Noname manuscript No.
(will be inserted by the editor)

Minimizing Latency of Capacitated k-Tours*

Christopher S. Martin ·
Mohammad R. Salavatipour

the date of receipt and acceptance should be inserted later

Abstract We study variants of the capacitated vehicle routing problem. In the *multiple depot capacitated k-travelling repairmen problem* (MD-CkTRP), we have a collection of clients to be served by one vehicle in a fleet of k identical vehicles based at given depots. Each client has a given demand that must be satisfied, and each vehicle can carry a total of at most Q demand before it must resupply at its original depot. We wish to route the vehicles in a way that obeys the constraints while minimizing the *average time* (latency) required to serve a client. This generalizes the Multi-depot k-Travelling Repairman Problem (MD-kTRP) [9,17] to the capacitated vehicle setting, and while it has been previously studied [16,18], no approximation algorithm with a proven ratio is known.

We give a 42.49-approximation to this general problem, and refine this constant to 25.49 when clients have unit demands. As far as we are aware, these are the first constant-factor approximations for capacitated vehicle routing problems with a latency objective. We achieve these results by developing a framework allowing us to solve a wider range of latency problems, and crafting various orienteering-style oracles for use in this framework. We also show a simple LP rounding algorithm has a better approximation ratio for the maximum coverage problem with groups (MCG), first studied by Chekuri and Kumar [10], and use it as a subroutine in our framework. Our approximation ratio for MD-CkTRP when restricted to uncapacitated setting matches the best known bound for it [17]. With our framework, any improvements to our oracles or our MCG approximation will result in improved approximations to the corresponding k-TRP problem.

* A preliminary version of this paper appeared in Proceedings of ISAAC 2016.

Supported by Alberta Innovates Technology Futures

Supported by NSERC

Department of Computing Science, University of Alberta, Edmonton, Canada
E-mail: csmartin@ualberta.ca · E-mail: mrs@ualberta.ca

Keywords approximation · k-travelling repairmen · capacitated · latency · group coverage

CR Subject Classification F.2.2 Nonnumerical Algorithms and Problems

1 Introduction

In many vehicle routing scenarios, minimizing response time is a much more important objective than minimizing the distances vehicles travel. Minimizing response time is commonly required in emergency response management, routing package delivery vehicles, school-bus routing, and repairman routing, and is broadly referred to as the *travelling repairman problem* (TRP).

Many variations of this problem have been studied in both the Operations Research and approximation algorithms community. In this paper (like [10]), we consider the following version (and some interesting special cases), which we call the *multiple-depot capacitated k-travelling repairmen problem* (MD-CkTRP). It has also been referred to as the *multiple depot cumulative capacitated vehicle routing problem with multiple trips* [16,18].

We are given a collection of k identical vehicles with capacity Q, that are initially located at k depots (roots) $R = \{r_1, r_2, \dots, r_k\}$, a set of clients $C = \{c_1, c_2, \dots, c_n\}$, a function $w : C \to \mathbb{Z}^{>0}$ specifying the demand of each client, and an undirected metric $d(u, v)$ over the vertices $u, v \in R \cup C$. We must find a routing for the vehicles to serve all clients in C, minimizing the average service time (or *latency*) over all clients in C, subject to the following constraints:

1. Each client must be completely served in one trip (called *unsplit delivery*).
2. Each vehicle can serve a total of at most Q demand, before it must return to its depot to resupply.

We define a *walk* to be a sequence of distinct nodes traversed in a given order, and possibly ending back at the starting node (when a walk does end back at its starting node, we call this a *tour*[1]). A *capacitated walk* is a sequence of 0 or more tours rooted at r_i, followed by an additional walk from r_i, where each tour/walk contains at most Q demand. A sequence of only tours rooted at the same node form a *flower*.

A feasible solution to MD-CkTRP is a collection F_i $(1 \le i \le k)$ of capacitated walks, one for each vehicle that starts at a depot r_i, and where each client c belongs to exactly one F_i. The latency of a client c that belongs to a walk rooted at r_i is the sum of the lengths of the edges traversed by the i'th vehicle before visiting c.

This general problem models many scenarios in package delivery management, where serving clients requires carrying a specific-sized package in a vehicle with limited space. One can further generalize the model to the case

[1] This differs slightly from the typical definition of a tour, since a tour here must be composed of *exactly* one cycle.

where vehicles have non-uniform capacities, and where each client c has a service delay $\delta(c)$, which is added to the latency of c and every client served after c by that vehicle. We call this latter version MD-kTRP with service delays.

Another problem for which we propose a new (improved) approximation algorithm is the Maximum Coverage Problem with Groups (MCG). The MCG appears as a key subroutine in various approximation algorithms, including the framework we develop. The problem is the following: suppose we are given a collection of elements $\mathcal{I}$, a collection of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of $\mathcal{I}$, and a partition of $\mathcal{S}$ into groups $G_1, G_2, \ldots G_\ell$. The objective is to pick a collection of subsets from $\mathcal{S}$ maximizing the size of their union, such that at most one subset from each group is picked.

This problem can be approximated directly via LP-rounding if $|\mathcal{S}|$ is polynomially bounded in $\mathcal{I}$ (*e.g.* with pipage rounding [1]). It is special case of submodular function optimization subject to matroid constraints [6], but in those settings the algorithm has a running time that is polynomial in $|\mathcal{S}|$ while the version we consider can have $|\mathcal{S}|$ exponentially large in $|\mathcal{I}|$; in this case we are instead given an implicit representation of $\mathcal{S}$. In such settings, suppose we were given an oracle $\mathcal{A}(i, \theta)$ that takes as input a group index i and a weight function $\theta : \mathcal{I} \to \{0, 1\}$, and returns some subset $S_j \in G_i$ such that $\sum_{e \in S_j} \theta(e)$ is maximized. Call $\mathcal{A}$ a $(1/\rho)$-approximate oracle if it returns a subset $S_j \in G_i$ such that $\sum_{e \in S_j} \theta(e) \geq \frac{1}{\rho} \max_{S' \in G_i} \sum_{e \in S'} \theta(e)$ (*i.e.* the returned subset covers at least a $\frac{1}{\rho}$-fraction of the optimal number of elements). In this paper, we focus on approximating MCG for which $|\mathcal{S}|$ can be exponential in $|\mathcal{I}|$ and we have access to an oracle $\mathcal{A}$ as above; this version will be useful in our approximation algorithms for MD-CkTRP. We will therefore never state $\mathcal{S}$ explicitly, instead only giving the oracle $\mathcal{A}$ and the groups G_i defining the input instance.

