)$-time) constant-factor approximation. Our guarantee yields the *first* improvement over the (polytime) $O(\log n)$ approximation factor obtained by Bansal et al. [2] in over 15 years, even for graphical metrics, but is achieved in super-polynomial time. Prior to our work, it was unclear whether $\log n$ is a real inapproximability barrier for deadline TSP. Our result provides the first concrete indication that this is not the case,[1] and raises the enticing possibility that deadline TSP might admit a polynomial-time constant-factor approximation.

As noted above, constant-factor approximation algorithms are known for P2P orienteering [2, 6, 14], which is a special case of deadline TSP. Our chief technical contribution lies in providing a novel *reduction* showing that *an α-approximation algorithm for* P2P *orienteering can be utilized to obtain an $O(\alpha)$-approximation algorithm for deadline TSP* in $O\big(n^{O(\log n\Delta)}\big)$ time (see Theorem 3.1).

We obtain the *same approximation guarantees* for a substantial generalization of deadline TSP, wherein each node v has a *non-increasing reward function* $\pi_v : \mathbb{R}_+ \mapsto \mathbb{R}_+$, with $\pi_v(x)$ giving the reward of node v if v is visited at time x, and, as before, the goal is to find a path that collects maximum reward. We call this problem *monotone-reward TSP* (Section 4.1). Notice that this problem also captures *discounted-reward TSP*, considered by Blum et al. [3], which is the special case where $\pi_v(x) = \pi_v \cdot \gamma^{-x}$, where $\gamma > 1$ is a discount factor. Our results here only require *value-oracle* access to π_v – an oracle that on input x returns $\pi_v(x)$ – and follow from two distinct approaches (see Theorem 4.1): (a) we show that we can reduce monotone-reward TSP in polytime to deadline TSP with a $(1-\epsilon)$-factor loss; and (b) our algorithm and analysis for deadline TSP readily extend to yield the same guarantees for monotone-reward TSP.

Interestingly, our reduction from orienteering to {deadline, monotone-reward} TSP also applies to *asymmetric metrics* (see Remark 3.8). The recent result of Svensson et al. [26] establishing an $O(1)$-integrality gap for the ATSP LP-relaxation implies (due to the work of [21]) an $O(\log n)$-approximation for directed orienteering; thus our reduction yields an $O(\log n)$-approximation for {deadline, monotone-reward} TSP in asymmetric metrics in $O\big(n^{O(\log n\Delta)}\big)$ time. While this does not improve upon the quasi-polytime $O(\log n)$-approximation for asymmetric deadline TSP that follows from the work of Chekuri and Pál [8], our reduction does show that improved (e.g., constant-factor) approximations to directed orienteering will *directly* translate to analogous improvements for directed deadline TSP in $O\big(n^{O(\log n\Delta)}\big)$ time.

In Section 4.2, we show that our results yield $O(1)$-approximation for some other variants of {deadline, monotone-reward} TSP, namely, settings where: (a) the path must start *and* end at some given nodes; and (b) nodes have release dates but no deadlines, or more generally have a non-decreasing reward function, and we seek a length-bounded path gathering maximum reward (and we are allowed to wait at nodes while traversing a path). We also consider *orienteering with time windows*, wherein we have a time window for each node and we collect its reward if it is visited within its time window. By combining our result for deadline TSP with certain results of Chekuri et al. [6], we obtain an $O\big(\log \frac{L_{\max}}{L_{\min}}\big)$-approximation for orienteering with time windows in $O\big(n^{\log n\Delta}\big)$ time, where $L_{\max}$ and $L_{\min}$ are the lengths of the longest and shortest time windows with non-zero length. This improves upon the approximation guarantee of [6] when the optimum value is large; in particular, if $L_{\max} = O(L_{\min})$, we obtain an $O(1)$-approximation, whereas [6] obtain an $O(\log(\text{optimal value}))$-approximation, but in polynomial time.

[1] More precisely, if *NP* $\not\subseteq$ *DTIME* $[O\big(n^{O(\log n)}\big)]$, then obtaining an $O(1)$-approximation in graphical metrics cannot be *NP*-hard. Also, any reduction showing *NP*-hardness of an $O(1)$-approximation in general metrics must involve exponentially large distances.

Our techniques. We briefly discuss the techniques that we utilize to obtain our result for deadline TSP (see also "Overview and intuition" in Section 3). We begin by guessing a suitable set of $O(\log n\Delta)$ nodes, which are a subsequence of the nodes encountered along an optimal path P^*. This guessing step requires some care, in light of the fact that we are dealing with *hard* deadlines – i.e., we need to satisfy deadlines *exactly* (and not just approximately) – which rules out certain standard approaches. For instance, a natural attempt would be to view nodes of P^* as being grouped into geometric buckets based on their visiting times and/or deadlines and guess the boundary nodes of the buckets; since nodes in a bucket involve roughly the same visiting time and/or deadline, it is tempting to solve a P2P orienteering instance with the boundary nodes as end-points. But this is too coarse an idea that is incompatible with hard deadlines. Indeed, since one cannot merge similar deadlines/visiting times, any such approach faces the issue that there could be $\Omega(n)$ distinct visiting times and deadlines to consider. Not surprisingly, [7], who take such an approach make the strong assumption that there are only $O(1)$ distinct deadlines. Alternatively, other works on deadline TSP [2, 6] are based on extracting a subset of P^* with a simpler deadline and visiting-time structure, but at the expense of an $O(\log n)$-factor loss in objective.

We seek to avoid both the above bottlenecks, but, as alluded to above, honing in on the correct choice of guessed nodes requires some insight and a more refined approach. At a high level, the sequence of nodes we guess $v_0 := r, v_1, \dots, v_{\log n\Delta}$ has the property that the length of the v_i-v_{i+1} portion of P^*, which we denote by $P^*_{v_i v_{i+1}}$, is about $c_{v_i v_{i+1}} + \gamma^i$, where γ is some constant (we use $\gamma = 1.5$); equivalently, we say that the v_i-relative regret-length of $P^*_{v_i v_{i+1}}$ is about γ^i. We still obtain P2P-orienteering solutions (with the above distance bound) between consecutive v_i, such that they cumulatively collect enough reward (but they may not have disjoint node-sets). Concatenating these yields a path P that may violate some deadlines. A key insight is that if we were to shortcut past the v_{i-2}-v_i portion of P, then each client visited between v_i and v_{i+1} after shortcutting, *now has its deadline satisfied*: we "save" roughly $\gamma^{i-2} + \gamma^{i-1} \geq \gamma^i$ distance in this shortcutting process, which is enough to guarantee that clients visited between v_i and v_{i+1} in our algorithm have their deadlines satisfied exactly. Thus, we can cover P using a constant number of deadline-TSP solutions. Our eventual algorithm is slightly more involved as we have to account for large leaps in distances as well (thereby requiring us to guess an additional $\log n\Delta$ nodes).

Finally, we consider two approaches for obtaining the P2P-orienteering solutions. The first approach is based on viewing the problem as an instance of the *maximum coverage problem with group budgets* [7], for which a simple greedy algorithm yields a good approximation. The second is a more-sophisticated LP-based approach that yields a somewhat better approximation guarantee. We write an LP to find paths between guessed nodes, and extract from the LP solution a distribution over P2P-orienteering solutions between consecutive guesses. Randomly picking one such path for each v_i will ensure each client is covered with probability proportional to the extent that the LP covers it. While recent LP-based insights [14] for orienteering allow us to obtain a compact LP to obtain such distributions, one can also work with a configuration-style LP that directly encodes the distribution requirement. *Any* P2P-orienteering approximation algorithm can then be used to approximately separate the dual of this LP, and hence yield the desired distributions.

Our $O(n^{\log n\Delta})$ running time stems from the need for enumerating $O(\log n\Delta)$ nodes. There is however some hope that this enumeration step and its resulting run-time blowup can be circumvented. At a high level, the search space of our algorithm can be represented by a directed layered graph, whose vertices in layer i encode the various choices for the v_i-v_{i+1} portion of P^*, and arcs encode compatible choices. While we use brute-force enumeration to find a suitable path in this digraph, one can envisage other means for finding this path, e.g., dynamic programming or linear programming. An important and useful fact to note is that we only have $O(\log n\Delta)$ layers, so that, for $\mathrm{poly}(n,\Delta)$ running time (hence, polytime for graphical metrics), one can afford to take time exponential in the number of layers. (We note that such savings were achieved in the context of another VRP, namely the directed latency problem: Nagarajan and Ravi [20] gave an $O\big(n^{\log n}\big)$-time, $O(\log n)$-approximation based on guessing $O(\log n)$ nodes, and subsequently Friggstad et al. [12] obtained the same approximation in polytime by showing, in essence (roughly speaking), that this guesswork can be eliminated by formulating a suitable LP to provide the guessed nodes.)

Related work. We limit ourselves to a discussion of the work that is most relevant to the max-reward VRP problems we consider; we refer the reader to [27] for more information on vehicle routing problems in general. As mentioned earlier, the current-best approximation factor for deadline TSP is $O(\log n)$ due to Bansal et al. [2]. They also give a bicriteria $O(\log 1/\epsilon)$-approximation that violates deadlines by at most a $(1+\epsilon)$-factor; with integer deadlines this yields an $O(\log D_{\max})$-approximation (where $D_{\max} := \max_v D_v$). Since we obtain $O\big(n^{O(\log n\Delta)}\big)$ running time, it is also relevant to compare our result with the recursive greedy approach of Chekuri and Pál [8]. This is a versatile approach that yields logarithmic approximation guarantees for various problems (including deadline TSP) in quasi-polytime. However, this approach seems hard-pressed to yield anything *better* than logarithmic guarantees (even for deadline TSP); in particular, the $\Omega(\log n)$-inapproximability result for submodular orienteering[2] suggests that one cannot improve their approach to obtain $o(\log n)$ approximation guarantees.

The special case of deadline TSP with uniform node-deadlines is called *rooted orienteering*. Blum et al. [3] obtained the first $O(1)$-approximation algorithm for this problem. Their ideas were refined by [2, 6] to obtain the current-best $(2+\epsilon)$-approximation, which also applies to P2P orienteering [6]. Recently, Friggstad and Swamy [14] developed a different LP-based approach for orienteering, which also yields (slightly inferior) $O(1)$-approximations for these problems. We utilize some of their insights in our work.

Various (other) generalizations of orienteering and deadline TSP have also been studied. Bansal et al. [2] give an $O(\log^2 n)$-approximation for orienteering with time windows, and an $O(\log D_{\max})$ approximation with integer release dates and deadlines. They make the informal remark that a deadline-TSP approximation that relies on an α-approximation for P2P orienteering will translate to an α^2-approximation for orienteering with time windows. However, this comment seems to be in the *specific* context of their approach, and it is unclear if it can be applied with our framework to obtain an $O(1)$-approximation for the time-windows problem in $O\big(n^{O(\log n\Delta)}\big)$; such a result would be quite interesting. Chekuri et al. [6] show that an α-approximation for P2P orienteering yields an $O\big(\alpha \max\{\log \mathsf{opt}, \log \frac{L_{\max}}{L_{\min}}\}\big)$-approximation for orienteering with time windows, where opt is the optimal value, and $L_{\max}$ and $L_{\min}$ are the lengths of the longest and shortest time windows with non-zero length.

