Approximation Algorithms for Regret-Bounded Vehicle Routing and Applications to Distance-Constrained Vehicle Routing*

Zachary Friggstad†
Dept. of Computing Science
Univ. Alberta, Edmonton, AB T6G 2E8
zacharyf@cs.ualberta.ca

Chaitanya Swamy‡
Dept. of Combinatorics and Optimization
Univ. Waterloo, Waterloo, ON N2L 3G1
cswamy@math.uwaterloo.ca

ABSTRACT

We consider vehicle-routing problems (VRPs) that incorporate the notion of *regret* of a client, which is a measure of the waiting time of a client relative to its shortest-path distance from the depot. Formally, we consider both the additive and multiplicative versions of, what we call, the *regret-bounded vehicle routing problem* (RVRP). In these problems, we are given an undirected complete graph $G = (\{r\} \cup V, E)$ on n nodes with a distinguished root (depot) node r, edge costs $\{c_{uv}\}$ that form a metric, and a regret bound R. Given a path P rooted at r and a node $v \in P$, let $c_P(v)$ be the distance from r to v along P. The goal is to find the fewest number of paths rooted at r that cover all the nodes so that for every node v covered by (say) path P: (i) its additive regret $c_P(v) - c_{rv}$, with respect to P is at most R in *additive-RVRP*; or (ii) its multiplicative regret, $c_P(v)/c_{rv}$, with respect to P is at most R in *multiplicative-RVRP*.

Our main result is the *first* constant-factor approximation algorithm for additive-RVRP. This is a substantial improvement over the previous-best $O(\log n)$-approximation. Additive-RVRP turns out be a rather central vehicle-routing problem, whose study reveals insights into a variety of other regret-related problems as well as the classical *distance-constrained VRP* (DVRP), enabling us to obtain guarantees for these various problems by leveraging our algorithm for additive-RVRP and the underlying techniques. We obtain approximation ratios of $O\big(\log(\frac{R}{R-1})\big)$ for multiplicative-RVRP, and $O\big(\min\{OPT, \frac{\log D}{\log\log D}\}\big)$ for DVRP with distance bound D via reductions to additive-RVRP; the latter improves upon the previous-best approximation for DVRP.

A noteworthy aspect of our results is that they are obtained by devising rounding techniques for a natural *configuration-style LP*. This furthers our understanding of LP-relaxations for VRPs and enriches the toolkit of techniques that have been utilized for configuration LPs.

*A full version is available from the CS arXiv.

†Work done while the author was a postdoctoral scholar at the University of Waterloo and supported in part by the second author's grants.

‡Supported by NSERC grant 327620-09, an NSERC Discovery Accelerator Supplement Award, and an Ontario Early Researcher Award.

STOC '14, May 31 - June 03 2014, New York, NY, USA.

ACM ACM 978-1-4503-2710-7/14/05 ...$15.00.
http://dx.doi.org/10.1145/2591796.2591840.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2 [**Discrete Mathematics**]

Keywords

Approximation algorithms; Vehicle routing; Configuration LPs; LP rounding; Distance-constrained vehicle routing

1. INTRODUCTION

Vehicle-routing problems (VRPs) constitute a broad class of combinatorial-optimization problems that find a wide range of applications and have been widely studied in the Operations Research and Computer Science communities (see, e.g., [15, 19, 27, 4, 2, 20, 7] and the references therein). These problems are typically described as follows. There are one or more vehicles that start at some depot and provide service to an underlying set of of clients, and the goal is to design routes for the vehicles that visit the clients as quickly as possible. The most common way of formalizing the objective of minimizing client delays is to seek a route of minimum length, or equivalently, a route that minimizes the maximum client delay, which gives rise to (the path variant of) the celebrated traveling salesman problem (TSP). However, this objective does not differentiate between clients located at different distances from the depot, and a client closer to the depot may end up incurring a larger delay than a client that is further away, which can be considered a source of unfairness and hence, client dissatisfaction. Adopting a client-centric approach, we consider an alternate objective that addresses this unfairness and seeks to design routes that promote customer satisfaction.

Noting that the delay of a client is inevitably at least the shortest-path distance from the depot to the client location, following [25, 22], we seek to ensure that the *regret* of a client, which is a measure of its waiting time *relative to its shortest-path distance from the depot*, is bounded. More precisely, we

Symmetric metrics					Asymmetric metrics	
RVRP	kRVRP	Multiplicative-RVRP	Multiplicative-kRVRP	DVRP	RVRP	kRVRP
31	$O(k^2)$	$O\big(\log(\frac{R}{R-1})\big)$	$O(1)$	$O\big(\frac{\log D}{\log\log D}\big)$	$O(\log n)$	$O(k^2 \log n)$

Table 1: Summary of our results. Our main result, for RVRP, yields guarantees for other symmetric-metric problems.

consider the following genre of vehicle-routing problems. We are given an undirected complete graph $G = (\{r\} \cup V, E)$ on n nodes with a distinguished root (depot) node r, and metric edge costs or distances $\{c_{uv}\}$. Given a path P rooted at r and a node $v \in P$, let $c_P(v)$ be the distance from r to v along P (i.e., the length of the r-v subpath of P). There are two natural ways of comparing $c_P(v)$ and c_{rv} to define the regret of a node v on path P. We define the *additive regret* of v with respect to P to be $c_P(v) - c_{rv}$,[1] and the *multiplicative regret* of v with respect to P to be $c_P(v)/c_{rv}$. We are also given a regret bound $R \geq 0$. Fixing a regret measure, a feasible solution is a collection of paths rooted at r that cover all the nodes in G such that the regret of every node with respect to the path covering it is at most R. Thus, a feasible solution to: (i) the additive-regret problem yields the satisfaction guarantee that every client v is visited by time $c_{rv} + R$; and (ii) the multiplicative-regret problem ensures that every client v is visited by time $c_{rv} \cdot R$. The goal is to find a feasible solution that uses the fewest number of paths. We refer to these two problems as *additive-regret-bounded VRP* (additive-RVRP) and *multiplicative-regret-bounded VRP* (multiplicative-RVRP) respectively.

Additive-RVRP has been sometimes referred to as the *school-bus problem* in the literature [25, 22, 5]. However, this term is used to refer to an umbrella of vehicle-routing problems, some of which do not involve regret, so we use the more descriptive name of additive-RVRP. Both versions of RVRP are *APX*-hard via a simple reduction from TSP (Theorem 7.1), so we focus on approximation algorithms.

Our results. We undertake a systematic study of regret-related vehicle-routing problems from the perspective of approximation algorithms. As we illustrate below, additive-RVRP turns out to be the more fundamental of the above two problems and a rather useful problem to investigate, and our study yields insights and techniques that can be applied, often in a black-box fashion, to derive algorithms for various vehicle-routing problems that include both regret-related problems as well as classical problems such as distance-constrained vehicle routing. We therefore focus on additive regret; unless otherwise stated, regret refers to additive regret, and a regret-related problem refers to the problem under the additive-regret measure.

Our main result is the *first constant-factor* approximation algorithm for (additive) RVRP (Theorem 3.2). This is a substantial improvement over the previous-best $O(\log n)$-approximation ratio for RVRP obtained in [5] via the standard set-cover greedy algorithm and analysis.

A noteworthy aspect of our result is that we develop linear-programming (LP) based techniques for the problem. While LP-relaxations have been exploited with striking success in the design and analysis of approximation algorithms, our understanding of LP-relaxations for VRPs is quite limited, and this has been a stumbling block in the design of approximation algorithms for many of these problems. Notably, we develop LP-rounding techniques for a natural *configuration-style LP-relaxation* for RVRP, which is an example of the set-partitioning model for vehicle routing with time windows (see [27]). While it is not difficult to come up with such (approximately-solvable) configuration LPs for vehicle-routing problems, and they have been observed computationally to provide excellent lower bounds on the optimal value [9], there are few theoretical bounds on the effectiveness of these LPs. Moreover, the limited known guarantees (for general metrics) typically only establish logarithmic bounds [20, 5], which follow from the observation that the configuration LP can be viewed as a standard set-cover LP. In contrast, we exploit the structure of our configuration LP for RVRP using novel methods and prove a *constant* integrality gap for the relaxation, which serves to better justify the good empirical performance of these LPs. Although configuration LPs are often believed to be powerful, they have been leveraged only sporadically in the design of approximation algorithms; some notable exceptions are [16, 3, 26, 23, 10]. Our work contributes to the toolkit of techniques that have been utilized for configuration LPs, and our techniques may find wider applicability.

We use our algorithm for additive-RVRP to obtain an $O\big(\log(\frac{R}{R-1})\big)$-approximation for multiplicative-RVRP with regret bound R (Theorem 4.2). Thus, we obtain a constant-factor approximation for any fixed R.