For the approximation we develop, we will require a *weighted* version of $\mathcal{A}$; that is, the input θ will instead be a function returning any non-negative value. Many oracles (including the ones we present) can be converted to this form with only a small loss in approximation using standard techniques, such as scaling weights and duplicating elements.

1.1 Related Work

The special case of $k = 1$ and $Q = \infty$ for the MD-CkTRP is the Minimum Latency or Travelling Repairman problem, which has been studied extensively [4,3,9,13,20]. This case is known to be APX-hard in general metrics [5], and the 3.59-approximation of Chaudhuri *et al.* [9] is the best known for this case. The special case where the metric is an edge-weighted tree is also known to be NP-hard [19], and a PTAS for this was only recently found [20].

For the uncapacitated k-vehicle situation where $r_1 = r_2 = \cdots = r_k$ and $Q = \infty$, an 8.497-approximation was known [11]; this was recently improved to 7.183 [17]. For the multi-depot uncapacitated case, Chekuri and Kumar [10] proved a 24-approximation.[2] This was recently improved to 8.497 by Post and Swamy [17]. This improvement came from using a time-index configuration LP that was introduced in [8] for the single vehicle case, while extending it to the multi-vehicle setting and introducing an LP rounding algorithm.

The MCG was first considered by Chekuri and Kumar [10] in the context of their approximation for the MD-kTRP. They developed the first approximation for the problem given a $(1/\rho)$-approximate oracle, obtaining a simple greedy $1/(\rho+1)$-approximation. The submodular maximization problem with matroid constraints generalizes MCG: the instance can be represented by a monotone submodular function $f(S)$ denoting the number of elements covered by the set S, and a partition matroid $\mathcal{M}$ over the sets in $\mathcal{S}$ that define the groups. It was shown in [6] how to obtain a $(1-1/e)$-approximation for this problem with running time polynomial in $|\mathcal{S}|$ and $|\mathcal{I}|$. When $\mathcal{S}$ is not given explicitly and $|\mathcal{S}|$ is exponentially large in $|\mathcal{I}|$, the result of Chekuri and Kumar [10] is currently the best known.

To the best of our knowledge, *no* approximation algorithm for any capacitated variant of the travelling repairmen problem has been developed. Our specific problem has been studied in the operations research community, but only heuristic solutions are currently known [16, 18].

1.2 Our Results

We solve the capacitated variant of the travelling repairmen problem by building off of and extending the techniques used previously for the multi-depot travelling repairmen problem and for capacitated vehicle routing. Our algorithm uses ideas from both [10] and [17], in particular the greedy combinatorial algorithm of [10], coupled with a new LP-based approximation algorithm for the MCG inspired by [17]. One feature of our algorithm is that if we restrict it to the case of $Q=\infty$ (*i.e.* the uncapacitated setting), we obtain an approximation ratio matching the best known bound for that setting [17].

To achieve this, we develop a modular framework (Theorem 4) that uses a user-provided oracle as a subroutine to solve different versions of the multi-depot travelling repairmen problem. The exact problem we solve is captured by a collection of *feasible walks*, which are separated over using the provided oracle as a black-box. Given such an oracle, we can build $O(1)$-approximation algorithms for the various latency problems we consider. We obtain the following results with this approach:

Theorem 1 *There is a* 25.49*-approximation to the unit-demand capacitated multi-depot k-TRP.*

Theorem 2 *There is a* 42.49*-approximation to the unsplit-delivery capacitated multi-depot k-TRP.*

[2] The approximation ratio was stated in [10] to be 12, but due to a technical issue in their analysis, they were off by a factor of 2. We provide the corrected analysis in the appendix.

We show a simple LP-rounding gives an improved approximation for MCG, which we use as a subroutine in our framework:

Theorem 3 *There is a* $(1-e^{-1/\rho})$*-approximation to the MCG given a* $(1/\rho)$*-approximate oracle.*

Theorems 1 and 2 are the first (constant) approximations for the MD-CkTRP, and also extend to more general cases where we have non-uniform vehicles capacities. We may additionally add service delays $\delta(c)$ at each client with an extra $+0.5$ loss in the ratio. These extensions are covered in Section 5. The framework we develop to prove Theorems 1 and 2 is presented as Theorem 4. The algorithm we give to prove that theorem finds progressively longer rooted *flowers* from each depot that cover a large number of clients, where the length of these flowers is bounded against a rooted *walk*. Suppose that C is the set of clients to be served/covered by a walk from r_i, and B is a given budget on the length of the walk (depending on our problem, walks might be capacitated). The *single-depot orienteering problem* (SD-OP) is to find such a walk with total cost at most B starting at r_i that covers as many (distinct) clients of C as possible.

We can generalize the notion of capacitated walks/tours by giving a set $\mathcal{W}_i$ that contains all r_i-rooted walks vehicle i is allowed to traverse for the given problem. A capacitated walk/tour is then a walk/flower built using only walks from $\mathcal{W}_i$. We call these $\mathcal{W}_i$*-restricted walks/flowers*. Our approach centres around a black-box algorithm to (approximately) solve the SD-OP problem over the set of walks $\mathcal{W}_i$; that is, only walks in $\mathcal{W}_i$ are considered feasible for vehicle i.