Blum et al. [3] considered the special case of monotone-reward TSP called discounted-reward TSP (wherein the reward of node v at time t is $\pi_v\gamma^{-t}$), and devised a 6.753-approximation algorithm. This factor was slightly improved to 5.195 by [10]. Chekuri

[2] Submodular orienteering captures "group orienteering", wherein we are given (disjoint) groups of vertices, and the reward of a path is the number of groups it hits. Using set-cover ideas, one can show that an α-approximation for group orienteering yields an $O(\alpha \log n)$-approximation for the group Steiner tree problem. The $\Omega(\log^2 n)$-inapproximability result for group Steiner tree [16] thus translates to an $\Omega(\log n)$-inapproximability result for submodular orienteering.

and Pál [8] show that their recursive-greedy approach yields a quasi-polytime $O(\log n)$-approximation for a generalization of monotone-reward TSP where the node reward is an *arbitrary* function of time. They obtain the same guarantee for a further generalization of orienteering (even in asymmetric metrics) that they introduce, called *submodular orienteering with time windows*, where the reward of a path is given by a monotone submodular function of the set of nodes visited within their time windows.

In *asymmetric metrics*, Chekuri et al. [6] give an $O(\log^2 \mathsf{opt})$-approximation for P2P orienteering; this also yields guarantees for the time-windows problem via their aforementioned reduction (which also applies to asymmetric metrics). The current-best approximation for directed orienteering is $O(\log n)$, which follows by combining the $O(1)$-integrality gap for the ATSP LP [26] with a result of [21] showing that an LP-relative α-approximation for ATSP yields an $O(\alpha \log n)$-approximation for directed orienteering.

Finally, there is a wealth of literature on fixed-node-set VRPs; we refer the reader to some of the most recent work on these problems [25, 29, 26] for further pointers.

2 Preliminaries and notation

All the problems we consider involve an underlying complete graph $G = (\{r\} \cup V, E)$, where r is a distinguished root node, and metric edge costs $\{c_{uv}\}$. By scaling, we may assume that c_{uv} is an integer for all $u, v \in V \cup \{r\}$. Let $\Delta := \max_{u,v \in V \cup \{r\}} c_{uv}$ be the diameter of the scaled metric space. For a set S of edges, we often use $c(S)$ to denote $\sum_{e \in S} c_e$. We often use V' to denote $\{r\} \cup V$. Let $n = |V'| = |V| + 1$. We call a path P in G rooted if it begins at r. We always think of the nodes on a rooted path P as being ordered in increasing order of their distance along P from r. For any path P and nodes $u, v \in P$, let P_{uv} denote the u-v subpath of P. For a rooted path P and node $v \in P$, the visiting time of v is the distance from r to v along P, which we denote by $c_P(v) := c(P_{rv})$. To avoid excessive notation, we will view a path P sometimes as its edge-set, and sometimes as its node-set; the meaning will be clear from the context.

In *deadline TSP*, each node $v \in V$ has a deadline $D_v \geq 0$ and reward $\pi_v \geq 0$. For notational convenience, set $D_r = \pi_r = 0$. The goal is to find a simple rooted path P such that $c_P(v) \leq D_v$ for all $v \in P$ that maximizes $\sum_{v \in P} \pi_v$. We may assume that $D_v \geq c_{rv}$ for all $v \in V$, as otherwise v can never be visited by a feasible solution and we can simply delete v from our metric space.

Monotone-reward TSP is a substantial generalization of deadline TSP, wherein each node v has a *non-increasing reward function* $\pi_v : \mathbb{R}_+ \mapsto \mathbb{R}_+$. For notational convenience, set $\pi_r(x) = 0$ for all x. The goal is to find a simple rooted path P that maximizes $\sum_{v \in P} \pi_v(c_P(v))$. We assume that each $\pi_v(.)$ is specified via a value oracle, that on input x returns $\pi_v(x)$.

Regret distances. For any $u \in V \cup \{r\}$, and any ordered pair $v, w \in V \cup \{r\}$, define the *regret distance* of (v, w) with respect to u to be $c_u^{\mathsf{reg}}(v, w) := c_{uv} + c_{vw} - c_{uw}$. The regret distances $\{c_u^{\mathsf{reg}}(v, w)\}_{v,w \in V \cup \{r\}}$ form an *asymmetric* metric. The regret-length of a path P with respect to its start node is called the excess of P in [3, 2, 6]. A simple but key insight that we will repeatedly use is that if P is a rooted path, and u, v are nodes on P where u comes before v (recall that nodes on P are ordered by increasing $c_P(\cdot)$), then replacing P_{uv} by the edge uv reduces the length of the path by exactly $c_u^{\mathsf{reg}}(P_{uv}) = c(P_{uv}) - c_{uv}$.

Point-to-point (P2P) orienteering. In P2P orienteering, we have a start node s, *end node* t, and a length bound B, and we seek an s-t path P with $c(P) \leq B$ that maximizes $\sum_{v \in P} \pi_v$. We will often need to restrict the path to only visit nodes from a certain subset $N \subseteq V'$

(where $\{s,t\} \subseteq N$); we refer to this as P2P orienteering with node set N. It will often be convenient to cast the length-bound B, as a regret-bound of $B - c_{st}$ on the regret of P with respect to s, i.e., $c_s^{\mathsf{reg}}(P)$.

Observe that P2P orienteering can be cast as a special case of deadline TSP (with root node s) by setting $D_v = \max\{0, B - c_{vt}\}$ for all $v \in V$: if w is the end-node of a feasible solution to this deadline-TSP instance, then we can always append the edge wt to this path and stay within the length bound of $B = D_t$.

Recently, Friggstad and Swamy [14] devised an LP-based 6-approximation algorithm for P2P orienteering by formulating a polynomial-size LP for the problem and devising an LP-rounding algorithm (see Section 5). Their algorithm yields a distribution of P2P-orienteering solutions that visits each node with probability proportional to the extent it is visited in the LP solution. Our LP-rounding based algorithm (see subroutine LP-Round) for deadline TSP makes use of the latter type of distributional guarantee. In fact, *any* approximation algorithm for P2P orienteering can be used to obtain such a distribution – this follows from [4, 18] – and this leads to our $(7.63 + \epsilon)$-approximation algorithm for deadline TSP.

3 Constant-factor approximation for deadline TSP

We now describe our constant-factor approximation algorithm for deadline TSP that runs in time $O\big(n^{O(\log n\Delta)}\big)$. Let P^* be an optimal path, and $\mathsf{opt} = \pi(P^*)$ be the optimal value. We prove the following result.

▶ **Theorem 3.1.** *Algorithm 1 runs in $O\big(n^{O(\log n\Delta)}\big)$ time. Let $\mathcal{A}$ be an α-approximation algorithm for* P2P *orienteering (where $\alpha \geq 1$).*

(a) *Using subroutine* Greedy *in step D1.2 (with algorithm $\mathcal{A}$), Algorithm 1 returns a deadline-TSP solution whose reward is at least $\frac{1}{3(\alpha+1)} \cdot \mathsf{opt}$.*

(b) *Using subroutine* LP-Round *in step D1.2 (with algorithm $\mathcal{A}$), Algorithm 1 returns a deadline-TSP solution with a slightly better expected reward of at least $\frac{1}{3/(1-e^{-1/\alpha})} \cdot \mathsf{opt} \geq \frac{1}{3(\alpha+1)} \cdot \mathsf{opt}$. This guarantee can be derandomized.*

Taking $\alpha = (2 + \epsilon)$ above [6], we obtain a $(9 + \epsilon)$-approximation using Greedy, and an improved $(7.63 + \epsilon)$-approximation using the LP-approach in LP-Round.

Overview and intuition. We first give an overview of the algorithm and convey the underlying intuition; the detailed description appears below as Algorithm 1. Let $\gamma = 1.5$. Note that $\gamma^2 \leq \gamma + 1$.

Let $R^* = c_r^{\mathsf{reg}}(P^*)$ be the regret of P^* with respect to the root r. Set $u_0 := r$. Suppose we "guess" (i.e., enumerate all possible choices for) a sequence of nodes $w_0, u_1, w_1, \ldots$ occurring on P^* (in this order), which are defined as follows. Given u_i for $i \geq 0$, we define w_i, u_{i+1} as follows. Let u_{i+1} be the first node on P^* after u_i such that the regret of $P^*_{u_i u_{i+1}}$ with respect to u_i is at least γ^i. So we have $c_{u_i}^{\mathsf{reg}}(P_{u_i u_{i+1}}) \geq \gamma^i$ and $c_{u_i}^{\mathsf{reg}}(P^*_{u_i v}) < \gamma^i$ for all nodes v prior to u_{i+1} on $P^*_{u_i u_{i+1}}$; since all distances are integers, this implies that $c_{u_i}^{\mathsf{reg}}(P^*_{u_i v}) \leq \lceil \gamma^i \rceil - 1$ for all nodes v prior to u_{i+1} on $P^*_{u_i u_{i+1}}$. Define w_i to be the predecessor of u_{i+1}. If there is no such node u_{i+1}, then u_{i+1} is undefined, and *define* w_i to be the end-node of P^*. Let k be the largest index such that u_k is well defined. Observe that $k = O(\log R^*)$ since for every $i = 0, \ldots, k-1$, we have that $\gamma^i \leq c_{u_i}^{\mathsf{reg}}(P^*_{u_i u_{i+1}}) \leq c_r^{\mathsf{reg}}(P^*_{u_i u_{i+1}})$, and $c_r^{\mathsf{reg}}(P^*) = R^*$. This leads to the $O\big(n^{O(\log n\Delta)}\big)$ running time, since we need to consider $n^{O(\log R^*)}$ guesses for $w_0, \ldots, u_k, w_k$, and $R^* \leq c(P^*) \leq n\Delta$.

We first observe that we can obtain the following lower bound on the deadlines of nodes in $P^*_{u_iw_i}$. The proof of the following lemma is deferred to the analysis.

▶ **Lemma 3.2.** *Consider any index $i = 0, \dots, k$. The visiting time of a node $v \in P^*_{u_iw_i}$, and hence its deadline D_v, is at least $\mathsf{lb}_{i,v} := \sum_{j=0}^{i-1} \max\{c_{u_jw_j} + c_{w_ju_{j+1}}, c_{u_ju_{j+1}} + \gamma^j\} + c_{u_iv}$.*

Lemma 3.2 implies that the P2P-orienteering instance with node-set $N^i := \{v \in V' : D_v \geq \mathsf{lb}_{i,v}\}$, start node u_i, end node w_i, and regret-bound $\lceil \gamma^i \rceil - 1$ (with respect to u_i), has optimal value at least $\pi(P^*_{u_iw_i})$. (Note that $N^0 = V'$.) Suppose that we are able to find paths $Q^0, Q^1, \dots, Q^k$, such that:

(i) for every $i = 0, \dots, k$, we have that Q^i is a u_i-w_i path, visits only nodes of N^i, and $c^{\mathsf{reg}}_{u_i}(Q^i) \leq \lceil \gamma^i \rceil - 1$; and

(ii) $\pi\big(Q^0 \cup \ldots \cup Q^k\big) \geq \rho \cdot \pi(P^*)$, where $0 < \rho \leq 1$ is some constant.