Interestingly, our algorithm for RVRP yields *improved guarantees* for (the path-variant of) the classical *distance-constrained vehicle-routing problem* (DVRP) [17, 19, 20]—find the fewest number of rooted paths of *length* at most D that cover all the nodes—via a reduction to RVRP. (DVRP usually refers to the version where we seek tours containing the root; [20] shows that the path- and tour-versions are within a factor of 2 in terms of approximability.) We obtain an $O\big(\frac{\log R_{\max}}{\log\log R_{\max}}\big)$-approximation for DVRP (Theorem 5.1), where $R_{\max} \leq D$ is the maximum regret of a node in an optimal solution, which improves upon the previous-best $O(\log D)$-guarantee for DVRP [20]. We believe that this reduction is of independent interest. We also exploit our LP-based guarantee for RVRP to show that the integrality gap of the natural configuration LP for DVRP is at most $O(OPT_{\text{LP}})$, where OPT_{LP} is the optimal value of the LP. This is interesting because for the standard set-cover LP, there are $O(\log n)$-integrality-gap examples even when the optimal LP-value is a constant; although the configuration LP for DVRP is also a set-cover LP, our result precludes such an integrality-gap construction for this LP and raises the enticing possibility that the additional structure in DVRP can be further exploited, perhaps by refining our methods, to derive improved guarantees.

We leverage our techniques to obtain guarantees for various variants and generalizations of RVRP (Section 6), including, most notably, (i) the variants where we fix the number k of rooted paths (used to cover the nodes) and seek to minimize the maximum additive/multiplicative regret of a node,

[1] The distinction between the delay and additive regret of a client is akin to the distinction between the completion time and flow time of a job in scheduling problems.

which we refer to as *additive/multiplicative- kRVRP*; and (ii) (additive) RVRP and kRVRP in *asymmetric metrics*.

We obtain an $O(k^2)$-approximation for additive-kRVRP, (Theorem 6.2), which is the *first* approximation guarantee for kRVRP. Previously, the only approximation results known for kRVRP were for the special cases where we have a tree metric [5] (note that the $O(\log n)$-distortion embedding of general metrics into tree metrics does not approximate regret), and when $k = 1$ [4]. In particular, *no* approximation guarantees were known previously *even when* $k = 2$; in contrast, we achieve a constant-factor approximation for any fixed k. Partially complementing this result, we show that the integrality gap of the configuration LP for kRVRP is $\Omega(k)$ (Theorem 7.3). Multiplicative-kRVRP turns out to be an easier problem, and the LP-rounding ideas in [6] yield an $O(1)$-approximation for this problem (Theorem 6.3).

For asymmetric metrics, we exploit the simple but key observation that regret can be captured via a suitable asymmetric metric that we call the *regret metric* c^{reg} (see Fact 2.1). This alternative view of regret yields surprising dividends, since we can directly plug in results for asymmetric metrics to obtain results for regret problems. In particular, results for k-person asymmetric s-t TSP-path [12, 11] translate to results for asymmetric RVRP and kRVRP, and we achieve approximation ratios of $O(\log n)$ and $O(k^2 \log n)$ respectively for these two problems. Although regret metrics form a strict subclass of asymmetric metrics, we uncover an interesting connection between the approximability of asymmetric RVRP and ATSP. We show that an α-approximation for asymmetric RVRP implies a 2α-approximation for ATSP (Theorem 7.2); thus an $\omega(\log\log n)$-improvement to the approximation we achieve for asymmetric RVRP would improve the current best $O\big(\frac{\log n}{\log\log n}\big)$-approximation for ATSP [1].

Our techniques. Our algorithm for additive-RVRP (see Section 3) is based on rounding a fractional solution to a natural configuration LP (P), where we have a variable for every path of regret at most R and we enforce that every node is covered to an extent of 1 by such paths. Although this LP has an exponential number of variables, we can obtain a near-optimal solution x^* by using an approximation algorithm for *orienteering* [4, 7] (see "Related work") to provide an approximate separation oracle for the dual LP.

Let $k^* = \sum_P x^*_P$. To round x^*, we first observe that it suffices to obtain $O(k^*)$ paths of *total regret* $O(k^* R)$ (see Lemma 2.2). At a high level, we would ideally like to ensure that directing the paths in the support of x^* away from the root yields a directed acyclic graph H. If we have this, then by viewing x^* as the path decomposition of a flow in H, and by the integrality property of flows, we can round x^* to an integral flow that covers all the nodes, has value at most $\lceil k^* \rceil$, and whose cost in the regret metric is at most the c^{reg}-cost of x^*, which is at most $k^* R$. This integral flow decomposes into a collection of $\lceil k^* \rceil$ paths that cover V (since H is acyclic), which yields the desired rounding.

Of course, we need not be in this ideal situation. Our goal will be to identify a subset W of "witness nodes" such that: (a) x^* can be converted into a fractional solution that covers W and has the above acyclicity-property without blowing up the c^{reg}-cost by much; and (b) nodes in $V \setminus W$ can be attached to W incurring only an $O(k^* R)$ cost. The new fractional solution can then be rounded to obtain integral paths that cover W, which can then be extended so that they cover V. In achieving this goal, we gain significant leverage from the fact that the configuration LP yields a collection of fractional paths that cover all the nodes, which is a stronger property than having a flow where every node has at least one unit of incoming flow. We build a forest F of cost $O(k^* R)$ and select one node from each component of F as a witness node; this immediately satisfies (b). The construction ensures that: first, every witness node w has an associated collection of "witness paths" that cover it to a large extent, say, $\frac{1}{2}$; and second, for every path P, the witness nodes that use P as a witness path have strictly increasing distances from the root r and occur on P in order of their distance from r. It follows that by shortcutting each path to only contain the witness nodes that use the path as a witness path, and blowing up the x^* values by 2, we achieve property (a).

Our algorithms for multiplicative-RVRP and DVRP capitalize on the following insight. If there exist k paths covering a given set S of nodes incurring additive regret at most R for these nodes, then, for any $\epsilon > 0$, we can use our algorithm for RVRP to cover S with $O\big(\frac{k}{\epsilon}\big)$ paths causing additive regret at most ϵR to the nodes in S. For multiplicative-RVRP with regret bound R, an $O\big(\log(\frac{R}{R-1})\big)$-approximation follows by applying this observation to the nodes in every "ring" $V_i := \{v : c_{rv} \in [2^{i-1}, 2^i)\}$, and concatenating the paths obtained for the V_is whose indices are $O\big(\log(\frac{R}{R-1})\big)$ apart.

For DVRP, we use dynamic programming to obtain the regret bounds and the corresponding node-sets to cover via paths satisfying the regret bound. Crucially, in the analysis, we bound the number of paths needed to cover a set of nodes with a given regret bound by suitably *modifying* the paths of a *structured near-optimal solution* $\mathcal{O}$. We then argue that a specific choice (depending on $\mathcal{O}$) of regret bounds and node-sets yields an $O\big(\frac{\log R_{\max}}{\log\log R_{\max}}\big)$-approximation. In doing so, we argue that each choice of regret bound is such that we make progress by decreasing either the regret bound *or* the number of paths needed. Since our RVRP-algorithm is in fact LP-based, this also yields a bound on the integrality gap of the natural configuration LP for DVRP.

For the $O(OPT_{\mathrm{LP}})$ integrality-gap result for DVRP, we show that one can partition the nodes so that for each part S, there is a distinct node t_S such that the paths ending at t_S cover the S-nodes to an extent of $\Omega\big(\frac{1}{OPT_{\mathrm{LP}}}\big)$. Multiplying the LP-solution by $O(OPT_{\mathrm{LP}})$ then yields a fractional solution that covers the S-nodes

Related work. There is a wealth of literature on vehicle routing problems (see, e.g., [27]), and the survey [22] discusses a variety of problems under the umbrella of schoolbus-routing problems; we limit ourselves to the work that is relevant to our problems. The use of regret as a vehicle-routing objective seems to have been first considered in [25], who present various heuristics and empirical results.

Bock et al. [5] developed the first approximation algorithms for RVRP, but focus mainly on tree metrics, for which they achieve a 3-approximation. For general metrics, they observe that RVRP can be cast as a covering problem, and finding a minimum-density set is an *orienteering* problem [14, 4]: given node rewards, end points s, t, and a length bound B, find an s-t path of length at most B that gathers maximum total node-reward. Thus, the greedy set-cover algorithm combined with a suitable $O(1)$-approximation for orienteering [4, 7] immediately yields an $O(\log n)$-approximation for RVRP. Previously, this was the best approximation algorithm for RVRP in general metrics. For kRVRP, *no* previous results were known for general metrics, even when $k = 2$. (Note that we obtain a constant approximation for kRVRP for any fixed k.) [5] obtain a 12.5-approximation for kRVRP in tree metrics. When $k = 1$, kRVRP becomes as a special case of the *min-excess path* problem introduced by [4], who devised a $(2+\epsilon)$-approximation for this problem.

To the best of our knowledge, multiplicative regret, and the asymmetric versions of RVRP and kRVRP have not been considered previously. Our algorithm for multiplicative-kRVRP uses the LP-based techniques developed by [6] for the minimum latency problem. The set-cover greedy algorithm can also be applied to asymmetric RVRP. This yields approximation ratios of $O\big(\frac{\log^3 n}{\log\log n}\big)$ in polytime, and $O(\log^2 n)$ in *quasi-polytime* using the $O\big(\frac{\log^2 n}{\log\log n}\big)$- and $O(\log OPT)$- approximation algorithms for directed orienteering in [21] and [8] respectively. Both factors are significantly worse than the $O(\log n)$-approximation that we obtain via an easy reduction to kATSPP (find k s-t paths of minimum total cost that cover all nodes). Friggstad et al. [12] obtained the first results for kATSPP which were later improved by [11] to an $O(k\log n)$-approximation and a bicriteria result that achieves $O(\log n)$-approximation using at most $2k$ paths.