Definition 1 A $(1/\rho, \gamma)$-approximation to the $\mathcal{W}_i$-restricted SD-OP problem is an algorithm that finds a walk of cost $\leq \gamma B$ covering at least a $1/\rho$-fraction of the number of clients on an optimal walk.

If this black-box returns a flower rather than a walk, but with cost still bounded by the optimal walk, then we call this a $(1/\rho, \gamma)$-*flower* approximation. We use this algorithm as an oracle to find interesting walks/flowers over the sets $\mathcal{W}_i$ defined by the problem. With this, we obtain the following result:

Theorem 4 *Let* $\mathcal{W}_i$ *be the set of all* r_i*-rooted walks that can be feasibly traversed by vehicle* i*. Then for constants* ρ, γ*, there is an* $f(\rho, \gamma)$*-approximation algorithm to the* $\mathcal{W}_i$*-restricted multi-depot* k*-TRP, if we have a* $\mathcal{W}_i$*-restricted* $(1/\rho, \gamma)$*-flower approximation to the* $\mathcal{W}_i$*-restricted SD-OP, where* $f(\rho, \gamma) = \frac{\gamma(\tau+1)(1-e^{-1/\rho})}{2\ln(\tau)(1-\tau e^{-1/\rho})}$ *for any constant* $1 < \tau < e^{1/\rho}$*.*

When $\mathcal{W}_i$ is the set of *all* possible walks from depot r_i, we are solving the uncapacitated multi-depot k-TRP (MD-kTRP), studied in [10,17]. If we restrict $\mathcal{W}_i$ to only capacitated walks (with capacity Q), we are solving the unsplit MD-CkTRP variant. Using Theorem 4, we can find a constant-factor approximation to the MD-CkTRP given an oracle satisfying definition 1 that returns flowers. We give oracles for the unit-demand and unsplit-delivery cases in Section 4, which when combined with Theorem 4 yields Theorems 1 and 2.

We start by proving Theorem 3 in Section 2. We then proceed to prove Theorem 4 in Section 3, by showing how to combine ideas from [10], [9], and [17] to create a combinatorial approximation algorithm for the problem, which requires solving an MCG instance as a subroutine.

An alternative approach for approximating latency problems that avoids explicitly solving an MCG instance was introduced and expanded in [8,17]. They solve a time-indexed configuration LP directly for the multi-depot latency problem, and use randomized rounding to obtain the final collection of tours. Our approach is in fact equivalent to theirs for that specific problem; the combination of our greedy algorithm and MCG LP yields their time-indexed LP. By writing the configuration LP for a more general covering problem (namely MCG) and using that as a subroutine in our latency algorithm we feel that the approach becomes more easily adaptable to different problems beyond latency. In a sense, we unify and generalize the combinatorial algorithm of [10] and the LP rounding algorithm of [17] in a framework using MCG rounding.

2 A $(1 - e^{-1/\rho})$-Approximation for MCG

We can express an instance of the MCG as an integer configuration program. For item $e \in \mathcal{I}$ and group G_i, let x_e be a binary variable indicating whether item e is being covered by a set or not. For a set $S \in \mathcal{S}$, let z_S be a binary variable indicating whether set S is chosen to form a part of the solution. The linear relaxation of the configuration program is given as (LP) (and its dual as (DP)).

$$\begin{aligned}
\max \quad & \sum_e x_e && && \text{(LP)} \\
\text{s.t.} \quad & x_e \le 1 && \forall e \quad (\alpha_e) && (1) \\
& \sum_{S \in G_i} z_S \le 1 && \forall i \quad (\beta_i) && (2) \\
& \sum_{S : S \ni e} z_S \ge x_e && \forall e \quad (\theta_e) && (3) \\
& x, z \ge 0. && &&
\end{aligned}$$

$$\begin{aligned}
\min \quad & \sum_e \alpha_e + \sum_i \beta_i && && \text{(DP)} \\
\text{s.t.} \quad & \alpha_e + \theta_e \ge 1 && \forall e && (4) \\
& \sum_{e \in S} \theta_e \le \beta_i && \forall i, S \in G_i && (5) \\
& \alpha, \beta, \theta \ge 0. && &&
\end{aligned}$$

For every set $S \in G_i$ we use $\theta(S)$ to denote $\sum_{e \in S} \theta_e$. As stated before, we assume we are given a approximate *weighted* oracle $\mathcal{A}(i, \theta)$; that is, for each group i, given θ_e on elements it will find a set S in group G_i such that $\theta(S) \ge 1/\rho \max_{S' \in G_i} \theta(S')$.

$\mathcal{A}$ will become our approximate separation oracle for the dual. Solving an exponential size LP approximately using such an oracle is a standard technique following from the work of Carr and Vempala [7]. We briefly describe how to obtain a good solution following the more recent presentation in [12].

Define the polytope $\mathcal{P}(\upsilon, a) = \{(\alpha, \beta, \theta) : (4), (5), \sum_e \alpha_e + a \sum_i \beta_i \leq \upsilon\}$. With our ρ-approximate (weighted) oracle $\mathcal{A}$, given some υ and point (α, β, θ), we can certify that either $(\alpha, \rho\beta, \theta) \in \mathcal{P}(\upsilon, 1)$, or give a hyperplane certifying that $(\alpha, \beta, \theta) \notin \mathcal{P}(\upsilon, \rho)$, as follows. For each i, run $\mathcal{A}$ with element weights θ_e. If the returned set S has weight $\theta(S) > \beta_i$, then since $\theta(S) \geq (1/\rho) \max_{S' \in G_i} \theta(S')$, we return the constraint (5) corresponding to i, S as the separating hyperplane. The other constraints can be checked trivially. If no constraint is violated, we must have $(\alpha, \rho\beta, \theta) \in \mathcal{P}(\upsilon, 1)$, and so the ellipsoid algorithm will certify in polynomial time that either $\mathcal{P}(\upsilon, \rho) = \emptyset$, or give a point $(\alpha, \rho\beta, \theta) \in \mathcal{P}(\upsilon, 1)$. Note that $\mathcal{P}(OPT_{LP}, 1)$ defines the collection of optimum solutions for (DP), and so OPT_{LP} is the smallest υ such that $\mathcal{P}(\upsilon, 1) \neq \emptyset$; we can determine this value by binary search on υ.