We show that we can use these paths to obtain a deadline-TSP solution of value $\rho \cdot \pi(P^*)/3$; this yields an $O(1)$-approximation for deadline TSP.

Let Z be the path obtained by concatenating all (the nodes of) $Q^0, \dots, Q^k$. How "far" is Z from being a feasible solution? Assume that the Q^is are node-disjoint (which we can always ensure by shortcutting past all occurrences of a node other than its first occurrence). Consider a node $v \in Q^i$. We can upper bound the visiting time of v by

$$\begin{cases} \sum_{j=0}^{i-1}(c_{u_jw_j} + \lceil \gamma^j \rceil - 1 + c_{w_ju_{j+1}}) & \text{if } v = u_i; \\ \sum_{j=0}^{i-1}(c_{u_jw_j} + \lceil \gamma^j \rceil - 1 + c_{w_ju_{j+1}}) + c_{u_iv} + \lceil \gamma^i \rceil - 1 & \text{otherwise,} \end{cases}$$

where in the latter case, $c_{u_iv} + \lceil \gamma^i \rceil - 1$ upper bounds the time taken to go from u_i to v along Z (as $\lceil \gamma^i \rceil - 1 \geq c^{\mathsf{reg}}_{u_i}(Q^i) \geq c^{\mathsf{reg}}_{u_i}(Q^i_{u_iv})$). For $v \in Q^0$, this shows that its visiting time is at most $c_{u_0v} = \mathsf{lb}_{0,v}$, which is at most D_v. Nodes in $Q^1 \cup \ldots \cup Q^k$ may however be visited after their deadlines.

The chief insight is that if we replace the u_j-u_{j+1} portion of Z, which currently consists of the node-sequence Q^j, by the direct edge u_ju_{j+1}, then we incur a γ^j-savings in the (above upper bound for) visiting times of nodes on Q^i for $i > j$: the term $c_{u_jw_j} + \lceil \gamma^j \rceil - 1 + c_{w_ju_{j+1}}$ in the above upper bound gets replaced by $c_{u_ju_{j+1}}$, and $c^{\mathsf{reg}}_{u_j}(P^*_{u_jw_j}) \leq \lceil \gamma^j \rceil - 1$ implies that

$$c_{u_jw_j} + \lceil \gamma^j \rceil - 1 + c_{w_ju_{j+1}} \geq c(P^*_{u_jw_j}) + c_{w_ju_{j+1}} = c(P^*_{u_ju_{j+1}}) = c^{\mathsf{reg}}_{u_j}(P^*_{u_ju_{j+1}}) + c_{u_ju_{j+1}} \geq \gamma^j + c_{u_ju_{j+1}}.$$

Moreover, since $\gamma^2 \leq \gamma + 1$, this implies that if we "sync up" with P^* at w_{j-1} by visiting w_{j-1} by time $\mathsf{lb}_{j-1,w_{j-1}}$, then deleting the u_j-u_{j+2} portion – i.e., going directly to u_{j+2} from w_{j-1} – ensures that: (a) every $v \in Q^{j+2}$ is visited by its deadline (and, in fact, by $\mathsf{lb}_{j+2,v}$), and (b) we remain in sync with P^* at w_{j+2}. Note also that Z is in sync with P^* at nodes u_0, w_0. Finally, for nodes in Q^1, it suffices to replace the u_0-u_1 portion of Z by the direct edge u_0u_1 in order to visit these nodes by their deadlines, and have w_1 sync up with P^*.

The upshot of these insights is that: (1) for any $\ell \in \{0, 1, 2\}$, the path $Z^{(\ell)}$ given by the node-sequence $r, \{Q^j\}_{0 \leq j \leq k: j = \ell \bmod 3}$ (where r is possibly a duplicated node) is feasible (Lemma 3.3); and (2) together these paths cover all the nodes of Z. Hence, the best of these 3 paths collects reward at least $\pi\big(Q^0 \cup \ldots \cup Q^k\big)/3 \geq \frac{\rho}{3} \cdot \pi(P^*)$.

Finally, we discuss two approaches for finding the $Q^0, \dots, Q^k$ paths. The first approach is based on observing that the problem of finding these paths to maximize $\pi\big(Q^0 \cup \ldots \cup Q^k\big)$ is an instance of the *maximum coverage problem with group budgets* considered by [7]. Thus, a simple greedy approach (subroutine Greedy) works, where we repeatedly find $Q^0, Q^1, \dots$ in that order, and to find Q^i, we use a P2P-orienteering α-approximation algorithm with the subset of N^i that has not been covered by Q^j for $j < i$ (or equivalently, we zero out the rewards of nodes in $Q^0 \cup \ldots \cup Q^{i-1}$). Chekuri and Kumar [7] show that this yields a collection of paths that together obtain reward at least $\mathsf{opt}/(\alpha+1)$; we include the analysis for completeness (Lemma 3.4).

The second approach is a more sophisticated LP-based approach (subroutine LP-Round) that yields a better guarantee. We write a *configuration LP* (Ap-P) to find the Q^i paths. We use variables x^i_v to denote the extent to which v lies in $P^*_{u_i w_i}$, for $i \geq 0$. Let $\mathcal{P}^i := \mathcal{P}^i(u_i, w_i, N^i, \lceil \gamma^i \rceil - 1)$ be the collection of simple u_i-w_i paths of length at most $c_{u_i w_i} + \lceil \gamma^i \rceil - 1$ that visit only nodes of N^i. (Note that if u_i or w_i is not in N^i, then $\mathcal{P}^i = \emptyset$. Also, if $u_i = w_i$ and lies in N^i, then $\mathcal{P}^i$ consists of only the trivial singleton path $\{u_i\}$.) We also have variables $\{z^i_P\}_{P \in \mathcal{P}^i}$, where z^i_P denotes that we choose path $P \in \mathcal{P}^i$. While we cannot solve this LP optimally, we show (see Lemma 3.6) that an α-approximation algorithm for P2P orienteering can be used to obtain a solution $(\overline{x}, \overline{z})$ of objective value at least $OPT_{\text{Ap-P}}$, where each node v is covered by the paths from $\mathcal{P}^i$ to an extent of at least $\overline{x}^i_v/\alpha$. We choose $Q^i \in \mathcal{P}^i$ by sampling from the $\{\overline{z}^i_P\}_{\in \mathcal{P}^i}$ distribution. It is not hard to argue then that $\mathrm{E}\big[\pi\big(\bigcup_{i=2}^k Q^i\big)\big] \geq \big(1 - e^{-1/\alpha}\big) OPT_{\text{Ap-P}}$ (Lemma 3.7), which yields the guarantee stated in Theorem 3.1 (b). We now describe the algorithm in detail and proceed to analyze it.

Algorithm 1 Deadline TSP

Let $\gamma = 1.5$.

Input: metric $(\{r\} \cup V, c)$, deadlines $\{D_v\}_{v \in V}$, rewards $\{\pi_v\}_{v \in V}$; an α-approximation algorithm $\mathcal{A}$ for P2P orienteering.

Output: An r-rooted path P such that every $v \in P$ is visited by time D_v.

D1 Initialize $\mathcal{Q} \leftarrow \emptyset$. Let $u_0 := r$. For $k = 0, 1, \ldots, \frac{\log n\Delta}{\log \gamma}$, and every choice of nodes $w_0, u_1, w_1, \ldots, u_k, w_k$, perform the following steps.

- **D1.1** For $i = 0, \ldots, k$, and $v \in V'$, define $\mathsf{lb}_{i,v} := \sum_{j=0}^{i-1} \max\{c_{u_j w_j} + c_{w_j u_{j+1}}, c_{u_j u_{j+1}} + \gamma^j\} + c_{u_i v}$. Define $N^i := \{v \in V' : D_v \geq \mathsf{lb}_{i,v}\}$. If $u_i \notin N^i$ or $w_i \notin N^i$ for some $i \in \{0, \ldots, k\}$, then reject this guess – i.e., omit steps 12–14 – and move on to the next choice of u_i, w_i nodes.
- **D1.2** Use subroutine Greedy, or subroutine LP-Round below to obtain paths $Q^0, Q^1, \ldots, Q^k$, where each Q^i is a u_i-w_i path with $c^{\mathsf{reg}}_{u_i}(Q^i) \leq \lceil \gamma^i \rceil - 1$ visiting only nodes of N^i.
- **D1.3** For $\ell \in \{0, 1, 2\}$, let $Z^{(\ell)}$ be the path given by the node-sequence $r, \{Q^j\}_{0 \leq j \leq k : j = \ell \bmod 3}$.
- **D1.4** Add the path $Z^{\max} \in \{Z^{(0)}, Z^{(1)}, Z^{(2)}\}$ that gathers maximum reward to $\mathcal{Q}$.

D2 Return the best solution found in $\mathcal{Q}$, shortcutting the path to retain only the first occurrence of each node.

Subroutine Greedy.

G1 For $i = 0, \ldots, k$, use algorithm $\mathcal{A}$ to (approximately) solve the P2P-orienteering instance with start node u_i, end node w_i, node-set $N^i \setminus \big(\bigcup_{j=0}^{i-1} P^j\big)$, and length bound $c_{u_i w_i} + \lceil \gamma^i \rceil - 1$ (and node rewards $\{\pi_v\}_{v \in V'}$), to obtain a simple path P^i (so if $u_i = w_i$, then $P^i = \{u_i\}$ or $P^i = \emptyset$).
Return paths $P^0, \ldots, P^k$.

Subroutine LP-Round.

L1 Let $\mathcal{P}^i := \mathcal{P}^i(u_i, w_i, N^i, \lceil\gamma^i\rceil - 1)$ denote the collection of simple u_i-w_i paths with $c^{\mathsf{reg}}_{u_i}$-length at most $\lceil\gamma^i\rceil - 1$ (so the c-length is at most $c_{u_iw_i} + \lceil\gamma^i\rceil - 1$) and visiting only nodes from N^i. Note that if $u_i = w_i$ and $u_i \in N^i$, then $\mathcal{P}^i$ consists of only the trivial singleton path $\{u_i\}$. Consider the following LP.

$$\max \quad \sum_{i=0}^{k}\sum_{v\in V} \pi_v x_v^i \tag{Ap-P}$$

$$\text{s.t.} \quad \sum_{P\in\mathcal{P}^i} z_P^i \le 1 \qquad \forall i = 0,\dots,k \tag{1}$$

$$x_v^i \le \sum_{P\in\mathcal{P}^i : v\in P} z_P^i \qquad \forall v \in V,\ \forall i = 0,\dots,k \tag{2}$$

$$\sum_{i=0}^{k} x_v^i \le 1 \qquad \forall v \in V \tag{3}$$

$$x_{u_i}^i = x_{w_i}^i = 1 \qquad \forall i = 0,\dots,k \tag{4}$$

$$x, z \ge 0.$$

L2 Use Lemma 3.6 to obtain in polytime a solution $(\overline{x}, \overline{z})$ (with polynomial-size support) such that: (i) $\sum_{i=0}^{k}\sum_{v\in V}\pi_v\overline{x}_v^i \ge OPT_{\text{Ap-P}}$, (ii) $\sum_{P\in\mathcal{P}^i : v\in P}\overline{z}_P^i \ge \overline{x}_v^i/\alpha$ for all $v \in V$ and $i = 0,\dots,k$, and (iii) $(\overline{x}, \overline{z})$ satisfies the remaining constraints of (Ap-P).