Replacing the notion of client-regret in our problems with client-delay gives rise to some well-known vehicle-routing and TSP problems. The client-delay version of RVRP corresponds to (path-) DVRP. Nagarajan and Ravi [20] give an $O(\log\min\{D, n\})$-approximation for general metrics, and a 2-approximation for trees. Obtaining a constant-factor approximation for DVRP in general metrics has been a longstanding open problem. As noted earlier, regret can be captured by the asymmetric regret metric and thus RVRP is *precisely* (path-) DVRP in the regret metric. Thus, our work yields an $O(1)$-approximation for DVRP in this specific asymmetric metric. We find this to be quite interesting and surprising since one would normally expect that DVRP would become *harder* in an asymmetric metric.

The client-delay version of kRVRP yields the kTSP problem of finding k rooted paths of minimum maximum cost that cover all nodes, which admits a constant-factor approximation via a reduction to TSP.

The orienteering problem plays a key role in vehicle-routing problems, including our algorithm for RVRP where it yields an approximate separation oracle for the dual LP. Blum et al. [4] obtained the first constant-factor approximation algorithm for orienteering, and the current best approximation is $2+\epsilon$ due to Chekuri et al. [7]. [21, 7] study (among other problems) directed orienteering and obtain approximation ratios of $O\big(\frac{\log^2 n}{\log\log n}\big)$ and $O(\log^2 OPT)$ respectively. The backbone of all of these algorithms is the min-regret K-path problem (called the min-excess path problem in [4])—choose a min-regret path covering at least K nodes—which captures kRVRP when $k = 1$.[8] used a different approach and gave a quasi-polytime $O(\log OPT)$-approximation for directed orienteering. Finally, Bansal et al. [2] and Chekuri et al. [7] consider orienteering with time windows, where nodes have time windows and we seek to maximize the number of nodes that are visited in their time windows, and its special case where nodes have deadlines, both of which generalize orienteering. They obtain polylogarithmic approximation ratios for these problems.

2. PRELIMINARIES

Recall that an instance of RVRP is specified by a complete undirected graph $G = (\{r\}\cup V, E)$, where r is a distinguished root node, with metric edge costs $\{c_{uv}\}$, and a regret bound R. Let $n = |V| + 1$. We call a path in G rooted if it begins at r. Unless otherwise stated, we think of the nodes on P as being ordered in increasing order of their distance along P from r, and directing P away from r means that we direct each edge $(u, v) \in P$ from u to v if u precedes v (under this ordering). We use D_v to denote c_{rv} for all $v \in V \cup \{r\}$. For a set S of edges, we sometimes use $c(S)$ to denote $\sum_{e\in S} c_e$. We may assume that $c_{uv} > 0$ and is an integer for all $(u, v) \in V \cup \{r\}$ since nodes at distance 0 from each other may be merged.

Unless otherwise stated, we focus throughout on additive regret. It will be convenient to assume that $R > 0$: if $R = 0$ then we can determine whether an edge (u, v) lies on a shortest rooted path, and if so direct (u, v) as $u \to v$ if $D_v = D_u + c_{uv}$, to obtain a directed acyclic graph (DAG) H. Our problem then reduces to finding the minimum number of directed rooted paths in H to cover all the nodes, which can be solved efficiently using network-flow techniques. The following equivalent way of viewing regret will be convenient. For every ordered pair of nodes $u, v \in V \cup \{r\}$, define the *regret distance* (with respect to r) to be $c^{\mathsf{reg}}_{uv} := D_u + c_{uv} - D_v$.

Fact 2.1 *(i) The regret distances c^{reg}_{uv} are nonnegative and satisfy the triangle inequality: $c^{\mathsf{reg}}_{uv} \le c^{\mathsf{reg}}_{uw} + c^{\mathsf{reg}}_{wv}$ for all $u, v, w \in V \cup \{r\}$. Hence, $\{c^{\mathsf{reg}}_{uv}\}$ forms an asymmetric metric that we call the* regret metric.
(ii) For a $u \rightsquigarrow v$ path P, we have $c^{\mathsf{reg}}(P) := \sum_{e\in P} c^{\mathsf{reg}}_e = D_u + c(P) - D_v$, and for a cycle Z, we have $c^{\mathsf{reg}}(Z) = c(Z)$. Properties (i) and (ii) hold even when the underlying $\{c_{uv}\}$ metric is asymmetric.

We infer from Fact 2.1 that if P is a rooted path and $v \in P$, then the regret of v with respect to P is simply the c^{reg}-distance to v along P, which we denote by $c^{\mathsf{reg}}_P(v)$, and the regret of nodes on P cannot decrease as one moves away from the root (since $c^{\mathsf{reg}} \ge 0$). We define the regret of P to be the regret of the end-node of P, which by part (ii) of Fact 2.1 is given by $c^{\mathsf{reg}}(P) = \sum_{e\in P} c^{\mathsf{reg}}_e$.

Lemma 2.2 makes the key observation that one can always convert a collection of paths with *average regret* at most αR into one where every path has regret at most R by blowing up the number of paths by an $O(\max\{\alpha, 1\})$ factor, and hence, it suffices to obtain a near-optimal solution with average regret $O(R)$.

Lemma 2.2 *Given rooted paths $P_1, \ldots, P_k$ with total regret αkR, we can efficiently find at most $(\alpha + 1)\cdot k$ rooted paths, each regret at most R, that cover $\bigcup_{i=1}^k P_i$.*

PROOF. Let $\alpha_1 R, \ldots, \alpha_k R$ be the regrets of $P_1, \ldots, P_k$ respectively. We show that for each path P_i, we can obtain $\max\{\lceil\alpha_i\rceil, 1\}$ rooted paths of regret at most R that cover the nodes of P_i. Applying this to each path P_i, we obtain at most $\sum_{i=1}^k(\alpha_i + 1) = (\alpha + 1)\cdot k$ rooted paths with regret at most R that cover $\bigcup_{i=1}^k P_i$.

Fix a path P_i. If $\alpha_i \le 1$, there is nothing to be done, so assume otherwise. The idea is to simply break P_i at each point where the regret exceeds a multiple of R, and connect the starting point of each such segment directly to r. More formally, for $\ell = 1, \ldots, \beta_i := \lceil \alpha_i \rceil - 1$, let v_ℓ be the first node on P with $c_P^{\mathsf{reg}}(v) > \ell R$, and let $u_{\ell-1}$ be its (immediate) predecessor on P. Let $v_0 = r$ and u_{β_i} be the end point of P_i. We create the $\lceil \alpha_i \rceil$ paths given by $r, v_\ell \rightsquigarrow u_\ell$ for $\ell = 0, \ldots, \beta_i$, which clearly together cover the nodes of P_i. The regret of each such path is $c_{rv_\ell}^{\mathsf{reg}} + c_P^{\mathsf{reg}}(u_\ell) - c_P^{\mathsf{reg}}(v_\ell) = c_P^{\mathsf{reg}}(u_\ell) - c_P^{\mathsf{reg}}(v_\ell) \leq (\ell+1)R - \ell R = R$, where the last inequality follows from the definitions of v_ℓ, $v_{\ell+1}$ and u_ℓ (which precedes $v_{\ell+1}$). $\square$

Algorithms for symmetric TSP variants often exploit the fact that edges may be traversed in any direction, to convert a connected subgraph into an Eulerian tour while losing a factor of 2 in the cost. This does not work for RVRP since c^{reg} is an asymmetric metric. Instead, we exploit a key observation of Blum et al. [4], who identify portions of a rooted path P whose total c-cost can be charged to $c^{\mathsf{reg}}(P)$.

Definition 2.3 Let P be a rooted path ending at w. Consider an edge (u, v) of P, where u precedes v on P. We call this a *red* edge of P if there exist nodes x and y on the r-u portion and v-w portion of P respectively such that $D_x \geq D_y$; otherwise, we call this a *blue* edge of P. For a node $x \in P$, let $\mathsf{red}(x, P)$ denote the maximal subpath Q of P containing x consisting of only red edges (which might be the trivial path $\{x\}$).

We call a maximal blue/red subpath of a rooted path P a blue/red interval of P. The blue and red intervals of P correspond roughly to the type-1 and type-2 segments of P, as defined in [4]. Distinguishing the edges on P as red or blue serves two main purposes. First, the total cost of the red edges is proportional to the regret of P (Lemma 2.4). Second, if we shortcut P so that it contains only one node from each red interval, then the resulting edges must all be distance increasing (Lemma 2.5). Consequently, if we perform this operation on a collection of paths and direct edges away from the root, then we obtain a DAG.