Suppose we run the ellipsoid algorithm with input $OPT_{LP} - \epsilon$ for any $\epsilon > 0$. This yields a certificate showing $\mathcal{P}(OPT_{LP} - \epsilon, \rho) = \emptyset$, consisting of polynomially-many separating hyperplanes, including the inequality $\sum_e \alpha_e + \rho \sum_i \beta_i \leq OPT_{LP} - \epsilon$. Consider the dual polytope of $\mathcal{P}(\upsilon, a)$: $\mathcal{Q}(\upsilon, a) = \{(x, z) : (1), \sum_{S \in G_i} z_S \leq a, (3), \sum_e x_e \geq \upsilon\}$. By duality, the certificate corresponds to a point $(x, z) \in \mathcal{Q}(OPT_{LP} - \epsilon, \rho)$ with polynomially-many non-zero variables. Note that $(x/\rho, z/\rho)$ is a feasible (approximate) solution to (LP); further, (x, z) is *almost* a feasible solution with objective value $OPT_{LP} - \epsilon$ that only violates (2).[3] This property will be crucial to our rounding scheme.

2.1 Pipage Rounding

There are many ways to round a solution to a linear program; *pipage rounding* is one technique first introduced by Ageev and Sviridenko [1]. This is a general rounding scheme that works with linear programs of a specific form. We briefly describe this approach, before showing how to apply it to round our approximate solution while maintaining a good approximation ratio.

We wish to approximately solve a generalized version of the following bipartite matching problem: given a bipartite graph $H = (U \cup W, E)$, vertex capacities p_v, and a poly-time computable function $F(x)$ defined over the vectors $x = (x_e : e \in E)$, $x_e \in [0, 1]$, pick a collection of edges from E such that each vertex v has at most p_v edges incident with it, maximizing the value of $F(x)$. If we pick edge e for our collection, then we set $x_e = 1$, and $x_e = 0$ otherwise. Note that with $p_v = 1$ for all v and $F(x) = \sum_{e \in E} x_e$, this becomes the maximum bipartite matching problem.

[3] We omit the ϵ for the remainder of this discussion for clarity.

This general problem can be expressed as an integer program. The relaxed version, where a solution may be a rational vector, is as follows:

$$\begin{aligned} \max\ & F(x) && \text{(PIPE-LP)} \\ \text{s.t.}\ & \sum_{e \in \delta(v)} x_e \leq p_v \quad \forall v \in (U \cup W) && (6) \\ & x \in [0,1]. \end{aligned}$$

Note that $F(x)$ may not be a linear function, so as written this may not be a linear program and so may not be solvable using standard techniques. For our purposes however, will assume that some fractional solution x has been provided.

Let F^* be the value of an optimal *integer* solution to (PIPE-LP), and x a fractional solution to (PIPE-LP). The pipage rounding algorithm transforms the solution x into an integral solution $\bar{x}$ which, given some conditions on F, will have the property that $F(\bar{x}) \geq F(x)$. If, for an optimal fractional solution $\check{x}$, we also had $F(\check{x}) \geq F^*/\alpha$, then we would have an α-approximate solution to the original problem.

The algorithm. The pipage rounding algorithm is an iterative procedure, where in each step we convert a fractional solution x to a new solution x' with at least one less fractional component. The algorithm terminates when $x = \bar{x}$ is integral.

In each step, if we do not terminate, then x has some non-integral entry. Consider the bipartite subgraph H_x of H, where edge $e \in E$ is in H_x if and only if x_e is non-integral. Let R be a cycle in H_x, or, if no cycle exists, a path whose endpoints have degree 1 in H_x. In either case, since H_x is bipartite the cycle/path R can be represented as the union of two matchings M_1 and M_2. We will compute a new solution $x(\epsilon, R)$ using these matchings; if $e \in M_1$, then $x_e(\epsilon, R) = x_e + \epsilon$; otherwise if $e \in M_2$, then $x_e(\epsilon, R) = x_e - \epsilon$; otherwise $x_e(\epsilon, R) = x_e$.

Let ϵ_1 be the smallest ϵ we can subtract such that some $e \in M_1$ becomes 0 or $e \in M_2$ becomes 1, and let ϵ_2 be the smallest ϵ we can add such that some $e \in M_1$ becomes 1 or $e \in M_2$ becomes 0. Let $x_1 = x(-\epsilon_1, R)$, and $x_2 = x(\epsilon_2, R)$. Set $x' = x_1$ if $F(x_1) > F(x_2)$, and $x' = x_2$ otherwise. This concludes one iteration of the algorithm.

In order for a solution returned by this algorithm to have the property that $F(\bar{x}) \geq F(x)$, we require that in each step $F(x') \geq F(x)$. This latter inequality holds when $F(x(\epsilon, R))$, for $\epsilon \in [-\epsilon_1, \epsilon_2]$, is maximized at either endpoint of the interval. We call this the *ϵ-convexity condition.*[4]

Definition 2 The function F is *ϵ-convex* if, for any step of the pipage rounding algorithm and for $\epsilon \in [-\epsilon_1, \epsilon_2]$, $F(x(\epsilon, R))$ is maximized at either $-\epsilon_1$ or ϵ_2.

[4] Note that any F that is a linear function of x satisfies this condition.

Bounding the integrality gap. To bound the integrality gap of (PIPE-LP) for an arbitrary F, we can use the following technique. Suppose we are given a second poly-time computable function $L(x)$, defined over the same set of vectors x as F, and where the following conditions hold:

Condition 1 *For binary x, $L(x) = F(x)$.*

Condition 2 *For any optimal fractional solution $\check{x}$, $F(\check{x}) \geq L(\check{x})/\alpha$.*

These are called the *F/L lower bound conditions*. If (PIPE-LP) is poly-time solvable when the objective is to maximize $L(x)$ instead of $F(x)$, then since $L(\check{x}) \geq F^*$ (by condition 1), by condition 2 we would then have $F(\check{x}) \geq F^*/\alpha$, as desired.