L3 For each $i = 0,\dots,k$, sample a random path $P^i \in \mathcal{P}^i$ from the $\{\overline{z}_P^i\}_{P\in\mathcal{P}^i}$ distribution. Return paths $P^0,\dots,P^k$.

Analysis. The running time stated in Theorem 3.1 follows since $k = O(\log n\Delta)$, and for each k, we enumerate over all sequences of $2k$ nodes.

Recall that P^* is an optimal solution. Let $\mathsf{opt} = \pi(P^*)$ be the optimal value. We will assume in the sequel that we have the right choice of k and the u_i, w_i nodes. Recall that $u_0 = r$. That is, w_0 is the last node on P^* such that $c^{\mathsf{reg}}_{u_0}(P_{u_0w_0}) = 0$. Given u_i, for $i \ge 0$, we have that u_{i+1} is the first node v on P^* after u_i such that $c^{\mathsf{reg}}_{u_i}(P_{u_iv}) \ge \gamma^i$, and w_i is the predecessor of u_{i+1}. If there is no such node u_{i+1}, then u_{i+1} is undefined and w_i is the the end-node of P^*. Also, k is the largest index such that u_k is well defined. Note that $k \ge 0$.

We begin by proving Lemma 3.2.

Proof of Lemma 3.2. For any $j \in \{0,\dots,i-1\}$, we have that $c(P^*_{u_ju_{j+1}}) \ge c(P^*_{u_jw_j}) + c_{w_ju_{j+1}} \ge c_{u_jw_j} + c_{w_ju_{j+1}}$ and also $c(P^*_{u_ju_{j+1}}) = c^{\mathsf{reg}}_{u_j}(P^*_{u_ju_{j+1}}) + c_{u_ju_{j+1}} \ge \gamma^j + c_{u_ju_{j+1}}$, where the inequality follows from the definition of u_{j+1}. The visiting time of $v \in P^*_{u_iw_i}$ is $c_{ru_0} + \sum_{j=0}^{i-1} c(P^*_{u_ju_{j+1}}) + c_{u_iv}$, which, using the above bounds, is at least $\mathsf{lb}_{i,v}$. ◀

Next, Lemma 3.3 shows that the paths $Z^{(\ell)}$ obtained in step D1.3 are feasible. Since these paths together cover $\bigcup_{i=0}^{k} Q^i$, the path $Z^{\max}$ returned in step D1.4 gathers reward at least $\pi\big(\bigcup_{i=0}^{k} Q^i\big)/3$.

▶ **Lemma 3.3.** *Consider $\ell \in \{0,1,2\}$, and the path $Z^{(\ell)}$ computed in step D1.3. For every Q^j that is part of $Z^{(\ell)}$, and every $v \in Q^j$, the visiting time of this occurrence of v in $Z^{(\ell)}$ is at most $\mathsf{lb}_{j,v} \le D_v$.*

Proof. Consider any Q^j that is part of $Z^{(\ell)}$. So $0 \le j \le k$ and $j = \ell \bmod 3$. We argue by induction on j that the visiting time (in $Z^{(\ell)}$) of every node $v \in Q^j$ is at most $\mathsf{lb}_{j,v}$.

For the base case, when $j = \ell$, the visiting time of any $v \in Q^\ell$ is at most $c_{u_0 u_\ell} + c_{u_\ell v} + \lceil \gamma^\ell \rceil - 1$ since $c^{\mathsf{reg}}_{u_\ell}(Q^\ell_{u_\ell v}) \le c^{\mathsf{reg}}_{u_\ell}(Q^\ell) \le \lceil \gamma^\ell \rceil - 1$. We can check by inspection that this bound is always at most $\mathsf{lb}_{\ell,v}$:

- for $\ell = 0$, the bound is $c_{u_0 v} = \mathsf{lb}_{0,v}$;
- for $\ell = 1$, the bound is $c_{u_0 u_1} + c_{u_1 v} + 1 \le \mathsf{lb}_{1,v}$;
- for $\ell = 2$, the bound is at most $c_{u_0 u_2} + c_{u_2 v} + \gamma^2 \le (c_{u_0 u_1} + 1) + (c_{u_1 u_2} + \gamma) + c_{u_2 v} \le \mathsf{lb}_{2,v}$.

Now suppose $j > \ell$. The visiting time of any $v \in Q^j$ is at most (visiting time of w_{j-3}) + $c_{w_{j-3} u_j} + c_{u_j v} + \gamma^j$ since the $c^{\mathsf{reg}}_{u_j}$-length of $Q^j_{u_j v}$ is at most $\lceil \gamma^j \rceil - 1 \le \gamma^j$. By our induction hypothesis, the visiting time of w_{j-3} is at most $\mathsf{lb}_{j-3,w_{j-3}}$. Since $\gamma^j \le \gamma^{j-2} + \gamma^{j-1}$, we can upper bound the visiting time of v by

$$\big(\mathsf{lb}_{j-3,w_{j-3}} + c_{w_{j-3} u_{j-2}}\big) + \big(c_{u_{j-2} u_{j-1}} + \gamma^{j-2}\big) + \big(c_{u_{j-1} u_j} + \gamma^{j-1}\big) + c_{u_j v} \le \mathsf{lb}_{j,u_j} + c_{u_j v} \le \mathsf{lb}_{j,v}.$$

The first inequality is because for any index $i \ge 1$, we have $\mathsf{lb}_{i,u_i} \ge \mathsf{lb}_{i-1,w_{i-1}} + c_{w_{i-1} u_i}$ and $\mathsf{lb}_{i,u_i} \ge \mathsf{lb}_{i-1,u_{i-1}} + c_{u_{i-1} u_i} + \gamma^{i-1}$. This completes the induction step and proves the lemma. ◀

Finally, we prove guarantees for the paths returned by subroutine Greedy and subroutine LP-Round.

▶ **Lemma 3.4** (Follows from [7]). *The paths $Q^0, \ldots, Q^k$ returned if we use subroutine* Greedy *in step D1.2 satisfy $\pi\big(Q^0 \cup \ldots \cup Q^k\big) \ge \mathsf{opt}/(\alpha + 1)$.*

Proof. Clearly, for each $i = 0, \ldots, k$, we have that $P^*_{u_i w_i} \setminus \big(\bigcup_{j=0}^{i-1} Q^j\big)$ is a feasible solution to the P2P-orienteering instance that is fed as input to algorithm $\mathcal{A}$ in iteration i. So for each $i = 0, \ldots, k$, we have

$$\pi\Big(Q^i \setminus \big(\bigcup_{j=0}^{i-1} Q^j\big)\Big) \ge \frac{1}{\alpha} \cdot \pi\Big(P^*_{u_i w_i} \setminus \big(\bigcup_{j=0}^{i-1} Q^j\big)\Big). \tag{5}$$

Adding the above for $i = 0, \ldots, k$ yields an inequality whose LHS is $\pi\big(Q^0 \cup \ldots \cup Q^k\big)$, and whose RHS is at least $\frac{1}{\alpha} \cdot \pi\big(P^* \setminus (Q^0 \cup \ldots \cup Q^k)\big) \ge \frac{1}{\alpha} \cdot \big[\pi(P^*) - \pi\big(Q^0 \cup \ldots \cup Q^k\big)\big]$. It follows that $\pi\big(Q^0 \cup \ldots \cup Q^k\big) \ge \pi(P^*)/(\alpha + 1)$. ◀

Combining Lemma 3.4 with Lemma 3.3 leads to the proof of Theorem 3.1 (a).

Part (b) of Theorem 3.1

We now analyze the paths returned by subroutine LP-Round and prove Theorem 3.1 (b).

We first observe in Claim 3.5 that $OPT_{\text{Ap-P}} \ge \pi(P^*) = \mathsf{opt}$. Lemma 3.6 shows that using our P2P-orienteering approximation algorithm, we can obtain in polytime a solution $(\overline{x}, \overline{z})$ satisfying the properties stated in step L2. Given this, Lemma 3.7 proves that the random paths $Q^0, \ldots, Q^k$ returned by subroutine LP-Round satisfy $\mathrm{E}\big[\pi\big(\bigcup_{i=0}^k Q^i\big)\big] \ge \big(1 - e^{-1/\alpha}\big) OPT_{\text{Ap-P}}$. This yields the randomized guarantee stated in Theorem 3.1 (b). We then show how to derandomize the algorithm without affecting its guarantee or running time.

The following claim simply observes that, for each $i = 0, \ldots, k$, setting $z^i_{P^*_{u_i w_i}} = 1$ and x^i to be the indicator vector of the node-set of $P^*_{u_i w_i}$ yields a feasible solution to (Ap-P).

▷ Claim 3.5. We have $OPT_{\text{Ap-P}} \ge \pi(P^*)$.

Lemma 3.6 proves the main result regarding the polytime solvability of (Ap-P). Its proof is a bit technical, and is deferred to the end of this section.

▶ **Lemma 3.6.** *Using algorithm $\mathcal{A}$, we can obtain in polynomial time a solution $(\overline{x}, \overline{z})$ such that: (i) $\sum_{i=0}^{k}\sum_{v\in V}\pi_v\overline{x}_v^i \geq OPT_{\text{Ap-P}}$, (ii) $\sum_{P\in\mathcal{P}^i:v\in P}\overline{z}_P^i \geq \overline{x}_v^i/\alpha$ for all $v\in V$ and $i=0,\ldots,k$, and (iii) $(\overline{x},\overline{z})$ satisfies the remaining constraints of* (Ap-P).

▶ **Lemma 3.7.** *The paths $Q^0,\ldots,Q^k$ returned if we use subroutine* LP-Round *in step D1.2 satisfy $\mathrm{E}\big[\pi\big(\bigcup_{i=0}^k Q^i\big)\big] \geq \big(1-e^{-1/\alpha}\big)OPT_{\text{Ap-P}}$.*

Proof. Defining $\rho_v := \Pr[v \in (Q^0 \cup \ldots \cup Q^k)]$ for $v \in V$, we have $\mathrm{E}\big[\pi\big(\bigcup_{i=0}^k Q^i\big)\big] = \sum_{v\in V}\pi_v\rho_v$. The paths $Q^0,\ldots,Q^k$ are chosen independently, and $\Pr[v\in Q^i] \geq \overline{x}_v^i/\alpha$ for each $v \in V$ (by Lemma 3.6 (ii)). So for every $v\in V$, we have

$$\rho_v \geq 1-\prod_{i=0}^{k}\Big(1-\tfrac{\overline{x}_v^i}{\alpha}\Big) \geq 1-\Big(1-\frac{\sum_{i=0}^k \overline{x}_v^i/\alpha}{k}\Big)^k \geq \Big[1-\Big(1-\tfrac{1}{\alpha k}\Big)^k\Big]\cdot\sum_{i=0}^{k}\overline{x}_v^i \geq \big(1-e^{-1/\alpha}\big)\sum_{i=0}^{k}\overline{x}_v^i$$

which proves the lemma. The second inequality above follows from the AM-GM inequality, and the third uses the fact that the function $f(x) = 1-\big(1-\frac{x}{k}\big)^k$ is concave, and so $f(x) \geq x\cdot\frac{f(1/\alpha)-f(0)}{1/\alpha}$ for all $x \leq 1/\alpha$. ◀

Finishing up the proof of Theorem 3.1 (b). We first prove the randomized guarantee. The expected value of the solution returned is $\mathrm{E}\big[\max_{P\in\mathcal{Q}}\pi(P)\big]$, which is at least $\max_{P\in\mathcal{Q}}\mathrm{E}\big[\pi(P)\big]$. We lower bound the latter quantity by focusing on the path in $\mathcal{Q}$ returned for the right choice of k and the u_i-w_i nodes.