Lemma 2.4 (Blum et al. [4]) *For any rooted path P, we have $\sum_{e \text{ red on } P} c_e \leq \frac{3}{2} c^{\mathsf{reg}}(P)$.*

Lemma 2.5 *(i) Suppose u, v are nodes on a rooted path P such that u precedes v on P and $\mathsf{red}(u, P) \neq \mathsf{red}(v, P)$, then $D_u < D_v$. (ii) Hence, if P' is obtained by shortcutting P so that it contains at most one node from each red interval of P, then for every edge (x, y) of P' with x preceding y on P', we have $D_x < D_y$.*

PROOF. Since u precedes v on P and $\mathsf{red}(u, P) \neq \mathsf{red}(v, P)$, there must be some edge $(a, b) \in P$ such that (a, b) is blue on P, and a, b lie on the u-v portion of P (it could be that $a = u$ and/or $b = v$). So if $D_u \geq D_v$ then (a, b) would be classified as red. Part (ii) follows immediately from part (i). $\square$

Orienteering. Our algorithms are based on rounding the solution to an exponential-size LP-relaxation of the problem. A near-optimal solution to this LP can be obtained by solving the dual LP approximately. The separation oracle for the dual LP corresponds to a *point-to-point orienteering* problem, which is defined as follows. We are given an undirected complete graph with nonnegative node-rewards, edge lengths that form a metric, origin and destination nodes s, t, and a length bound B. The goal is to find an s-t path P of total length at most B that gathers maximum total reward. In the *rooted orienteering* problem, we only specify the origin s, and a path rooted at s. Unless otherwise stated, we use orienteering to mean point-to-point orienteering. Clearly, an algorithm for orienteering can also be used for rooted orienteering. A related problem is the *min-excess path* (MEP) problem defined by [4], where we are given s, t, and a target reward Π, and we seek to find an s-t path of minimum regret that gathers reward at least Π.

In the unweighted version of these problems, all node rewards are 0 or 1. Observe that the weighted versions of these problems can be reduced to their unweighted version in pseudopolynomial time by making co-located copies of a node. For orienteering, by suitably scaling and rounding the node-rewards, one can obtain a $\mathsf{poly}\big(\text{input size}, \frac{1}{\epsilon}\big)$-time reduction where we lose a $(1+\epsilon)$-factor in approximation. For MEP, this data rounding yields a bicriteria approximation where we obtain an s-t path with reward at least $\Pi/(1+\epsilon)$. Both the unweighted and weighted versions of orienteering and MEP are *NP*-hard. The current best approximation factors for these problems are $(2+\epsilon)$ for orienteering due to Chekuri et al. [7], and $(2+\epsilon)$ for *unweighted* MEP due to Blum et al. [4], for any positive constant ϵ.

3. AN LP-ROUNDING CONSTANT-FACTOR APPROXIMATION FOR (ADDITIVE) RVRP

We consider the following configuration-style LP-relaxation for RVRP, which was also mentioned in [5]. Let $\mathcal{C}_R$ denote the collection of all rooted paths with regret at most R. We introduce a variable x_P for each path $P \in \mathcal{C}_R$ to denote if P is chosen. Throughout, we use P to index paths in $\mathcal{C}_R$.

$$\min \sum_P x_P \quad \text{s.t.} \quad \sum_{P: v \in P} x_P \geq 1 \quad \forall v \in V, \quad x \geq 0. \qquad \text{(P)}$$

Let OPT denote the optimal value of (P). Note that $OPT \geq 1$. It is easy to give a reduction from TSP showing that it is *NP*-complete to decide if there is a feasible solution that uses only 1 path; hence, it is *NP*-hard to achieve an approximation factor better than 2 (Theorem 7.1). Complementing this, we devise an algorithm for RVRP based on LP-rounding that achieves a constant approximation ratio (and thus yields a corresponding integrality-gap bound). This is a significant improvement over the previous-best $O(\log n)$-approximation ratio obtained by [5]. Although (P) has an exponential number of variables, one can obtain a near-optimal solution x^* by solving the dual LP (which has an exponential number of constraints) to near-optimality via the use of an approximation algorithm for orienteering to obtain an approximate separation oracle for the dual.

Lemma 3.1 *We can use a γ_{orient}-approximation algorithm for orienteering to efficiently compute a feasible solution x^* to* (P) *of value at most $\gamma_{\mathsf{orient}} \cdot OPT$.*

Let $k^* = \sum_P x_P^* \leq \gamma_{\mathsf{orient}} \cdot OPT$. Our goal is to round x^* to a solution using at most $O(k^*)$ paths that have average regret $O(R)$. We can then apply Lemma 2.2 to obtain $O(k^*)$ paths, each having regret at most R, and thereby obtain an $O(1)$-approximate solution. We prove the following theorem.

Theorem 3.2 *We can efficiently round x^* to a solution using at most $(8 + 4\sqrt{3})k^* + 1$ rooted paths. This yields*

$(8+4\sqrt{3})\gamma_{\mathsf{orient}} OPT + 1 \leq 30.86 \cdot OPT$ *rooted paths by taking* $\gamma_{\mathsf{orient}} = 2+\epsilon$ *[7], and shows that the integrality gap of* (P) *is at most* $9+4\sqrt{3} \leq 15.93$.

We present a rounding procedure that obtains a slightly worse approximation ratio. We show how to obtain the guarantee stated above in the full version of the paper. Let $\mathrm{supp}(x^*) := \{P : x^*_P > 0\}$ be the paths in the support of x^*. To gain some intuition, suppose first that it happens that when we direct every path $P \in \mathrm{supp}(x^*)$ away from r, we obtain a directed graph H that is acyclic. We can then set up a network-flow problem to find a minimum c^{reg}-cost flow in H of value at most $\lceil k^* \rceil$ such that every node has at least one unit of flow entering it. Since x^* can be viewed as a path decomposition of a feasible flow of c^{reg}-cost at most k^*R, by the integrality property of flows, there is an integral flow of c^{reg}-cost at most k^*R. Since H is acyclic, this flow may be decomposed into at most $\lceil k^* \rceil$ paths that cover all the nodes, and the average regret of this path collection is at most R, so we obtain the desired rounding.

Of course, in general H will not be acyclic and rounding x^* as above may yield an integral flow that does not decompose into a collection of only paths. So we seek to identify a subset $W \subseteq V$ of "witness" nodes and a collection of $O(k^*)$ fractional paths from $\mathcal{C}_R$ covering W such that: (a) directing each path in this collection away from r yields a DAG; and (b) given any collection of integral paths covering W, one can graft the nodes of $V \setminus W$ into these paths (to obtain new paths covering V) incurring an additional c^{reg}-cost of $O(k^*R)$. Property (a) allows one to use the aforementioned network-flow argument to obtain $O(k^*)$ paths covering W with total regret $O(k^*R)$, and property (b) enables one to modify this to obtain $O(k^*)$ (integral) paths covering V while keeping the total regret to $O(k^*R)$ (so that one can then apply Lemma 2.2).

To obtain W, we carefully construct a forest F of cost $O(k^*R)$ (step A1 below) with the property that for every component Z of F, we can associate a single node $w \in Z$, which we include in W, such that there is a total x^*-weight of at least 0.5 in paths P containing w for which $\mathsf{red}(w,P) \subseteq Z$. Notably, we achieve this in a rather clean and simple way by defining a *downwards-monotone cut-requirement function* based on the fractional solution x^* that encodes the above requirement, an idea that we believe has wider applicability, especially for network-design problems.

Once we have such a forest, property (b) holds by construction since the total cost of F is $O(k^*R)$ (Lemma 3.3). Moreover (step A2), if we shortcut each path $P \in \mathrm{supp}(x^*)$ so that it only contains nodes $w \in W$ for which $\mathsf{red}(w,P)$ is contained in some component of F, then the resulting paths cover each node in W to an extent of at least 0.5 and satisfy the conditions of part (ii) of Lemma 2.5 (see Lemma 3.4). So by doubling the fractional values of the resulting paths, we obtain a fractional-path collection satisfying property (a). Hence, we can obtain $O(k^*)$ integral paths covering W (step A3) and attach the nodes of $V \setminus W$ to these paths (step A4) while ensuring that the total regret remains $O(k^*R)$ (Lemma 3.5), and then apply Lemma 2.2. We prove in Theorem 3.6 that the resulting solution uses at most $16k^*+1 \leq 16\gamma_{\mathsf{orient}} \cdot OPT + 1$ paths. The improved guarantee stated in Theorem 3.2 follows by fine-tuning the threshold used to form the forest F. We now describe the algorithm in detail and proceed to analyze it.

Algorithm 1 Input: A fractional solution x^* to (P); $k^* = \sum_P x^*_P$. Output: $O(k^*)$ paths with regret at most R covering all the nodes.

A1. **Finding a low-cost forest F.** For a subset $S \subseteq V \cup \{r\}$ and a node v, define $\tau(v,S) := \sum_{P:\mathsf{red}(v,P)\subseteq S} x^*_P$; define $f(S)=1$ if $\tau(v,S) < \frac{1}{2}$ for all $v \in S$, and 0 otherwise. Note that f is a *downwards-monotone* cut-requirement function: if $\emptyset \neq A \subseteq B$ then $f(A) \geq f(B)$. We call a set S with $f(S)=1$, an *active* set.