2.2 Rounding a Solution to MCG

To round the solution $(x/\rho, z/\rho)$, we apply pipage rounding by adapting the ideas used by Ageev and Svirendenko [1] for proving an integrality gap for the standard Maximum Coverage problem. Observe that (LP) is equivalent to the following linear program:

$$\max \sum_e \min\left(1, \sum_{S \ni e} z_S\right) \tag{LP2}$$

$$\text{s.t.} \sum_{S \in G_i} z_S \leq 1 \qquad \forall i \tag{7}$$

$$z_S \in [0,1].$$

Constraints (1) and (3) have been rewritten as the minimum in the objective, and so given a fractional solution $(x/\rho, z/\rho)$ to (LP), we can obtain a solution z/ρ to (LP2) of equal objective value (*i.e.* at least OPT_{LP}/ρ). (LP2) is now in pipage rounding form as described previously, and so we can apply the pipage rounding algorithm to obtain an integer solution $\bar{z}$.

We now bound the integrality gap. Let $L(z) = \sum_e \min(1, \sum_{S \ni e} z_S)$. We will define a function $F(z)$ that is both ϵ-convex on the input z/ρ and satisfies the F/L lower bound conditions. Suppose that, for an optimal (fractional) solution $\check{z}$, our sub-optimal solution z/ρ has the property that $F(z/\rho) \geq L(\check{z})/\alpha$ for some α. Let F^* be the value of an optimal *integer* solution to (LP2); if L and F are coincident on binary inputs, then $L(\check{z}) \geq F^*$, and so this new condition would imply we have an α-approximation after pipage rounding. We claim that the function $F(z) = \sum_e (1 - \prod_{S \ni e}(1 - z_S))$ satisfies these conditions.

Lemma 1 *$F(z)$ satisfies the F/L lower bound conditions.*

Proof Clearly for binary z, $L(z) = F(z)$. We now show some α exists such that for any optimal solution $\check{z}$, $F(z/\rho) \geq L(\check{z})/\alpha$. Let n_e be the number of sets in

$\mathcal{S}$ which contain e. Using the arithmetic-geometric mean inequality, and the fact that the solution z only violates constraints (2),

$$\begin{aligned} 1 - \prod_{S \ni e}(1 - z_S/\rho) &\geq 1 - \left(1 - \frac{1}{\rho n_e}\min(1, \sum_{S \ni e} z_S)\right)^{n_e} \\ &\geq 1 - \left(1 - \frac{1}{\rho n_e}\right)^{n_e}\min(1, \sum_{S \ni e} z_S) \\ &\geq (1 - e^{-1/\rho})\min(1, \sum_{S \ni e} z_S). \end{aligned}$$

Thus, $F(z/\rho) \geq (1 - e^{-1/\rho})L(z)$. But since $L(z) \geq OPT_{LP}$, then $L(z) \geq L(\check{z})$, and so $F(z/\rho) \geq (1 - e^{-1/\rho})L(\check{z})$. □

Lemma 2 *$F(z/\rho)$ is ϵ-convex.*

Proof Since the groups G_i define a partition over $\mathcal{S}$, the bipartite graph used during the pipage rounding algorithm is in fact a forest, with each tree having height 1. This implies that in each step of the pipage rounding algorithm, the chosen R must be a path of length at most 2. Rewriting $F((z/\rho)(\epsilon, R))$ as a function of ϵ, we then have either a linear or quadratic function. Since for all e, $z_e/\rho \in [0, 1]$, then this quadratic will have a non-negative main term, and so $F(z/\rho)$ is ϵ-convex. □

We can therefore apply pipage rounding on the fractional solution z/ρ, using the function F to guide the algorithm. This yields a deterministic $(1 - e^{-1/\rho})$-approximation to the MCG problem.

3 Proof of Theorem 4

We present the framework by generalizing and modifying the combinatorial algorithm of Chekuri and Kumar [10] to suit our redefined problem. The key subroutine of their algorithm is an approximation for the MCG, which they use to determine a set of tours to "stitch" together for routing vehicles from each depot. Their algorithm uses an oracle as a black-box to solve an orienteering-style problem in order to find "good" tours to use in their MCG instance. In [10] these tours are built from an ℓ-MST, using the algorithm from [9]; we will instead use the user-provided black-box oracle for this task and show that we still obtain a good approximation.

Recall we are given as input a set of clients C, a set of k depots R, a vehicle initially located at each depot, and a metric distance function d. We wish to find $\mathcal{W}_i$-restricted walks for each vehicle i starting at their respective depots that collectively visit all clients, and minimize the total latency of all walks. The latency of a walk W that starts at root r and visits clients $c_1, c_2, \ldots c_m$ is given by $\sum_{i=1}^{m} d_W(r, c_i)$, where d_W is the distance along the walk between two points.

The computation is split up into *phases*, with each phase given a budget with which to cover as many clients as possible. The latency of the clients we cover in this phase can then be bounded by the total budget we have spent in this phase and all prior phases. Let $j \geq 1$ be the current phase, and let C_j^u be the set of uncovered clients at the start of phase j. Let $\tau > 1$ be some global constant to be chosen later, $U \in [0, 1)$ be a number chosen uniform randomly, and $b = \tau^U$.