Fixing this choice, Claim 3.5 and Lemma 3.7 show that $\mathrm{E}\big[\pi\big(\bigcup_{i=0}^k Q^i\big)\big] \geq \big(1-e^{-1/\alpha}\big)\pi(P^*)$. Since the $Z^{(\ell)}$ paths for $\ell=0,1,2$ are feasible (Lemma 3.3) and together cover $\bigcup_{i=0}^k Q^i$, the path $Z^{\max}$ returned in step D1.4 satisfies $\mathrm{E}\big[\pi(Z^{\max})\big] \geq \frac{1-e^{-1/\alpha}}{3}\cdot\pi(P^*)$.

Derandomization. The above guarantee can be easily derandomized. We use randomization only in sampling, for each $i=0,\ldots,k$ independently, a random path Q^i from a polynomial-size distribution of u_i-w_i paths. Let $\mathcal{C}^i$ denote the support of this distribution, and for $P\in\mathcal{C}^i$, let λ_P^i denote the probability of choosing path P. The quantity of interest that determines the performance guarantee is $\Phi_0 = \Phi(\lambda^0,\ldots,\lambda^k) := \mathrm{E}\big[\pi\big(\bigcup_{i=0}^k Q^i\big)\big]$. For $P\in\mathcal{C}^i$, let $\mathbb{1}_P$ be the distribution that chooses P deterministically with probability 1. We show how to deterministically choose the Q^is so that $\Phi(\mathbb{1}_{Q^0},\ldots,\mathbb{1}_{Q^k}) \geq \Phi_0$. We have $\Phi = \sum_{v\in V}\pi_v\rho_v$, where

$$\rho_v = \rho_{0,v} + (1-\rho_{0,v})\rho_{1,v} + \ldots + (1-\rho_{0,v})(1-\rho_{1,v})\cdots(1-\rho_{k-1},v)\rho_{k,v},$$

$$\text{and}\quad \rho_{i,v} = \rho_{i,v}(\lambda^0,\ldots,\lambda^k) = \sum_{P\in\mathcal{C}^i}\lambda_P^i \quad \forall i=0,\ldots,k.$$

It is evident that each ρ_v is linear in λ^i, and therefore Φ is linear in λ^i. Therefore, to derandomize: (1) we choose $Q^0\in\mathcal{C}^0$ so that $\Phi(\mathbb{1}_{Q^0},\lambda^1,\ldots,\lambda^k) \geq \Phi_0$; (2) given that we have chosen $Q^0,\ldots,Q^{i-1}$, we choose $Q^i\in\mathcal{C}^i$ so that $\Phi(\mathbb{1}_{Q^0},\ldots,\mathbb{1}_{Q^i},\lambda^{i+1},\ldots,\lambda^k) \geq \Phi(\mathbb{1}_{Q^0},\ldots,\mathbb{1}_{Q^{i-1}},\lambda^i,\ldots,\lambda^k)$. ◀

Proof of Lemma 3.6. Since (Ap-P) has an exponential number of variables, we consider the dual (Ap-D). The dual has polynomially many constraints corresponding to the polynomially many x_v^i primal variables, and an exponential number of constraints corresponding to the z_P^i primal variables. Using standard ideas (see, e.g., [13]), we show that we can use algorithm $\mathcal{A}$ to approximately separate over these exponentially many constraints, and hence leverage the ellipsoid method to obtain the desired primal solution. Let $\theta^i \geq 0$ and $\mu_v^i \geq 0$ denote respectively the dual variables corresponding to constraints (1), (2). The dual constraints corresponding to the z_P^i variables are:

$$\sum_{v \in P} \mu_v^i \leq \theta^i \qquad \forall i = 0, \dots, k, \ \forall P \in \mathcal{P}^i \tag{P2P}$$

For $b \geq 0$, we let (P2P$_b$) denote (P2P) with the RHS changed to θ^i/b. The effect of this on the primal is that it changes the RHS of (1) to b; let (Ap-P$_b$) denote (Ap-P) with this modified version of (1).

We focus on constraints (P2P) and do not explicitly write down the remaining (polynomially many) dual constraints (including nonnegativity constraints); we collectively denote these constraints by (Ap-D-*). Letting β denote the remaining dual variables, the objective function of (Ap-D) is of the form $\sum_{i=0}^k \theta^i + h^T\beta$, where h is a fixed vector.

Define $\mathcal{K}(\nu; b) := \big\{(\beta, \mu, \theta) : \mu, \theta \geq 0, \ \text{(Ap-D-*)}, \ (\text{P2P}_b), \ \sum_{i=0}^k \theta^i + h^T\beta \leq \nu\big\}$. Note that the optimal value of the dual, and hence (Ap-P), is the smallest ν such that $\mathcal{K}(\nu; 1) \neq \emptyset$. Given ν, (β, μ, θ), we can use $\mathcal{A}$ to either show that $(\beta, \mu, \theta) \in \mathcal{K}(\nu; 1)$, or find a hyperplane separating (β, μ, θ) from $\mathcal{K}(\nu; \alpha)$. We first check if $\mu, \theta \geq 0$, (Ap-D-*) and $\sum_{i=0}^k \theta^i + h^T\beta \leq \nu$ hold, and if not use the violated inequality as the separating hyperplane. Next, for each $i = 0, \dots, k$, we run $\mathcal{A}$ on the P2P-orienteering instance with start and end nodes u_i, w_i respectively, length bound $c_{u_i w_i} + \lceil \gamma^i \rceil - 1$, and node-rewards $\{\mu_v^i\}_{v \in V'}$. If for some i, we obtain a path $P \in \mathcal{P}^i$ with reward greater than θ^i/α, then we return $\sum_{v \in P} \mu_v^i \leq \theta^i/\alpha$ as the separating hyperplane. Otherwise, for all $i = 0, \dots, k$ and all $P \in \mathcal{P}^i$, we know that $\sum_{v \in P} \mu_v^i \leq \theta^i$, and so $(\beta, \mu, \theta) \in \mathcal{K}(\nu; 1)$. Thus, for a fixed ν, in polynomial time, the ellipsoid method either certifies that $\mathcal{K}(\nu; \alpha) = \emptyset$, or returns a point $(\beta, \mu, \theta) \in \mathcal{K}(\nu; 1)$.

It is easy to find an upper bound UB such that $\mathcal{K}(\mathsf{UB}; 1) \neq \emptyset$. For a given $\epsilon > 0$, we use binary search in the range $[0, \mathsf{UB}]$ to find ν^* such that the ellipsoid method when run for ν^* (with the above separation oracle) returns $(\beta^*, \mu^*, \theta^*) \in \mathcal{K}(\nu^*; 1)$, and when run for $\nu^* - \epsilon$ certifies that $\mathcal{K}(\nu^* - \epsilon; \alpha) = \emptyset$. Since $\mathcal{K}(\nu^*; 1) \neq \emptyset$, we have that $OPT_{\text{Ap-P}} \leq \nu^*$, and $\mathcal{K}(\nu^* - \epsilon; \alpha) = \emptyset$ implies that the optimal value of (Ap-P$_\alpha$) is at least $\nu^* - \epsilon$. For $\nu^* - \epsilon$, the inequalities returned by the separation oracle during the execution of the ellipsoid method together with the inequality $\sum_{i=0}^k \theta^i + h^T\beta \leq \nu^* - \epsilon$ yield a polynomial-size certificate for the emptiness of $\mathcal{K}(\nu^* - \epsilon; \alpha)$. By duality (or Farkas' lemma), this implies that if we restrict (Ap-P$_\alpha$) to only use the (polynomially many) z_P^i variables corresponding to the violated inequalities of type (P2P$_\alpha$) returned during the execution of the ellipsoid method, we obtain a polynomial-size feasible solution $(\overline{x}, \hat{z})$ to (Ap-P$_\alpha$) of value at least $\nu^* - \epsilon$. If we take ϵ to be inverse exponential in the input size, this also implies $(\overline{x}, \hat{z})$ has value at least $\nu^* \geq OPT_{\text{Ap-P}}$. Finally, setting $\overline{z} = \hat{z}/\alpha$, we obtain that $(\overline{x}, \overline{z})$ has the desired properties. ◀

▶ Remark 3.8. It is worthwhile to note that *none* of our arguments above rely on the symmetry of the underlying metric, and so *the reduction in Theorem 3.1 also holds in asymmetric metrics*. Given an asymmetric metric $\{c_{u,v}\}_{u,v \in V \cup \{r\}}$, we define regret distances in the same way – $c_u^{\mathsf{reg}}(v, w) := c_{u,v} + c_{v,w} - c_{u,w}$ – and they continue to form an asymmetric metric.

▶ Remark 3.9. In step D1, we only need to let k go up to $\log_\gamma D_{\max}$ (where $D_{\max} := \max_v D_v \leq n\Delta$), and so the running time is $O\big(n^{\log D_{\max}}\big)$. Also, for any integer $c \geq 1$, we can obtain an $O(c)$-approximation in $O\big(c \cdot n^{\frac{\log D_{\max}}{c}}\big)$ time, as follows. We divide the indices $0, 1, \dots, k$ (where $k \leq \log_\gamma D_{\max}$) groups of (roughly) $\frac{k}{c}$ consecutive indices, essentially run our algorithm for each group *separately*, and return the best solution found. To elaborate, for a group $\{a, a+1, \dots, b\}$, we guess the corresponding nodes $u_a, w_a, \dots, u_b, w_b$, and obtain a

u_i-w_i path Q^i for each $i \in \{a, \ldots, b\}$ such that $c^{\mathsf{reg}}_{u_i}(Q^i) \leq \lceil \gamma^i \rceil - 1$, and $(Q^a \cup \ldots \cup Q^b)$ obtains reward $\Omega\big(\pi(P^*_{u_a w_b})\big)$. Since we are considering each group in isolation and do not know the u_j, w_j nodes for $j < a$, we need to define $\mathsf{lb}_{i,v}$ differently now (which then specifies the node-set N^i that Q^i is allowed to visit); we now define $\mathsf{lb}_{i,v} := \sum_{j=0}^{a-1} \gamma^j + \sum_{j=a}^{i-1} \max\{c_{u_j w_j} + c_{w_j u_{j+1}}, c_{u_j u_{j+1}} + \gamma^j\} + c_{u_i v}$. Similar to before, for $\ell \in \{0, 1, 2\}$, we define $Z^{(\ell)}$ as the path with node-sequence r, $\{Q^j\}_{a \leq j \leq b: j-a=\ell \bmod 3}$, and it is not hard to mimic the earlier arguments to infer that each $Z^{(\ell)}$ is a feasible deadline-TSP solution.

Note that taking $c = \log_\gamma D_{\max}$, this shows that the best of the r, Q^i-paths, for $i = 0, \ldots, \log_\gamma D_{\max}$ yields a (very simple) $O(\log D_{\max})$-approximation in polynomial time.