- A1.1 Use the 2-approximation algorithm for $\{0,1\}$ downwards-monotone functions in [13] to obtain a forest F such that $|\delta(S) \cap F| \geq f(S)$ for every set $S \subseteq V \cup \{r\}$.
- A1.2 For every component Z of F, obtain a tour $h(Z)$ traversing all nodes of Z by doubling the edges of Z and shortcutting. If $r \notin Z$, choose a node $w \in Z$ such that $\tau(w,Z) \geq \frac{1}{2}$ (which exists since $f(Z)=0$); call w the *witness node* for Z, and denote Z by Z_w. Let $W \subseteq V$ be the set of all witness nodes.

A2. **Obtaining a fractional acyclic flow covering W.**

- A2.1 For every path $P \in \mathrm{supp}(x^*)$ we do the following. Let $P_W \subseteq P \cap W$ be the set of witness nodes $w \in P$ such that $\mathsf{red}(w,P)$ is contained in Z_w. We shortcut P past the nodes in $P \setminus (P_W \cup \{r\})$ to obtain a rooted path $\phi(P)$ spanning the nodes in P_W. Note that shortcutting does not increase the c^{reg}-cost. Let $\mathcal{C}' = \{\phi(P) : P \in \mathrm{supp}(x^*), \phi(P) \neq \{r\}\} \subseteq \mathcal{C}_R$ denote this new collection of non-trivial paths.
- A2.2 Let $H = (\{r\} \cup V, A_H)$ be the digraph obtained by directing each path in $\mathcal{C}'$ away from r. Let $z = (z_a)_{a \in A_H}$ be the flow that sends $\sum_{P:\phi(P)=P'} x^*_P$ flow along each path $P' \in \mathcal{C}'$. We prove in Lemma 3.4 that H is acyclic, and that $z^{\mathrm{in}}(w) := \sum_{a \in \delta^{\mathrm{in}}_H(w)} z_a \geq \frac{1}{2}$ for every $w \in W$.

A3. Use the integrality property of flows to round $2z$ to an integer flow $\hat{z}$ of no greater c^{reg}-cost and value $k \leq \lceil 2k^* \rceil$ such that $\hat{z}^{\mathrm{in}}(w) \geq 1$ for every $w \in W$. Since H is acyclic, we may decompose $\hat{z}$ into k rooted paths $\hat{P}_1, \ldots, \hat{P}_k$ so that (possibly after shortcutting) every node of W lies on exactly one $\hat{P}_i$ path.

A4. **Grafting in the nodes of $V \setminus W$.** If there is a component Z of F containing r, pick an arbitrary path, say $\hat{P}_1$; modify $\hat{P}_1$ by traversing $h(Z)$ first and then visiting the nodes of $\hat{P}_1 \setminus \{r\}$ (in the same order as $\hat{P}_1$). Next, for every path $\hat{P}_i$, $i=1,\ldots,k$, we walk along $\hat{P}_i$ and each time we visit a new node $w \in W$ on $\hat{P}_i$ we traverse $h(Z_w)$ before moving on to the next node on $\hat{P}_i$. Let $\tilde{P}_i$ denote the resulting new path.

A5. Apply Lemma 2.2 to $\tilde{P}_1, \ldots, \tilde{P}_k$ to obtain the final set of paths (having maximum regret R).

Analysis. Let $\mathcal{S}(F)$ denote the set of components of F. Note that $V \subseteq \bigcup_{Z \in \mathcal{S}(F)} Z$.

Lemma 3.3 *The forest F computed in step A1 has cost at most $6 \cdot k^* \cdot R$. Thus, $\sum_{Z\in\mathcal{S}(F)} c(h(Z)) \leq 12k^*R$.*

Proof. Consider the following LP for covering the cuts $\delta(S)$ for sets S in $\mathcal{A} = \{S \subseteq V \cup \{r\} : S \neq \emptyset, f(S)=1\}$.

$$\min \sum_e c_e z_e \quad \text{s.t.} \quad z(\delta(S)) \geq 1 \quad \forall S \in \mathcal{A}, \quad z \geq 0. \qquad \text{(C)}$$

Define z by setting $z_e = \sum_{P:e \text{ is red on } P} 2 \cdot x^*_P$ for all e. This is a feasible solution to (C) since for every active set S and every node $v \in S$, we have

$$\frac{1}{2} < 1 - \tau(v,S) \leq \sum_{P:\mathsf{red}(v,P)\not\subseteq S} x^*_P \leq \sum_{e\in\delta(S)} \Big(\sum_{P:e\in\mathsf{red}(v,P)} x^*_P \Big) \leq z(\delta(S))/2.$$

Also, $\sum_e c_e z_e = 2\sum_P x^*_P\big(\sum_{e \text{ red on } P} c_e\big) \leq 3\sum_P c^{\mathsf{reg}}(P)x^*_P \leq 3k^*R$. The penultimate inequality follows from Lemma 2.4, and the last inequality follows because $\text{supp}(x^*) \subseteq \mathcal{C}_R$ and $\sum_P x^*_P = k^*$. The 2-approximation algorithm of [13] then guarantees that $c(F) \leq 2 \cdot OPT_{(\text{C})} \leq 6k^*R$. Therefore, $\sum_{Z\in\mathcal{S}(F)} c(h(Z)) \leq 2c(F) \leq 12k^*R$. □

Lemma 3.4 *(i) For every path $P \in \mathcal{C}_R$, every red interval of P contains at most one node of P_W; so $\phi(P)$ visits nodes v in strictly increasing order of D_v. (ii) $\sum_{P:w\in\phi(P)} x^*_P \geq \frac{1}{2}$ for all $w \in W$. (iii) Hence, the digraph H constructed in step A2 is acyclic, and $z^{\text{in}}(w) \geq \frac{1}{2}$ for all $w \in W$.*

Proof. Part (iii) follows immediately from parts (i) and (ii). For part (i), recall that $P_W = \{w \in P\cap W : \mathsf{red}(w,P) \subseteq Z_w\}$. If there are two nodes u, w of P_W contained in some red interval of P then $Z_u \cap Z_w \neq \emptyset$, but this contradicts the fact that we add at most one node to W from each component of F. It follows that $\phi(P)$ contains at most one node from each red interval of P, and by Lemma 2.5, we have that $\phi(P)$ visits nodes v in strictly increasing order of distance D_v. For part (ii), we note that for a node $w \in W$, by definition, we have that $w \in \phi(P)$ iff $\mathsf{red}(w,P) \subseteq Z_w$. So $\sum_{P:w\in\phi(P)} x^*_P = \sum_{P:\mathsf{red}(w,P)\subseteq Z_w} x^*_P = \tau(w, Z_w) \geq \frac{1}{2}$, where the last inequality follows from the definition of Z_w. □

Lemma 3.5 *The total regret of the paths $\tilde{P}_1,\ldots,\tilde{P}_k$ obtained in step A4 is at most $14 \cdot k^* \cdot R$.*

Theorem 3.6 *Algorithm 1 returns a feasible solution with at most $16k^* + 1 \leq 16\gamma_{\mathsf{orient}} \cdot OPT + 1$ paths.*

Proof. Applying Lemma 2.2 to the paths $\tilde{P}_1,\ldots,\tilde{P}_k$, which have total regret at most $14k^*R$ (by Lemma 3.5), we obtain a collection of k' rooted paths of maximum regret R whose union covers all nodes, where $k' \leq \big(\frac{14k^*}{k}+1\big)k \leq 14k^* + k \leq 14k^* + \lceil 2k^* \rceil \leq 16k^* + 1$. □

4. MULTIPLICATIVE-RVRP

Recall that in multiplicative-RVRP, we are given a regret bound R, and we want to find the minimum number of paths covering all nodes so that each node v is visited by time $R \cdot D_v$. When $R = 1$, the problem can be solved in polytime (as this is simply additive-RVRP with regret bound 0), so we assume that $R > 1$. The following observation, which falls out of Lemma 2.2 will be quite useful.

Lemma 4.1 *Let γ_{RVRP} be the approximation factor of our RVRP-algorithm. Suppose there are k paths covering a given set S of nodes ensuring that every node in S has additive regret at most ρ. For any $\epsilon > 0$, one can efficiently obtain at most $\big\lfloor \gamma_{\mathsf{RVRP}} k \big\lceil \frac{1}{\epsilon} \big\rceil \big\rfloor$ paths covering S such that each node in S has regret at most $\epsilon\rho$.*

Theorem 4.2 *Multiplicative-RVRP can be reduced to additive-RVRP incurring an $O\big(\log(\frac{R}{R-1})\big)$-factor loss. This yields an $O\big(\log(\frac{R}{R-1})\big)$-approximation for multiplicative-RVRP.*

Proof. Let $R = 1+\delta$. For $i \geq 1$, define $V_i = \{v : 2^{i-1} \leq D_v < 2^i\}$. Note that the V_is partition the non-root nodes. Let O^* denote the optimal value of the multiplicative-RVRP instance. We apply Lemma 4.1 with $\epsilon = \frac{1}{4}$ to the V_is: for each V_i, there are O^* paths covering V_i such that each node in V_i has regret at most $\delta \cdot 2^i$, so we obtain at most $N = \lfloor 4\gamma_{\mathsf{RVRP}} O^* \rfloor = O(O^*)$ paths covering V_i such that each node in V_i has regret at most $\delta \cdot 2^{i-2}$. Pad these with the trivial path $\{r\}$ if needed, to obtain exactly N paths $P^i_1,\ldots,P^i_N$.