We define the *multi-depot group orienteering problem* (MD-GOP) as follows: given a subset of clients C' to be visited and a hard budget B, find for each depot $r_i \in R$ a *walk* of total length at most B such that all walks returned collectively cover as many (distinct) clients in C' as possible. We define $\mathcal{C}(C', B)$ to be some algorithm that solves the $\mathcal{W}_i$-restricted version of this problem approximately. $\mathcal{C}$ is a $\mathcal{W}_i$-restricted $(1/\rho, \gamma)$-flower approximation if it finds a collection of k *flowers* rooted at the depots r_i, such that each costs at most γB and together they cover at least a $\frac{1}{\rho}$-fraction of the vertices covered by an optimum MD-GOP (walk) solution. Note that for the case of uncapacitated vehicles, a flower is simply a single tour. Given this subroutine, the algorithm for phase j is as follows:

function Do-Phase(j)
Run $\mathcal{C}(C_j^u, b\tau^j)$ with clients C_j^u and budget $b\tau^j$.
Traverse the returned flower for each r_i in either direction, chosen uniformly at random.
Remove all covered clients from C_j^u.
end function

We build a bi-criteria $(1-e^{-1/\rho}, \gamma)$-flower approximation algorithm $\mathcal{C}$, given a user-provided oracle $\mathcal{A}$ as per the Theorem, using our MCG approximation (Theorem 3). Let S_W be the set of vertices contained in the walk W. Let $\mathcal{W}_i^B$ be the set of walks in $\mathcal{W}_i$ of length at most B. Let $G_i = \{S_W : W \in \mathcal{W}_i^B\}$ be the group of all $\mathcal{W}_i$-restricted walks of total length at most B. This forms a valid MCG instance, whose solution yields a collection of k walks, each of cost at most B that collectively cover as many clients as possible.

This instance can be approximately solved as follows. Using $\mathcal{A}$, we can find flowers in G_i covering as many new clients as possible, relative to the optimal walk. Since $\mathcal{A}$ finds a flower covering at least a $1/\rho$-fraction of the optimal number of new clients, by Theorem 3 the final solution covers a $(1 - e^{-1/\rho})$-fraction of the optimal number of clients, exceeding the budget for each flower by a factor of γ. Thus, $\mathcal{C}$ is a $(1 - e^{-1/\rho}, \gamma)$-flower approximation.

3.1 Analysis

We now prove that we have a constant-factor approximation to the $\mathcal{W}_i$-restricted multi-depot k-TRP, thus completing the proof of Theorem 4. Fix an optimal solution OPT, and let O_j denote the set of clients in OPT that have latency $\leq b\tau^j$. Let C_j^v be the clients we have visited by the *end* of phase j. We define C_0^v to be the empty set.

Lemma 3 *At the end of phase j, we have covered at least $(1 - e^{-1/\rho})|O_j - C^v_{j-1}|$ clients.*

Proof Let A_j be all clients covered after phase j, and let R_j denote $O_j - C^v_{j-1}$. Note that at stage j there is a collection of paths (each of lengths at most $b\tau^j$ rooted at some depot) that that can cover R_j. Thus by Theorem 3 and using our approximate oracle $\mathcal{A}$, we have $|A_j| \geq (1 - e^{-1/\rho})|R_j|$, yielding the lemma. □

Let n_j^{OPT} be the number of clients in OPT whose latency is *more* than $b\tau^j$, and let n_j be the number of clients that were left uncovered at the end of phase j. For $j \leq 0$, we define n_j^{OPT} and n_j to be n. Let B_j be the budget of phase j; for $j \geq 1$ this is $b\tau^j$, and for $j \leq 0$ we define it to be 0. For notational convenience, define $\Delta_j = B_j - B_{j-1}$.

Lemma 4 *For all j, $n_j \leq e^{-1/\rho} n_{j-1} + (1 - e^{-1/\rho}) n_j^{OPT}$.*

Proof From Lemma 3, it follows that $n_j \leq n_{j-1} - (1 - e^{-1/\rho})|O_j - C^v_{j-1}|$. Since $|O_j| = n - n_j^{OPT}$ and $|C^v_{j-1}| = n - n_{j-1}$, we can expand this expression and derive the result. □

Lemma 5 *In expectation, the latency of our solution is at most:*

$$\frac{\gamma(\tau+1)}{2(\tau-1)} \sum_{j \geq 1} B_j (n_{j-1} - n_j) = \frac{\gamma(\tau+1)}{2(\tau-1)} \sum_{j \geq 1} n_{j-1} \Delta_j. \qquad \text{(OUR-UB)}$$

Proof For a client c covered in phase j by vehicle i, its latency in our solution is at most the sum of the lengths of all flowers chosen for vehicle i in rounds $1 \leq j' \leq j$. Note however that the last flower is traversed in a random direction, so in expectation the additional latency c incurs in round j is 1/2 the total length of the flower picked in round j. This implies the expected latency of c is at most

$$\frac{\gamma b \tau^j}{2} + \sum_{j'=1}^{j-1} \gamma b \tau^{j'} \leq \frac{\gamma(\tau+1)}{2(\tau-1)} b\tau^j.$$

We can now upper-bound the total expected latency of all clients in our solution by

$$\frac{\gamma(\tau+1)}{2(\tau-1)} \sum_{j \geq 1} B_j (n_{j-1} - n_j) = \frac{\gamma(\tau+1)}{2(\tau-1)} \sum_{j \geq 1} n_{j-1} \Delta_j. \qquad \text{(OUR-UB)}$$

We obtain the right-hand side by re-arranging the summation, noting that $\Delta_1 = B_1$ and $n_0 = n$. □

Lemma 6 *In expectation, the latency of OPT is at least:*

$$\frac{\ln \tau}{\tau - 1} \sum_{j \geq 1} n_{j-1}^{OPT} \Delta_j. \qquad \text{(OPT-LB)}$$

Proof Since n_j^{OPT} is the number of clients with latency $> b\tau^j$, we can initially lower-bound the total latency of OPT with

$$\sum_{j\geq 0} b\tau^j (n_j^{OPT} - n_{j+1}^{OPT}) = \frac{1}{\tau}\sum_{j\geq 1} n_{j-1}^{OPT}\Delta_j. \tag{8}$$