▶ Remark 3.10. There are a couple of ways of improving the efficiency of algorithm LP-Round, while incurring some associated loss of approximation factor. First, as noted earlier, Friggstad and Swamy [14] gave a polynomial-size LP-relaxation for P2P orienteering, and an LP-rounding 6-approximation algorithm for P2P orienteering. Their algorithm explicitly yields the distributional guarantee proved in Lemma 3.6, namely, it returns a (polynomial-support) distribution of P2P-orienteering solutions that visits each node with probability at least $x^i_v/6$, where x^i_v has the same meaning as above (see Section 5). One could replace the exponential-size LP (Ap-P) in algorithm LP-Round with their compact LP, and sample paths from the distribution output by their algorithm. This yields a much more efficient guarantee relative to an LP upper bound, but the approximation guarantee degrades to $\frac{3}{1-e^{-1/6}} \approx 19.542$. (While this is worse than the guarantee obtained using Greedy and the $(2+\epsilon)$-approximation for orienteering [6], the benefit is that this guarantee is with respect to an LP upper bound; also, the orienteering algorithm in [14] is simpler than the one in [6].)

Second, one can replace the use of the ellipsoid method to approximately solve (Ap-P) by the multiplicative-weights method, incurring a small loss in approximation.

4 Extensions

Our techniques can be applied to handle various extensions, including monotone-reward TSP, which we discuss here, and some other extensions that we discuss in Section 4.2.

4.1 Monotone-reward TSP

Recall that in monotone-reward TSP, each node v has a non-increasing reward function $\pi_v : \mathbb{R}_+ \mapsto \mathbb{R}_+$, where we set $\pi_r(x) = 0$ for all x for notational ease. We overload notation, and for a rooted path P, we now *define* $\pi(P) := \sum_{v \in P} \pi_v(c_P(v))$, and call this the reward of P. The goal is to find a simple rooted path that obtains maximum reward. Each $\pi_v(.)$ function is specified via a value oracle, and we treat each call to this oracle as an elementary operation. Recall that we assume that c_{uv} is an integer for all $u, v \in V \cup \{r\}$.

We show that monotone-reward TSP can be reduced to deadline TSP. This reduction incurs a slight loss and increases the size of the instance. We also show that the algorithms and analysis from Section 3 carry over easily to monotone-reward TSP.

▶ **Theorem 4.1.**

(1) *For any $\epsilon > 0$, given a monotone-reward TSP instance $\mathcal{I}$ with n clients, we can obtain in polytime a deadline-TSP instance $\mathcal{I}'$ with $O\big(\frac{n}{\epsilon} \cdot \log \frac{n}{\epsilon}\big)$ clients such that an α-approximation for $\mathcal{I}'$ yields an $\alpha/(1-2\epsilon)$-approximation to $\mathcal{I}$.*

(2) *The algorithms for deadline TSP described in Section 3 can be easily adapted to monotone-reward TSP and yield the same guarantees as those stated in Theorem 3.1.*

Proof of part (1) of Theorem 4.1. Recall that Δ is the diameter of the metric. In any simple rooted path, each client is visited by time $n \cdot \Delta$; so we define $\pi_v(t) = 0$ for all $t > n\Delta$ for notational ease. In the deadline-TSP instance, we use the same metric, but for each node $v \in V$ and every time $t \in \{0, 1, \ldots, n\Delta\}$, we create a co-located client (v, t) with deadline t and reward $\pi_v(t) - \pi_v(t+1)$ (which is nonnegative since $\pi_v(.)$ is non-increasing). Thus, if v is visited at time t in the original graph, then we collect reward $\pi_v(t)$ in total from its co-located clients $(v, t), (v, t+1), \ldots$. This is a lossless, but pseudo-polytime, reduction.

To make this efficient, we apply a geometric bucketing idea on the rewards. Consider a fixed $\epsilon > 0$. Observe that $\mathsf{LB} := \max_{v \in V} \pi_v(c_{rv})$ is the maximum reward that any solution can collect from node v. Also, we can obtain a path with reward at least LB by considering the path r, v for the node v that attains the maximum. We we have $\mathsf{LB} \leq \mathsf{opt}_{\mathcal{I}} \leq n \cdot \mathsf{LB}$, where $\mathsf{opt}_{\mathcal{I}}$ is the optimal value for the monotone-reward TSP instance $\mathcal{I}$. Now for each $v \in V$, instead of creating $n \cdot \Delta$ co-located clients, we consider each integer $i \geq 0$ such that $(1-\epsilon)^i \geq \frac{\epsilon}{n}$; note that there are at most $O\big(\frac{1}{\epsilon} \cdot \log \frac{n}{\epsilon}\big)$ such values. For each such i, we use binary search (using the value oracle) to find the largest integer $t_{v,i}$ such that $\pi_v(t_{v,i}) \geq (1-\epsilon)^i \cdot \mathsf{LB}$. In the deadline TSP instance $\mathcal{I}'$, we create a client located at v with deadline $t_{v,i}$ and reward $(1-\epsilon)^i \cdot \mathsf{LB} - (1-\epsilon)^{i+1} \cdot \mathsf{LB}$. So, if a path reaches location v by time $t_{v,i}$ it will collect reward at least $(1-\epsilon)^i \cdot \mathsf{LB} - \frac{\epsilon}{n} \cdot \mathsf{LB}$.

Let P^* be an optimum monotone-reward TSP solution. Consider the value of P^* as a solution to the new deadline-TSP instance. Consider each v on P^*, and say v was visited at time t along P^*. We know that $\pi_v(t) \leq \mathsf{LB}$. Let i be the smallest integer such that $(1-\epsilon)^i \cdot \mathsf{LB} \leq \pi_v(t)$; so we also have $(1-\epsilon)^i \cdot \mathsf{LB} \geq (1-\epsilon)\pi_v(t)$. By the construction of $\mathcal{I}'$, we know that $t_{v,i} \geq t$, and so we collect total reward at least $(1-\epsilon)^i \cdot \mathsf{LB} - \frac{\epsilon}{n} \cdot \mathsf{LB} \geq (1-\epsilon)\pi_v(t) - \frac{\epsilon}{n} \cdot \mathsf{LB}$ from the clients co-located at v. So the total reward of P^* when viewing it as a deadline-TSP solution is at least $(1-\epsilon) \cdot \mathsf{opt}_{\mathcal{I}} - \epsilon \cdot \mathsf{LB} \geq (1-2\epsilon) \cdot \mathsf{opt}_{\mathcal{I}}$.

Conversely, any deadline-TSP solution P' when viewed as a monotone-reward TSP solution produces at least as much reward, since for any time t and any client v, $\pi_v(t)$ is at least the total reward that will be collected in instance $\mathcal{I}'$ by visiting the clients located at v at time t. Therefore, a solution to instance $\mathcal{I}'$ of value at least $\mathsf{opt}_{\mathcal{I}'}/\alpha$ yields a monotone-reward TSP solution with value at least $\frac{1-2\epsilon}{\alpha} \cdot \mathsf{opt}_{\mathcal{I}}$. ◀

Proof of part (2) of Theorem 4.1

We briefly sketch the changes to the algorithms and analyses from Section 3. The only changes to Algorithm 1 involve changes to the constituent subroutines Greedy and LP-Round, and entail figuring out what *fixed* node rewards to use when we consider a P2P-orienteering instance in step G1, or in LP (Ap-P). (As before, P^* denotes an optimal solution, and the right choice of u_i, w_i nodes continues to be as defined in Section 3; consequently Lemma 3.2 continues to hold. Also, steps D1.4 and D2 hold as is, given our modified definition of the reward of a path.)

Two observations guide the choice of rewards we use in step G1 and in (Ap-P). First, we know that the visiting time of a node $v \in P^*_{u_i w_i}$ is at least $\mathsf{lb}_{i,v}$. Second, suppose for each i, we obtain a u_i-w_i path Q^i with $c^{\mathsf{reg}}_{u_i}(Q^i) \leq \lceil \gamma^i \rceil - 1$ and construct the paths $\{Z^{(0)}, Z^{(1)}, Z^{(2)}\}$ as in step D1.3; then, by Lemma 3.3, for every $Z^{(\ell)}$, every $Q^j \subseteq Z^{(\ell)}$, and every $v \in Q^j$, we know that the visiting time of this occurrence of v is at most $\mathsf{lb}_{i,v}$. Consequently, in subroutine Greedy and subroutine LP-Round, we use the node rewards $\{\pi_v(\mathsf{lb}_{i,v})\}_{v \in V'}$ for the i-th orienteering instance; also, the node-set N^i is now simply V. Let (MRAp-P) denote the resulting analogue of (Ap-P), where the coefficient multiplying x^i_v in the objective is now $\pi_v(\mathsf{lb}_{i,v})$.

Essentially, all the statements proved for deadline TSP hold here as well, with minor notational tweaks, and for mostly the same reasons. (Recall that for a rooted path P, we now define $\pi(P) := \sum_{v\in P}\pi_v(c_P(v))$, and call this the reward of P.)

Algorithm Greedy is unchanged, and the guarantee of Lemma 3.4 still holds, but it needs to be stated more precisely, and its proof needs to be tweaked. The modified statement is that we have $\sum_{i=0}^{k}\sum_{v\in Q^i\setminus(Q^0\cup\ldots\cup Q^{i-1})}\pi_v(\mathsf{lb}_{i,v}) \geq \mathsf{opt}/(\alpha+1)$. In the proof, in place of (5), for all $i=0,\ldots,k$, we now have

$$\sum_{v\in Q^i\setminus(\bigcup_{j=0}^{i-1}Q^j)}\pi_v(\mathsf{lb}_{i,v}) \geq \frac{1}{\alpha}\cdot\sum_{\substack{v\in P^*_{u_iw_i}\\ v\notin\bigcup_{j=0}^{i-1}Q^j}}\pi_v(\mathsf{lb}_{i,v}) \geq \frac{1}{\alpha}\cdot\Bigg[\sum_{v\in P^*_{u_iw_i}}\pi_v\big(c_{P^*}(v)\big) - \sum_{\substack{v\in P^*_{u_iw_i}\\ v\in\bigcup_{j=0}^{i-1}Q^j}}\pi_v(\mathsf{lb}_{i,v})\Bigg] \quad (6)$$

Consider adding, for all $i=0,\ldots,k$, the inequality in (6) involving the LHS of (6) and the final RHS of (6). The LHS of the resulting inequality is $\sum_{i=0}^{k}\sum_{v\in Q^i\setminus(Q^0\cup\ldots\cup Q^{i-1})}\pi_v(\mathsf{lb}_{i,v})$. For a node v, consider its total contribution (across all i) to the negative terms on the final RHS of (6). Node v gets counted at most once among all these negative terms, and if i is the smallest index such that $v\in Q^i$, then the negative term where v is counted is due to some index $j>i$, and so is at least $-\pi_v(\mathsf{lb}_{i,v})$. Therefore, the total contribution of the negative terms on the RHS is at least $-\frac{1}{\alpha}\cdot\sum_{i=0}^{k}\sum_{v\in Q^i\setminus(Q^0\cup\ldots\cup Q^{i-1})}\pi_v(\mathsf{lb}_{i,v})$. Thus, adding (6) for all $i=0,\ldots,k$ and simplifying yields the inequality $\sum_{i=0}^{k}\sum_{v\in Q^i\setminus(Q^0\cup\ldots\cup Q^{i-1})}\pi_v(\mathsf{lb}_{i,v}) \geq \mathsf{opt}/(\alpha+1)$.