Let $M = \big\lceil \log_2\big(3+\frac{8}{\delta}\big)\big\rceil = O\big(\log_2(\frac{R}{R-1})\big)$. Now for every index $i = 1,2,\ldots,M$ and every $j = 1,\ldots,N$, we concatenate the paths $P^i_j, P^{i+M}_j, P^{i+2M}_j, \ldots$ by moving from the end-node of P^{i+aM}_j to r before following $P^{i+(a+1)M}_j$ for each $a \geq 0$. This yields MN paths that together cover all nodes.

To finish the proof, we show that every node v is visited by time $R \cdot D_v$. Suppose $v \in V_{i+aM}$ and is covered by path P^{i+aM}_j. It's visiting time is then at most $c_{P^{i+aM}_j}(v) + 2\sum_{a'<a} c(P^{i+a'M}_j) \leq D_v + \delta\cdot 2^{i+aM-2} + 2\big(1+\frac{\delta}{4}\big)\sum_{a'<a} 2^{i+a'M}$ which is at most $D_v\big(1+\frac{\delta}{2}+4\cdot\frac{1+\delta/4}{2^M-1}\big) \leq R\cdot D_v$. □

5. APPLICATIONS TO DVRP

Recall that the goal in DVRP is to find the fewest number of rooted paths of *length* at most D that cover all the nodes. We say that a rooted path P is feasible if $c(P) \leq D$. Let O^* be the optimal value, and $R_{\max} \leq D$ be the maximum regret of a path in an optimal solution, which we can estimate within a factor of 2.

As a warm-up, note that a simple $O\big(\log(\frac{R_{\max}}{D-\max_v D_v})\big)$-approximation follows by applying Lemma 4.1 with $\epsilon = \frac{1}{2}$ to the node-sets $V_0 = \big\{v : D - D_v \geq \frac{R_{\max}}{2}\big\}$, and $V_i = \big\{v : D - D_v \in \big[\frac{R_{\max}}{2^{i+1}}, \frac{R_{\max}}{2^i}\big)\big\}$ for $i = 1,\ldots,N = \Big\lceil \log_2\big(\frac{R_{\max}}{D-\max_v D_v}\big)\Big\rceil - 1$, which partition V. For each V_i, $i \geq 0$, we obtain $O(O^*)$ paths covering V_i, causing regret at most $\frac{R_{\max}}{2^{i+1}}$ to the V_i nodes; so the length-D prefixes of these paths cover V_i.

We now describe a more-refined reduction that yields an improved $O\big(\frac{\log R_{\max}}{\log\log R_{\max}}\big)$-approximation. The algorithm is again based on choosing suitable pairs of regret bounds and node-sets, and covering each node-set using paths of the corresponding regret bound. However, instead of *fixing* the regret bounds to be $\frac{R_{\max}}{2^i}$, we now obtain them by solving a dynamic program (DP). Let $S_i = \{v : D - D_v < 2^i\}$ for $i = 0,\ldots,M = \lceil \log_2 D \rceil$. We use DP to obtain a set of feasible paths $\mathcal{P}(i)$ covering S_i for all i. We use $F(i)$ to denote $|\mathcal{P}(i)|$. For all $0 \leq k < i$, we use our algorithm for RVRP to find a collection $\mathcal{Q}(i,k)$ of paths of regret at most 2^k that cover S_i, Let $\mathcal{P}(0)$ be the fewest number of paths of regret 0 (and hence are feasible) that cover the nodes with $D_v = D$, which we can efficiently compute. For $i > 0$, we set $F(i) = \min_{0\leq k<i}\big(|\mathcal{Q}(i,k)| + F(k)\big)$; if k' is the index that attains the minimum, then we set $\mathcal{P}(i) = \mathcal{P}(k) \cup$ (length-D prefixes of the paths in $\mathcal{Q}(i,k)$). We return the solution $\mathcal{P}(M)$ (which we show is feasible).

The analysis requires various novel ideas. Fix an optimal solution $\mathcal{O}$. We define a suitable set of indices, that is, regret bounds, such that using these indices in the DP yields the desired bound on the number of paths. In order to establish a bound on $F(i)$ by plugging in a suitable index $k < i$ we need two things. First, we need to bound $|\mathcal{Q}(i,k)|$. This requires a more sophisticated analysis than the one suggested by Lemma 4.1. Instead of directly using all the paths from $\mathcal{O}$ to bound the number of paths of certain regret required to cover a given set of nodes, we proceed as follows. We argue that if the paths in $\mathcal{O}$ satisfy a certain property, which we can obtain via preprocessing, then S_i is covered by paths of $\mathcal{O}$ of regret at most 2^i. We *modify* these $\mathcal{O}$-paths by breaking them up (as in Lemma 2.2) to obtain paths of regret at most 2^k that cover S_i, which yields a bound on $|\mathcal{Q}(i,k)|$. Second, we need to argue that we make proper progress when moving from index i to index k. In a crucial departure from the previous analysis, we make progress by either suitably decreasing the *number*, or the maximum regret, of the paths, needed from $\mathcal{O}$ to cover the remaining set of nodes.

Theorem 5.1 *$F(M) \le O\big(\frac{\log R_{\max}}{\log\log R_{\max}}\big)\cdot\gamma_{\mathsf{RVRP}}O^*$. So we obtain an $O\big(\frac{\log R_{\max}}{\log\log R_{\max}}\big)$-approximation algorithm for DVRP.*

In the full version, we prove that the natural configuration LP for DVRP has integrality gap at most $\min\big\{O(OPT_{\mathrm{LP}}), O\big(\frac{\log R_{\max}}{\log\log R_{\max}}\big)\big\}$. (Also, an $O(\log n)$ integrality gap follows from set cover.) This improves upon the $O(\log D)$ integrality gap proved in [20], and presents an interesting contrast with set cover for which there are $O(\log n)$-integrality-gap examples even when the optimal LP-value is a constant.

6. EXTENSIONS

Additive-kRVRP. Here, we fix the number $k \ge 1$ of rooted paths that may be used to cover all the nodes and seek to minimize the maximum regret of a node. We approach kRVRP by considering a related problem, *min-sum (additive) kRVRP*, where the goal is to minimize the sum of the regrets of the k paths. Our techniques are versatile and yield an $O(k)$-approximation for min-sum kRVRP, which directly yields an $O(k^2)$-approximation for kRVRP. These are the *first* approximation guarantees for these problems, even for $k = 2$. The only previous approximation results for kRVRP were for the special cases of tree metrics [5], and when $k = 1$ [4].

As in Section 3, our algorithm for min-sum kRVRP is based on LP rounding. Let $\mathcal{C}$ denote the collection of all rooted paths. We now consider the following LP-relaxation for the problem, where we have a variable x_P for every rooted path. We use OPT_R to denote the optimal value of (P2).

$$\min\quad\Big\{\sum_{P\in\mathcal{C}} c^{\mathsf{reg}}(P)x_P:\quad \sum_{P\in\mathcal{C}:v\in P} x_P \ge 1 \quad \forall v\in V \qquad (1)$$

$$\sum_{P\in\mathcal{C}} x_P \le k, \qquad x\ge 0.\Big\} \qquad \text{(P2)}$$

Lemma 6.1 *We can use a γ_{MEP}-approximation algorithm for unweighted MEP to compute, for any $\epsilon > 0$, a solution x^* satisfying (1), $\sum_{P\in\mathcal{C}} x^*_P \le \frac{k}{1-\epsilon}$, and $\sum_{P\in\mathcal{C}} c^{\mathsf{reg}}(P)x^*_P \le \frac{\gamma_{\mathsf{MEP}}}{1-\epsilon}\cdot OPT_R$, in time $\mathsf{poly}\big(\text{input size}, \frac{1}{\epsilon}\big)$.*

Let $k^* = \sum_{P\in\mathcal{C}} x^*_P$ and $\nu^* = \sum_{P\in\mathcal{C}} c^{\mathsf{reg}}(P)x^*_P$. The rounding procedure in Section 3 yields a bicriteria approximation. Choosing threshold $\delta = 1-\epsilon$ to define the cut-requirement function in step A1 (as opposed to $\frac{1}{2}$ as in A.1.2 of Algorithm 1) yields $\left\lceil\frac{k^*}{\delta}\right\rceil = \lceil(1+O(\epsilon))k\rceil$ paths with total regret at most $\big(\frac{1}{\delta}+\frac{6}{1-\delta}\big)\nu^* = O\big(\frac{1}{\epsilon}\big)OPT_R$.