The left-hand side is derived by rounding the latency of each client down to the nearest $b\tau^j$.[5] The right-hand side is obtained by rearranging the summation. We can improve this bound using the random value $b = \tau^U$: suppose client c is visited in OPT with latency $lat_c^{OPT} = d\tau^j$, where $1 \leq d < \tau$. If we choose U such that $b \leq d$, then c's latency will be rounded on the left-hand side of (8) down to $b\tau^j$. Otherwise, $b > d$, so the latency of c will be rounded down to $b\tau^{j-1}$. Over all uniform-random choices of U, the expected latency of c in our solution due to this rounding is

$$\begin{aligned}\int_0^{\log_\tau d} b\tau^j dU + \int_{\log_\tau d}^1 b\tau^{j-1} dU &= \tau^{j-1}\left(\tau\int_0^{\log_\tau d}\tau^U dU + \int_{\log_\tau d}^1 \tau^U dU\right)\\ &= d\tau^{j-1}\left(\frac{\tau-1}{\ln\tau}\right)\\ &= lat_c^{OPT}\cdot\frac{\tau-1}{\tau\ln\tau}.\end{aligned}$$

In expectation then, the rounding in (8) is a factor $\frac{\tau\ln\tau}{\tau-1}$ away from OPT, and so in expectation over all choices of U the following is also a lower bound on OPT:

$$\frac{\ln\tau}{\tau-1}\sum_{j\geq 1} n_{j-1}^{OPT}\Delta_j. \tag{OPT-LB}$$

□

Proof (Proof of Theorem 4) By summing Lemma 4 over all j, we see that

$$\begin{aligned}\sum_{j\geq 1}\Delta_j n_{j-1} &\leq e^{-1/\rho}\left(\sum_{j\geq 1}\Delta_j n_{j-2} + (e^{1/\rho}-1)\sum_{j\geq 1}\Delta_j n_{j-1}^{OPT}\right)\\ &= \tau e^{-1/\rho}\sum_{j\geq 1}\Delta_j n_{j-1} + \frac{(1-e^{-1/\rho})(\tau-1)}{\ln\tau}\frac{\ln\tau}{\tau-1}\sum_{j\geq 1}\Delta_j n_{j-1}^{OPT}\\ \implies \text{(OUR-UB)} &\leq \frac{\gamma(\tau+1)(1-e^{-1/\rho})}{2\ln(\tau)(1-\tau e^{-1/\rho})}\text{(OPT-LB)}.\end{aligned}$$

Our algorithm is therefore a $\frac{\gamma(\tau+1)(1-e^{-1/\rho})}{2\ln(\tau)(1-\tau e^{-1/\rho})}$-approximation for any constant $1 < \tau < e^{1/\rho}$, satisfying the requirements of the theorem. □

[5] There is an annoying detail hidden here - this lower bound only holds if no client has latency less than b in OPT. However, by scaling distances we can ensure this is always the case.

3.2 An Uncapacitated Oracle

We now give a $(1, 2+\epsilon)$-approximate oracle $\mathcal{A}$ for the uncapacitated multi-depot k-TRP (*i.e.* $\mathcal{W}_i$ is *all* possible walks from r_i). This oracle is used in [10] and earlier works for single-depot latency problems. First we describe an unweighted oracle (*i.e.* each node is assigned $\theta_e \in \{0,1\}$); we later describe how to extend it to the weighted version.

Using the algorithm in [9] for finding an ℓ-MST, we find a tree that covers at least as many clients as the optimal r_i-rooted walk with budget B, and costs at most $(1+\epsilon)$ times the optimal walk (see Theorem 1 in [9]). Since the optimal walk costs at most B, we find the largest ℓ such that the returned ℓ-MST has cost at most $(1+\epsilon)B$. Such a tree will cover at least as many clients as the optimal walk. Double the edges of this tree, and convert to a tour by shortcutting past repeated vertices.

For the case that we have weights on the nodes, at a loss of at most $1-\epsilon'$ on the total weight of nodes we can cover, we can reduce the problem to the unweighted case by scaling and discarding nodes with very small weight (so that $\frac{\max_e \theta_e}{\min_{e'} \theta_{e'}} \in O(n^2)$) and then duplicating vertices. This gives a $(1-\epsilon', 2+\epsilon)$-approximate (weighted) oracle $\mathcal{A}$ (for any $\epsilon, \epsilon' > 0$).

This leads to the following result for the uncapacitated MD-kTRP, which matches the current-best given in [17].

Corollary 1 *There is an* 8.497*-approximation to the uncapacitated multiple depot k-TRP* ($\tau \approx 1.405$)*.*

4 Capacitated Oracles and Proofs of Thms. 1 and 2

Previously, we showed that to solve the MD-CkTRP, we can use Theorem 4 and restrict $\mathcal{W}_i$ to only capacitated r_i-rooted walks. We thus need to find an oracle that can solve the related orienteering problem over this set of walks. Using standard techniques as before, we can reduce the weighted version of the problem (with weights on the nodes) to the unweighted version, which we present below.

The problem the oracle must solve is the following, which we call the *unsplit capacitated orienteering problem* (U-COP). We are given a collection of clients C, a root node r, a budget B, a vehicle capacity Q, a client demand function $w : C \to \mathbb{Z}^{>0}$, and an undirected distance metric d. We wish to find an r-rooted capacitated walk of total length at most B, where r must be re-visited after serving at most Q client demand, and we wish to cover as many clients as possible. Call the optimal number of clients ℓ^{OPT}, and let $d(W)$ denote the length of the walk W with respect to the metric d, and similarly for flowers and tours.

We give a $(1, 10+\epsilon)$-flower approximation algorithm, where the flower we find has total cost at most $(10+\epsilon)B$ and collectively covers ℓ^{OPT} clients, respecting the capacity constraint. We also consider a special case where

$w(c) = 1$ for all $c \in C'$; we call this the *unit-demand capacitated orienteering problem* (1-COP). With this demand constraint, we can improve the above ratio to $(1, 6 + \epsilon)$.