Subroutine LP-Round is also unchanged, except for the change in (Ap-P) due to the above-mentioned node rewards and since $N^i=V$. The new LP (MRAp-P) can be approximately solved as in Lemma 3.6. Let $\overline{x}$ denote the values of the x_i^v variables in the solution so obtained. Then we have $\overline{z}$ such that the objective value of $(\overline{x},\overline{z})$ is at least $OPT_{(\text{MRAp-P})}$, $\sum_{P\in\mathcal{P}^i:v\in P}\overline{z}_P^i \geq \overline{x}_v^i/\alpha$ for all $v\in V$ and $i=2,\ldots,k$, and $(\overline{x},\overline{z})$ satisfies the remaining constraints of (MRAp-P).

Claim 3.5 gets replaced by

$$\sum_{i=0}^{k}\sum_{v\in V}\pi_v(\mathsf{lb}_{i,v})\overline{x}_v^i \geq \sum_{i=0}^{k}\sum_{v\in P^*_{u_iw_i}}\pi_v(\mathsf{lb}_{i,v}) \geq \sum_{v\in P^*}\pi_v(c_{P^*}(v)) = \mathsf{opt} \quad (7)$$

The guarantee of Lemma 3.7 still holds, but it needs to be stated more precisely and requires a different proof, which we defer to the end of this section.

▶ **Lemma 4.2.** *For each $i=0,\ldots,k$, let Q^i be a random path obtained such that* $\Pr[v\in Q^i]\geq\overline{x}_v^i/\alpha$ *for all $v\in V$, for some $\alpha\geq 1$. Then*

$$\mathrm{E}\Bigg[\sum_{i=0}^{k}\sum_{v\in Q^i\setminus(Q^2\cup\ldots\cup Q^{i-1})}\pi_v(\mathsf{lb}_{i,v})\Bigg] \geq \big(1-e^{-1/\alpha}\big)\cdot\Bigg(\sum_{i=0}^{k}\sum_{v\in V}\pi_v(\mathsf{lb}_{i,v})\overline{x}_v^i\Bigg).$$

To finish up the analysis, as before we lower bound $\max_{\ell=0,1,2}\mathrm{E}\big[\pi(Z^{(\ell)})\big]$. Since Lemma 3.3 continues to hold, for every node v, we know that if i is the smallest index such that $v\in Q^i$, then v is visited by some path $Z^{(\ell)}$ by time $\mathsf{lb}_{i,v}$. Therefore, $\max_{\ell=0,1,2}\mathrm{E}\big[\pi(Z^{(\ell)})\big] \geq \frac{1}{3}\cdot\mathrm{E}\big[\sum_{i=0}^{k}\sum_{v\in Q^i\setminus(Q^2\cup\ldots\cup Q^{i-1})}\pi_v(\mathsf{lb}_{i,v})\big]$. Combining this with Lemma 4.2 and (7), we obtain the same approximation guarantee as in part (b) of Theorem 3.1.

Finally, the derandomization also proceeds as before. Let $\mathcal{C}^i$ denote the polynomial-size support of the distribution from which the u_i-w_i path Q^i is sampled, and let λ_P^i denote the probability of choosing a path $P\in\mathcal{C}^i$. We are now interested in the quantity $\Phi_0=$

$\Phi(\lambda^0,\ldots,\lambda^k) := \mathrm{E}\big[\sum_{i=0}^k \sum_{v\in Q^i\setminus(Q^2\cup\ldots\cup Q^{i-1})} \pi_v(\mathsf{lb}_{i,v})\big]$. We now have $\Phi = \sum_{v\in V} \mathsf{rewd}_v$, where

$$\mathsf{rewd}_v = \rho_{0,v}\pi_v(\mathsf{lb}_{0,v}) + (1-\rho_{0,v})\rho_{1,v}\pi_v(\mathsf{lb}_{1,v}) + \ldots + \Big(\prod_{i=0}^k (1-\rho_{i,v})\Big)\rho_{k,v}\pi_v(\mathsf{lb}_{k,v}), \text{ and}$$

$$\rho_{i,v} = \rho_{i,v}(\lambda^0,\ldots,\lambda^k) = \sum_{P\in\mathcal{C}^i} \lambda^i_P \quad \forall i = 0,\ldots,k.$$

Since each rewd_v is linear in λ^i for all i, as before, we can deterministically choose Q^is for all i so that $\Phi(\mathbb{1}_{Q^0},\ldots,\mathbb{1}_{Q^k}) \geq \Phi_0$.

Proof of Lemma 4.2. The following claim will be useful.

▷ Claim 4.3. Let $a_1 \geq a_2 \geq \ldots \geq a_q \geq a_{q+1} := 0$. Let $y \in [0,1]^q$, and $t \geq \sum_{i=1}^q y_i$. Define $F(y_1,\ldots,y_q) := y_1a_1 + (1-y_1)y_2a_2 + \ldots + (1-y_1)(1-y_2)\cdots(1-y_{q-1})y_qa_q$. We have: (i) $F(y) \geq (1-e^{-t})\cdot\frac{\sum_{i=1}^q a_iy_i}{t}$; and (ii) $F(z) \leq F(y)$ for any z such that $0 \leq z \leq y$.

Proof. The proof of part (i) follows from elementary arguments (see, e.g., [11]). For any $i = 1,\ldots,q$, we have

$$\begin{aligned} y_1 + (1-y_1)y_2 + \ldots + (1-y_1)(1-y_2)\cdots(1-y_{i-1})y_i &= \Big(1-(1-y_1)(1-y_2)\cdots(1-y_i)\Big) \\ &\geq 1-\Big(1-\frac{\sum_{j=1}^i y_j}{i}\Big)^i \geq \Big[1-\Big(1-\tfrac{t}{i}\Big)^i\Big]\cdot\frac{\sum_{j=1}^i y_j}{t} \geq \big(1-e^{-t}\big)\cdot\frac{\sum_{j=1}^i y_j}{t} \end{aligned}$$

The first inequality is the AM-GM inequality, the second uses the fact that the function $f(x) = 1-\big(1-\frac{x}{i}\big)^i$ is concave, and so $f(x) \geq x\cdot\frac{f(t)-f(0)}{t}$ for $x \leq t$. That is, we have

$$y_1 + (1-y_1)y_2 + \ldots + (1-y_1)(1-y_2)\cdots(1-y_{i-1})y_i \geq \big(1-e^{-t}\big)\cdot\frac{\sum_{j=1}^i y_j}{t} \tag{8}$$

Multiplying (8) by $a_i - a_{i+1}$ and adding the resulting inequalities for $i = 1,\ldots,q$ yields the stated bound.

Part (ii) follows from the fact that

$$\frac{\partial F}{\partial y_i} = (1-y_1)\cdots(1-y_{i-1})\Big[a_i - \Big(a_{i+1}y_{i+1} + (1-y_{i+1})y_{i+2}a_{i+2} + \ldots + (1-y_{i+1})\cdots(1-y_{q-1})y_qa_q\Big)\Big]$$

which is nonnegative since $a_i \geq a_{i+1},\ldots,a_q$. ◁

We now have everything to prove Lemma 4.2. Let $\rho^i_v := \Pr[v \in Q^i]$ for $v \in V$, and $i = 0,\ldots,k$. Let Φ denote the quantity on the LHS of the inequality in the lemma. Since $Q^0,\ldots,Q^k$ are chosen independently, we have $\Phi = \sum_{v\in V}\mathsf{rewd}_v$, where

$$\begin{aligned} \mathsf{rewd}_v &= \rho_{0,v}\pi_v(\mathsf{lb}_{0,v}) + (1-\rho_{0,v})\rho_{1,v}\pi_v(\mathsf{lb}_{1,v}) + \ldots + \Big(\prod_{i=0}^k(1-\rho_{i,v})\Big)\rho_{k,v}\pi_v(\mathsf{lb}_{k,v}) \\ &\geq \frac{\overline{x}^0_v}{\alpha}\cdot\pi_v(\mathsf{lb}_{0,v}) + \Big(1-\frac{\overline{x}^0_v}{\alpha}\Big)\frac{\overline{x}^1_v}{\alpha}\cdot\pi_v(\mathsf{lb}_{1,v}) + \ldots + \Big(\prod_{i=0}^k\Big(1-\frac{\overline{x}^i_v}{\alpha}\Big)\Big)\frac{\overline{x}^k_v}{\alpha}\cdot\pi_v(\mathsf{lb}_{k,v}). \end{aligned}$$

The inequality follows from part (ii) of Claim 4.3 since $\rho^i_v \geq \overline{x}^i_v/\alpha$ for all $i = 0,\ldots,k$. Applying Claim 4.3 with $a_i = \pi_v(\mathsf{lb}_{i,v})$ and $y_i = \frac{\overline{x}^i_v}{\alpha}$ for all $i = 0,\ldots,k$, and $t = \frac{1}{\alpha}$, we therefore obtain that $\mathsf{rewd}_v \geq \big(1-e^{-1/\alpha}\big)\sum_{i=0}^k \pi_v(\mathsf{lb}_{i,v})\overline{x}^i_v$. It follows that $\Phi \geq \big(1-e^{-1/\alpha}\big)\cdot\big(\sum_{i=0}^k\sum_{v\in V}\pi_v(\mathsf{lb}_{i,v})\overline{x}^i_v\big)$. ◀

4.2 Further extensions of Deadline TSP

Point-to-point {deadline, monotone-reward} TSP. In the point-to-point (P2P) version of {deadline, monotone-reward} TSP, in addition to the root node r, we are also given an end-node t and a length-bound B. The goal is to find an r-t path of length at most B that collects maximum reward.

The P2P-version easily reduces to the standard version of the problem. For deadline TSP, we can incorporate the above requirements by modifying the deadline of each node $v \in V'$ to $D_v^{\mathsf{new}} := \min\{D_v, B - c_{vt}\}$. If P is a rooted path ending at a node s such that all $v \in P$ are visited by time D_v^{new}, then $P' = P, st$ is an r-t path such that all $v \in P'$ are visited by time D_v, and $c(P') \leq D_s^{\mathsf{new}} + c_{st} \leq B$.

Similarly, for monotone-reward TSP, we modify the reward function of each $v \in V'$ to $\pi_v^{\mathsf{new}}(x) = \pi_v(x)$ if $x \leq B - c_{vt}$, and 0 if $x > B - c_{vt}$. As before if P is an r-s path earning a certain modified reward, then $P' = P, st$ is an r-t path of length at most B earning the same reward.

TSP with release dates, and TSP with increasing rewards. In TSP with release dates, each node v has a release date rel_v instead of a deadline, and we have a length bound B. A feasible solution is a rooted path P, and a traversal of P starting from the root node where we are allowed to also wait at nodes, so that the visiting time of each node $v \in P$ is at least rel_v and at most B; we seek a feasible solution that gathers maximum reward. Set $\mathsf{rel}_r = 0$ for notational convenience.