To obtain a true approximation, we choose ϵ in Lemma 6.1 so that $k^* \le k+\frac{1}{3}$ and set the threshold δ to be $1-\frac{1}{3k+2}$. Steps A1 and A2 of Algorithm 1 then yield a forest F such that $c(F) \le \frac{3}{1-\delta}\cdot\nu^* = 3(3k+2)\nu^*$, a set W of witness nodes, and an acyclic flow z such that $z^{\mathrm{in}}(w) \ge \delta$ for all $w\in W$. The flow $\hat{z} = z/\delta$ is a flow of value $k' \le k^*/\delta \le k+\frac{2}{3}$. But instead of using this to obtain an integral flow of value at most $\lceil k'\rceil$, we use the integrality property of flows in a more subtle manner. We may decompose $\hat{z}$ into a convex combination of integral flows $\tilde{z}_1,\ldots,\tilde{z}_\ell$ such that each $\tilde{z}_i$ is a flow of value at least $\lfloor k'\rfloor$ satisfying $\tilde{z}_i^{\mathrm{in}}(w) \ge 1$ for all $w\in W$. Therefore the convex combination must place a weight of at least $\frac{1}{3}$ on the $\tilde{z}_i$ flows that have value at most k. Choose the flow of value at most k with smallest c^{reg}-cost, and decompose this into $k'' \le k$ rooted paths $\hat{P}_1,\ldots,\hat{P}_{k''}$ so that (maybe after some shortcutting) every node of W lies on exactly one $\hat{P}_i$ path. It follows that the total c^{reg}-cost of $\hat{P}_1,\ldots,\hat{P}_{k''}$ is at most $3\cdot\sum_{a\in H} c^{\mathsf{reg}}_a\hat{z}_a \le 3\cdot\frac{3k+2}{3k+1}\cdot\sum_{a\in H} c^{\mathsf{reg}}_a z_a \le 4\sum_{P\in\mathcal{C}} c^{\mathsf{reg}}(P)x^*_P$. Now we apply step A4 to obtain the final set of paths $\tilde{P}_1,\ldots,\tilde{P}_{k''}$.

Theorem 6.2 *The above algorithm returns at most k rooted paths having total regret $O(k)\cdot\nu^* = O(k\cdot\gamma_{\mathsf{MEP}})\cdot OPT_R$. Thus, we obtain an $O(k)$-approximation algorithm for min-sum kRVRP. This leads to an $O(k^2)$-approximation for kRVRP.*

Partially complementing the above result, we prove in Section 7 that a natural LP-relaxation for kRVRP along the same lines as (P) and (P2) has an integrality gap of $\Omega(k)$.

Multiplicative-kRVRP. This is the version of kRVRP with multiplicative regret. [6] came up with an LP-formulation for the (k-route) minimum-latency problem involving variables $x_{v,t}$ denoting if node v is visited at time t. They prove that an LP-solution x^* can be rounded to a feasible solution where the visiting time of every node v is at most $O(1)\cdot\sum_t tx^*_{v,t}$. We can adapt these ideas to obtain an $O(1)$-approximation for multiplicative-kRVRP, and more generally, an $O(1)$-approximation for the problem of minimizing a weighted sum of the clients' multiplicative regrets.

Theorem 6.3 *There is an $O(1)$-approximation algorithm for multiplicative-kRVRP. This guarantee extends to the setting where we want to minimize a weighted sum of the multiplicative client-regrets (with nonnegative weights).*

Asymmetric metrics. We now consider RVRP and kRVRP in directed graphs, i.e., the distances $\{c_{uv}\}$ now form an asymmetric metric. The regret of a node v with respect to a directed path P rooted at r is defined as before, and we seek rooted (directed) paths that cover all the nodes. We crucially exploit that, as noted in Fact 2.1, the regret distances $\{c^{\mathsf{reg}}_{uv}\}$ continue to form an asymmetric metric. Thus, we readily obtain guarantees for asymmetric RVRP and asymmetric min-sum kRVRP by leveraging known results for k-person s-t asymmetric TSP-path (kATSPP), which is defined as follows: given two nodes s, t in an asymmetric metric and an integer k, find k s-t paths of minimum total cost that cover all the nodes. Friggstad et al. [12] showed how to obtain $O(k\log n)$ s-t paths of cost at most $O(\log n)\cdot OPT_k$, where OPT_k is the minimum-cost kATSPP solution that uses k paths; this was improved by [11] to the following.

Theorem 6.4 ([11]) *For any $b \ge 1$, we can efficiently find at most $k+\frac{k}{b}$ paths of total cost $O(b\cdot\log n)\cdot OPT_k$.*

Theorem 6.4 immediately yields results for asymmetric min-sum kRVRP—since this is simply kATSPP in the regret metric!—and hence, for asymmetric kRVRP.

Theorem 6.5 *There is an $O(k\log n)$-approximation algorithm for asymmetric min-sum kRVRP. This implies an $O(k^2\log n)$-approximation for asymmetric kRVRP.*

We now focus on *asymmetric RVRP*. We may now have $c_{uv} = 0$, but we may assume that $c_{uv} + c_{vu} > 0$, otherwise we can again merge nodes u and v. Consequently, at most one of (u, v) or (v, u) may lie on a shortest rooted path, and so if $R = 0$, we can again efficiently solve the problem by finding a minimum-cardinality path cover in a DAG. Let O^* denote the optimal value of the given asymmetric RVRP instance. Observe that Lemma 2.2 (as also Lemmas 2.4 and 2.5) continues to hold when c is asymmetric. Thus, we again seek to find $\alpha \cdot O^*$ paths of average regret $\beta \cdot R$, for suitable values of α and β. We show that this can be achieved by utilizing (even) a bicriteria approximation algorithm for kATSPP.

Theorem 6.6 *Suppose we have an algorithm for kATSPP that returns a solution with at most αk s-t paths and cost at most $\beta \cdot OPT_k$. Then, one can achieve an $O(\alpha + \beta)$-approximation for asymmetric RVRP. Thus, the results in [12, 11] yield an $O(\log n)$-approximation for asymmetric RVRP.*

PROOF. Create an auxiliary complete digraph $H = (V_H, A_H)$, where $V_H = \{r\} \cup V \cup \{t\}$. The cost of each arc (u, v) where $u, v \in \{r\} \cup V$ is its regret distance c^{reg}_{uv}; for every $v \in \{r\} \cup V$, the cost of (v, t) is 0 and the cost of (t, v) is ∞. One can verify that these arc costs form an asymmetric metric.

We consider all values k in $1, \ldots, n$ and consider the kATSPP instance specified by H, start node r, and end node t. When $k = O^*$, we know that there is a solution of cost at most $O^* \cdot R$, so using the given algorithm for kATSPP, we obtain at most αO^* $r \rightsquigarrow t$ paths in H of total cost at most $\beta \cdot O^* \cdot R$. So the smallest k for which we obtain at most αk paths of total cost at most $\beta \cdot k \cdot R$ satisfies $k \leq O^*$. Removing t from these (at most) $\alpha k \leq \alpha \cdot O^*$ paths yields a solution in the original metric having total c^{reg}-cost at most $\beta \cdot O^* \cdot R$. by Lemma 2.2, this can be converted to a feasible solution using $O\big((\alpha + \beta) \cdot O^*\big)$ rooted paths.

We can obtain an $O(\log n)$-approximation to OPT_k using (at most) $k \log n$ paths [12], or $2k$ paths (taking $b = 1$ in Theorem 6.4); plugging this in yields an $O(\log n)$-approximation for asymmetric RVRP. □

In Section 7, we prove that an α-approximation for asymmetric RVRP yields a 2α-approximation for ATSP (Theorem 7.2); thus an $\omega(\log \log n)$-factor improvement to the approximation ratio obtained in Theorem 6.6 would improve the state of the art for ATSP.

Non-uniform RVRP. In this broad generalization of RVRP—which captures both multiplicative-RVRP and DVRP—we have non-uniform integer regret bounds $\{R_v\}_{v \in V}$ and we seek the fewest number of rooted paths covering all the nodes where each node v has regret at most R_v. Let $R_{\max} = \max_v R_v$ and $R_{\min} = \min_{v: R_v > 0} R_v$. We apply Lemma 4.1 to the sets $V_0 = \{v : R_v = 0\}$, and $V_i = \{v : 2^{i-1} \leq R_v < 2^i\}$ for $i = 1, \ldots, O(\log R_{\max})$. There are at most $O\big(\log_2(\frac{R_{\max}}{R_{\min}})\big)$ non-empty V_is. Let O^* be the optimal value. We cover V_0 using at most O^* shortest paths, and cover every other V_i-set using $O(O^*)$ paths of regret at most 2^{i-1}. This yields a feasible solution using $O\big(\log(\frac{R_{\max}}{R_{\min}})\big) \cdot O^*$ paths.

We remark that applying the set-cover greedy algorithm yields an $O(\log^2 n)$-approximation, since finding a minimum-density set now amounts to a *deadline TSP* problem [2, 7] for which we only know of an $O(\log n)$-approximation [2].

Capacitated variants. Vehicle-routing problems are often considered in capacitated settings, where we are given a capacity bound C, and a path/route is considered feasible if it contains at most C nodes (and is feasible for the uncapacitated problem). Capacitated additive-kRVRP does not admit any multiplicative approximation in polytime, since it is NP-complete to decide if there is a solution with zero regret [24]. However, when we do not fix the number of paths, a standard reduction [20, 5] shows that an α-approximation to the uncapacitated problem yields an $(\alpha + 1)$-approximation to the capacitated version. This reduction also holds in asymmetric metrics. Thus, we obtain approximation ratios of 31.86 and $O(\log n)$ for capacitated RVRP in symmetric and asymmetric metrics.