As noted in [2], this can be reduced to deadline TSP as follows. If P is a feasible solution ending at node t, then we must have $\mathsf{rel}_t \leq B$ and $c(P_{vt}) \leq B - \mathsf{rel}_v$ for all $v \in P$. Conversely any r-t path P with $c(P_{vt}) \leq B - \mathsf{rel}_v$ for all $v \in P$ yields a feasible solution, where we wait at r for $B - c(P)$ time and then traverse P without any subsequent waiting. We thus infer that we seek a feasible solution to P2P-deadline TSP (and hence deadline TSP) with start node t, end-node r, length bound B, and deadlines $\{B - \mathsf{rel}_v\}_{v \in V'}$. Trying all possible choices of t completes the reduction.

Analogous to how monotone-reward TSP generalizes deadline TSP, we can consider a generalization of TSP with release dates wherein each node v has a *non-decreasing* reward function $\pi_v : \mathbb{R}_+ \mapsto \mathbb{R}_+$, and we seek a rooted path P with $c(P) \leq B$, and a traversal of P that yields maximum reward. As above, if we know the end-node t of an optimal solution, then this reduces to solving an instance of P2P monotone-reward TSP, where we seek a path starting at t and ending at r path of length at most B, and the reward of node v is given by the *non-increasing* function $\pi'_v(x) = \pi_v(B - x)$.

Orienteering with time windows. Chekuri et al. [6] obtain (among other results) an $O\big(\max\{\log \mathsf{opt}, \log \frac{L_{\max}}{L_{\min}}\}\big)$-approximation for orienteering with time windows, where opt is the optimal value, and $L_{\max}$ and $L_{\min}$ are the lengths of the longest and shortest time windows with non-zero length. The $\log \mathsf{opt}$ term in their guarantee is because they use the logarithmic approximation of [2] for deadline TSP.[3] Replacing this algorithm with our deadline-TSP algorithm therefore yields an $O\big(\log \frac{L_{\max}}{L_{\min}}\big)$-approximation for orienteering with time windows in $O\big(n^{\log n\Delta}\big)$ time.

[3] Chekuri et al. [6] work with $\{0,1\}$-rewards, and in this setting, the approximation factor obtained in [2] for deadline TSP can be seen to be $O(\log \mathsf{opt})$.

5 LP-rounding algorithm for P2P-orienteering in [14]

We briefly discuss the LP-rounding algorithm for P2P-orienteering by Friggstad and Swamy [14] that directly yields the distributional guarantee utilized in subroutine LP-Round. Recall that $(V' = \{r\} \cup V, c)$ is the underlying metric, and our P2P-orienteering instance $\mathcal{I}$ is specified by node-set $N \subseteq V'$, start and end-nodes $s, t \in N$ respectively, length-bound B, and node-rewards $\{\pi_v\}_{v \in N}$.

Let $D = (N, A)$ denote the complete (bidirected) graph on N, where the cost of an arc $(u, v) \in A$ is set to c_{uv}; thus c induces a metric on D. Let P^* be an optimal solution to $\mathcal{I}$. The idea underlying the LP relaxation is to "guess" the node $v \in P^*$ that maximizes $c_{sv} + c_{vt}$. The LP then searches for an $s \rightsquigarrow v$ path and a $v \rightsquigarrow t$ path – encoded by requiring an s-v flow and v-t flow of value 1 – that visit only nodes $u \in N$ such that $c_{su} + c_{ut} \le c_{sv} + c_{vt}$, have total length at most B, and together earn the maximum reward. Also, we replace the "guessing" step by having indicator variables z^v_v to denote if v is the node on P^* with maximum $c_{sv} + c_{vt}$ value.

This leads to the following LP. For every $v \in N$, we have the following set of variables (in addition to z^v_v). We use $x_v \in [0, 1]$ to denote the extent to which v is visited. We let y^{sv} denote an s-v flow of value z^v_v, and y^{vt} denotes a v-t flow of value x^v_v. We impose that $y^{sv}\big(\delta^{\text{in}}(u)\big) = y^{vt}\big(\delta^{\text{in}}(u)\big) = 0$ whenever $c_{su} + c_{ut} > c_{sv} + c_{vt}$. We use z^{sv}_u and z^{vt}_u to denote respectively the $s \rightsquigarrow u$ connectivity under x^{sv} and the $v \rightsquigarrow u$ connectivity under x^{vt}. So in an integral solution, z^{sv}_u and z^{vt}_u indicate respectively if u lies on the s-v portion or the v-t portion of the optimum path. We connect the x and z variables by imposing that $x_u = \sum_{v \in N}(z^{sv}_u + z^{vt}_u)$ for every $u \in N$. For nodes $v, p, q \in N$, and $\kappa \ge 0$, define

$$\mathcal{F}_v(p, q, \kappa) := \Big\{ y \in \mathbb{R}^A_+ : \quad y\big(\delta^{\text{out}}(p)\big) = \kappa = y\big(\delta^{\text{in}}(q)\big), \qquad y\big(\delta^{\text{in}}(p)\big) = 0 = y\big(\delta^{\text{out}}(q)\big)$$
$$y\big(\delta^{\text{in}}(w)\big) - y\big(\delta^{\text{out}}(w)\big) = 0 \qquad \forall w \in V' \setminus \{p, q\}$$
$$y\big(\delta^{\text{in}}(w)\big) = 0 \qquad \forall w \in V' : c_{sw} + c_{wt} > c_{sv} + c_{vt} \Big\}$$

Note that if $\kappa > 0$, then $\mathcal{F}(u, u, \kappa) = \emptyset$ for every u.

$$\max \quad \sum_{u, v \in N} \pi_u x_u \tag{P2P-O}$$
$$\text{s.t.} \quad y^{sv} \in \mathcal{F}_v(s, v, z^v_v),\ y^{vt} \in \mathcal{F}_v(v, t, z^v_v) \qquad \forall v \in N$$
$$y^{sv}\big(\delta^{\text{in}}(S)\big) \ge z^{sv}_u \qquad \forall v \in N, S \subseteq N \setminus \{s\}, u \in S \tag{9}$$
$$y^{vt}\big(\delta^{\text{in}}(S)\big) \ge z^{vt}_u \qquad \forall v \in N, S \subseteq N \setminus \{v\}, u \in S \tag{10}$$
$$\sum_{a \in A} c_a(y^{sv}_a + y^{vt}_a) \le B z^v_v \qquad \forall v \in N \tag{11}$$
$$\sum_{v \in N}(z^{sv}_u + z^{vt}_u) = x_u \qquad \forall u \in N \tag{12}$$
$$\sum_{v \in N} z^v_v = 1, \qquad y, z \ge 0, \quad x \in [0, 1]^N.$$

As noted in [14], we can rephrase the cut constraints (9), (10) using additional flow variables and constraints to obtain a polynomial-size formulation.

Friggstad and Swamy [14] devise the following algorithm for rounding an LP solution (x, y, z). They show that for each $v \in N$, we can utilize y^{sv} to obtain a polynomial collection of s-rooted paths, and weights for these paths such that:

(i) if a path P in the collection ends at a node u, then $P' = P, ut$ is an s-t path with $c(P') \leq B$;

(ii) the total weight of paths in the collection is at most $3z_v^v$; and

(iii) the total weight of paths containing a node u is at least z_u^{sv}, for all $u \in N$.

Similarly, we can utilize y^{vt} to obtain a suitable weighted collection of paths, each of which yields an s-t path of length at most B. Taking these collections for all $v \in N$, we obtain that the total weight of paths is at most 6, and for each $u \in N$, the total weight of paths containing u is at least $\sum_v (z_u^{sv} + z_u^{vt}) = x_u$. The distribution obtained by scaling the weights by 6 yields the desired distribution.

References

1 H. C. An, R. Kleinberg, and D. B. Shmoys. Improving Christofides' algorithm for the s-t path TSP. *Journal of the ACM*, 62(5):34, 2015.

2 N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-TSP and vehicle routing with time windows. In *Proceedings of 36th STOC*, pages 166–174, 2004.

3 A. Blum, S. Chawla, D. R. Karger, T. Lane, and A. Meyerson. Approximation algorithms for orienteering and discount-reward TSP. *SIAM J. Comput.*, 37(2):653–670, 2007.

4 R. Carr and S. Vempala. Randomized metarounding. *Random Structures and Algorithms*, 20(3):343–352, 2002.

5 K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees, and minimum latency tours. In *Proceedings of 44th FOCS*, pages 36–45, 2003.

6 C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms*, 8(3), 2012.

7 C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *Proceedings of 7th APPROX*, pages 72–83, 2004.

8 C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. In *Proceedings of 46th FOCS*, pages 245–253, 2005.

9 N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, CMU, 1976.

10 B. Farbstein and A. Levin. Discount reward TSP. *Algorithmica*, 80(2):472–495, 2018.

11 L. Fleischer, M. Goemans, V. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of 17th SODA*, pages 611–620, 2006.

12 Z. Friggstad, M. R. Salavatipour, and Z. Svitkina. Asymmetric traveling salesman path and directed latency problems. *SIAM J. Comput*, 42(4):1596–1619, 2013.

13 Z. Friggstad and C. Swamy. Approximation algorithms for regret-bounded vehicle routing and applications to distance-constrained vehicle routing. In *Proceedings of 46th STOC*, pages 744–753, 2014. Detailed version posted on *CS arXiv*, Nov 2013.

14 Z. Friggstad and C. Swamy. Compact, provably-good LPs for orienteering and regret-bounded vehicle routing. In *Proceedings of 19th IPCO*, pages 199–211, 2017.

15 B. L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics*, 34:307–318, 1987.

16 E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of 35th STOC*, pages 585–594, 2003.

17 A. Köhne, V. Traub, , and J. Vygen. The asymmetric traveling salesman path LP has constant integrality ratio. In *Proceedings of 20th IPCO*, pages 288–298, 2019.

18 R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. *Journal of the ACM*, 58(6), 2011.

19 C.-L. Li, D. Simchi-Levi, and M. Desrochers. On the distance constrained vehicle routing problem. *Operations research*, 40:790–799, 1992.

20 V. Nagarajan and R. Ravi. The directed minimum latency problem. In *Proceedings of 11th APPROX*, pages 193–206, 2008.

21 V. Nagarajan and R. Ravi. The directed orienteering problem. *Algorithmica*, 60(4):1017–1030, 2011.

22 V. Nagarajan and R. Ravi. Approximation algorithms for distance constrained vehicle routing problems. *Networks*, 59(2):209–214, 2012.

23 I. Post and C. Swamy. Linear-programming based techniques for multi-vehicle minimum latency problems. In *Proceedings of 26th SODA*, pages 512–531, 2015.

24 A. Sebo and A. van Zuylen. The salesman's improved paths: A $3/2 + 1/34$ approximation. In *Proceedings of 57th FOCS*, pages 118–127, 2016.

25 A. Sebo and J. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.

26 O. Svensson, J. Tarnawski, and L. Vegh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of 50th STOC*, pages 204–213, 2018.

27 P. Toth and eds D. Vigo. *The Vehicle Routing Problem.* SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.

28 V. Traub and J. Vygen. Approaching $\frac{3}{2}$ for the s-t path TSP. In *Proceedings of 29th SODA*, pages 1854–1864, 2018.

29 R. Zenklusen. A 1.5-approximation for path TSP. In *Proceedings of 30th SODA*, pages 1539–1549, 2019.