7. APPROXIMATION AND INTEGRALITY-GAP LOWER BOUNDS

We now present lower bounds on the inapproximability of RVRP and kRVRP, and the integrality gap of the configuration LPs considered. We obtain both absolute inapproximability results (assuming $P \neq NP$), and results relating the approximability of our problems to that of other well-known problems. A simple reduction from TSP shows the following.

Theorem 7.1 *It is NP-hard to achieve an approximation factor better than 2 for additive- and multiplicative- RVRP. Moreover, no additive approximation is possible in polytime.*

Next, we prove that the approximability of asymmetric RVRP is closely related to that of ATSP; in particular, improving the results in Theorem 6.6 by an $\omega(\log \log n)$-factor would improve the state-of-the-art for ATSP.

Theorem 7.2 *An α-approximation algorithm for RVRP in asymmetric metrics yields a 2α-approximation for ATSP.*

PROOF. Suppose we have an ATSP instance with distances c_{uv} whose optimal value is OPT_{ATSP}. For a given parameter R, the following algorithm will return a solution of cost at most $2\alpha \cdot R$ provided $R \geq OPT_{\text{ATSP}}$. We can then use binary search to find the smallest R for which the algorithm returns a solution of cost at most $2\alpha \cdot R$, and thus obtain an ATSP solution of cost at most $2\alpha \cdot OPT_{\text{ATSP}}$.

Fix any node as the root r. We firs runs the α-approximation for asymmetric RVRP on the RVRP instance specified by the metric c and regret bound R to find a collection of rooted paths $P_1, \ldots, P_k$. Let v_i be the end node of P_i. For each P_i, we add the (v_i, r) arc to obtain an Eulerian graph. The cost of the resulting Eulerian tour is $\sum_{i=1}^{k}(c(P_i) + c_{v_i r})$.

We claim that if $R \geq OPT_{\text{ATSP}}$ then this cost is at most $2\alpha \cdot R$. To see this, note that an optimal solution to the ATSP instance also yields a Hamiltonian path starting at r of cost at most R. Since the regret cost of a rooted path is at most its cost, we infer that the optimum solution to the asymmetric RVRP instance with regret bound R uses only 1 path. Thus, we obtain that $k \leq \alpha$. We know that $c(P_i) \leq D_{v_i} + R$, and $D_{v_i} + c_{v_i r} \leq OPT_{\text{ATSP}}$ for every $i = 1, \ldots, k$. Thus, $\sum_{i=1}^{k}(c(P_i) + c_{v_i r}) \leq \alpha(R + OPT_{\text{ATSP}}) \leq 2\alpha R$. □

Integerality-gap lower bounds. We prove that a natural configuration-style LP-relaxation for kRVRP has an $\Omega(k)$ integrality gap. A common technique used for min-max (or bottleneck) problems is to "guess" the optimal value B, which can often be used to devise stronger relaxations for the problem as well as strengthen the analysis, since B now serves as a lower bound on the optimal value; see, e.g., the algorithms of [18, 26] for unrelated-machine scheduling. We show that this approach does not work for kRVRP. Given a guess R on the maximum regret, similar to (P) and (P2), one can consider the following feasibility problem to determine if there are k rooted paths in $\mathcal{C}_R$ (i.e., the collection of rooted paths with regret at most R) that cover all nodes.

$$\sum_{P \in \mathcal{C}_R : v \in P} x_P \geq 1 \quad \forall v \in V, \qquad \sum_{P \in \mathcal{C}_R} x_P \leq k, \quad x \geq 0. \quad \text{(P3)}$$

Let R_{LP} be the smallest R for which (P3) is feasible, and R^* be the optimal value of the kRVRP instance. We describe instances where $R^* \geq k \cdot R_{\mathrm{LP}}$.

Theorem 7.3 *For any integers $h, c \geq 1$, there is a kRVRP instance with $k = c(2h-1)$ such that $R_{LP} \leq 1$ but any integer solution with maximum regret less than $2h-1$ must use at least $k+c$ rooted paths. Thus, (i) $c = 1$ yields $R^* \geq k \cdot R_{LP}$; (ii) taking $c = h$ shows that one needs $k+c = k+\frac{k}{2h-1}$ paths to achieve maximum regret less than $(2h-1)R_{LP}$.*

PROOF. Our instance will consist of copies of the following "ladder graph" $L_h = (\{r\} \cup V, E)$. We have $V = \{u_1, v_1, u_2, v_2, \ldots, u_{2h-1}, v_{2h-1}\}$. Define $u_0 = r = v_0$. E consists of the edges $\{(u_i, u_{i+1}), (v_i, v_{i+1}) : 0 \leq i < 2h-1\}$, which have cost h, along with edges $\{(u_i, v_i) : 1 \leq i \leq 2h-1\}$, which have unit cost.

Consider the shortest path metric of L_h. Any rooted path that covers all nodes of L_h must have regret at least $2h-1$ (which is achieved by the path $r, u_1, v_1, v_2, u_2, \ldots, u_{2h-1}, v_{2h-1}$). Consider the paths $P_1, \ldots, P_{2h-1}$ given by

$$P_i = \begin{cases} r, u_1, u_2, \ldots, u_i, v_i, v_{i+1}, v_{i+2}, \ldots, v_{2h-1} & \text{if } i \text{ is odd} \\ r, v_1, v_2, \ldots, v_i, u_i, u_{i+1}, u_{i+2}, \ldots, u_{2h-1} & \text{if } i \text{ is even} \end{cases}$$

Each P_i has regret exactly 1 and each node $w \neq r$ lies on h such paths. So setting $x_{P_i} = \frac{1}{h}$ for all $i = 1, \ldots, 2h-1$, and $x_P = 0$ for all other paths in $\mathcal{C}_1$ yields a solution that covers all nodes in V to an extent of 1 using $2 - \frac{1}{h}$ paths.

The final instance consists of ch copies of L_h that share the root r but are otherwise disjoint. We set $k = c(2h-1)$. Taking the above fractional solution for each copy of L_h, yields a feasible solution to (P3) when $R = 1$. Now consider any integer solution with maximum regret less than $2h-1$. Note that any rooted path with regret less than $2h$ can only traverse nodes from a single ladder L_h. Also, as noted above, if a single path covers all the nodes of some copy of L_h, then this path has regret at least $2h-1$. Therefore, the solution must use at least $2ch = k + c$ paths. $\square$

8. REFERENCES

[1] A. Asadpour, M. X. Goemans, A. Madry, S. Oveis Gharan, and A. Saberi. An $O(\frac{\log n}{\log \log n})$-approximation algorithm for the asymmetric traveling salesman problem. In *Proc. SODA*, 2010.

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

[3] N. Bansal and M. Sviridenko. The Santa Claus problem. In *Proc. STOC*, pages 31–40, 2006.

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

[5] A. Bock, E. Grant, J. Könemann, and L. Sanita. The school bus problem in trees. In *Proc. ISAAC*, 2011.

[6] D. Chakrabarty and C. Swamy. Facility location with client latencies: linear-programming based techniques for minimum-latency problems. In *Proc. IPCO*, 2011.

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

[8] C. Chekuri, M. Pál. A recursive greedy algorithm for walks in directed graphs. In *Proc. FOCS*, 2005.

[9] M. Desrochers, J. Desrosiers, and M. Solomon. New optimization algorithm for the vehicle routing problem with time windows. *Oper. Research*, 40:342–354, 1992.

[10] U. Feige. On maximizing welfare when utility functions are subadditive. *SICOMP*, 39:122–142, 2009.

[11] Z. Friggstad. Multiple traveling salesmen in asymmetric metrics. In *Proc. APPROX*, 2013.

[12] Z. Friggstad, M Salavatipour, and Z. Svitkina. Asymmetric traveling salesman path and directed latency problems. In *Proc. SODA*, 2010.

[13] M. X. Goemans and D. P. Williamson Approximating minimum-cost graph problems with spanning tree edges. *Operations Research Letters* 16:183–189, 1994.

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

[15] M. Haimovich and A. Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10:527–542, 1985.

[16] N. Karmarkar and R. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proc. FOCS*, 1982.

[17] G. Laporte, M. Desrochers, and Y. Nobert. Two exact algorithms for the distance constrained vehicle routing problem. *Networks*, 14:47–61, 1984.

[18] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Prog.*, 46:259–271, 1990.

[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. Approximation algorithms for distance constrained vehicle routing problems. Tepper School of Business, CMU, Pittsburgh, 2008.

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

[22] J. Park and B. Kim. The school bus routing problem: A review. *Eur. J. of Oper. Res.*, 202(2):311–319, 2010.

[23] T. Rothvoss. Approximating bin packing within $O(\log OPT \cdot \log\log OPT)$ bins. In *Proc. FOCS*, 2013.

[24] L. Sanita. *Personal communication.*

[25] M. Spada, M. Bierlaire, and T. Liebling. Decision-aiding methodology for the school bus routing and scheduling Problem. *Trans. Sci.*, 39(4):477–490, 2005.

[26] O. Svensson. Santa Claus schedules jobs on unrelated machines. *SICOMP*, 41:1318–1341, 2012.

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