An optimum solution to either problem consists of a sequence of tours (each visiting at most Q demands) followed by at most one walk of total demand at most Q. If we convert that last walk to a tour by returning to the root, we obtain a capacitated flower of cost at most $2B$. We will restrict our attention to finding such flowers.

The algorithms for 1-COP and U-COP are very similar, so we describe both simultaneously. If there is a difference between the two algorithms, we place the difference for U-COP in (parentheses). It will be useful to consider the input metric as the complete graph $G = (V, E)$ with $V = C \cup \{r\}$ and edge costs $c_G(uv) = d(u, v)$ for all $uv \in E$.

1. Let G^* be a new graph obtained from G by adding a "terminal" client c' to each $c \in C$ and edge cc' between client c and its new terminal client; the cost of these new edges will be $\frac{1}{Q}d(r, c)w(c)$ (for U-COP, use $\frac{2}{Q}d(r, c)w(c)$). Let G^T be the "terminal" graph obtained from the metric completion of G^*, with all non-terminal client vertices removed.
2. Using the ℓ-MST approximation of Chaudhuri *et al.* [9], find a tree of cost at most $3B + \epsilon$ $(5B + \epsilon)$ in G_T that covers as many terminals as possible. Doubling this tree produces a tour; call this tour O.
3. Convert O back into a tour in G^* that visits the same number of terminal clients of no greater cost (always possible since G^T is the metric completion of G^*). Prune away the terminals and short-cut to obtain a new tour O' in G.
4. Let G' be a complete graph containing r and $w(c)$ copies of each client $c \in C$; let Ω_c denote the copies of c in G' (so $|\Omega_c| = w(c)$). If clients u, v were distance $c_G(uv)$ apart in G, then for all vertices $i \in \Omega_u, j \in \Omega_v$, $c_{G'}(ij) = c_G(uv)$. Define edge costs to r similarly. For each $i, j \in \Omega_c$, let $c_{G'}(ij) = 0$.[6]
5. Convert O' into a split-delivery, capacitated flower as follows. Map O' onto G' without increasing the cost while covering $\sum_{c \in O'} w(c)$ clients (possible by construction). Number the vertices of this tour in the order they are visited, and pick a random offset R in the range $[1, Q]$. Walk along the tour starting at R, and cut away a strip of the tour every Q vertices (short-cutting past r). Add an edge at each end of a strip back to r, to make each strip an r-rooted tour. [7]
6. For the 1-COP, return this capacitated flower. For the U-COP, we can "unsplit" our solution as follows. Note that if some client's delivery is split, it will be covered by at most two tours; remove any such c from both tours and place it in its own separate tour. Return the resulting capacitated flower.

[6] This construction was first described in [2], and can be used to prove Inequality 11.

[7] If Q is not poly-bounded, note there is a simple poly-time algorithm to do this that avoids explicitly building the graph and trying more than $|V|$ values of R.

We now prove the above procedure is in fact a good approximation for both problems. Consider a fixed optimal capacitated walk W^{OPT} of cost OPT which covers a set of clients C^{OPT}; let $\ell^{OPT} = |C^{OPT}|$. Let TSP^{OPT} be an optimal TSP tour that covers the clients C^{OPT}, and let F^{OPT} be an optimal capacitated flower; one must exist with cost at most $2OPT$.

We utilize two classic results in capacitated vehicle routing.

Lemma 7 *The following inequalities hold for the 1-COP and U-COP:*

$$d(TSP^{OPT}) \le d(F^{OPT}) \le 2OPT \tag{9}$$

$$\frac{2}{Q} \sum_{c \in C^{OPT}} d(c, r) \le d(F^{OPT}) \le 2OPT. \tag{10}$$

Proof (9) holds trivially, since an optimal capacitated walk can be no cheaper than an optimal uncapacitated walk that visits the same clients (*i.e.* shortcut each occurrence of r).

(10) was first shown in [15]. Let C_j' denote the clients covered by the jth tour in F^{OPT}. It follows that

$$d(C_j) \ge 2 \max_{c \in C_j} d(c, r) \ge 2 \frac{\sum_{c \in C_j} d(c, r)}{|C_j|} \ge \frac{2}{Q} \sum_{c \in C_j} d(c, r).$$

We then have

$$d(F^{OPT}) = \sum_j d(C_j) \ge \frac{2}{Q} \sum_{c \in C^{OPT}} d(c, r),$$

as desired. □

(10) can be strengthened for the case where $w(c)$ is any integer ≥ 1:

Lemma 8 *We have the following additional inequality for the U-COP:*

$$\frac{2}{Q} \sum_{c \in C^{OPT}} d(r, c) w(c) \le d(F^{OPT}) \le 2OPT. \tag{11}$$

Proof This result takes inspiration from [2]. Build a new unit-weight graph G' containing r and $w(c)$ copies of each client $c \in C$. Let Ω_c denote the copies of c in G' (so $|\Omega_c| = w(c)$). Duplicate the edges of G for each copy of a client; that is, if clients u, v were distance $d(u, v)$ apart in G, then for all vertices $i \in \Omega_u, j \in \Omega_v$, $d_{G'}(i, j) = d(u, v)$. Define distances to r similarly. For each $i, j \in \Omega_c$, let $d(i, j) = 0$.

Note that F^{OPT} can be converted into a feasible flower in this new unit-weight graph that covers $\sum_{c \in C^{OPT}} w(c)$ vertices, by simply visiting all copies of a client c when that client is visited and then going back to the root at the end, of total cost at most $2OPT$. We can now apply (10) to this tour (since in the proof of (10), we did not make use of the fact that the flower was optimal). Collapsing the sum yields the lemma. □

For each set of clients H define $S_H = \frac{1}{Q}\sum_{c \in H} d(r,c)w(c)$; by (11), $S_{C^{OPT}} \leq B$. Note that W^{OPT} can be converted into a walk in G^* of cost at most $B + 2S_{C^{OPT}} \leq 3B$ (for U-COP, $B + 4S_{C^{OPT}} \leq 5B$) that visits C^{OPT} and the corresponding terminal clients. We can further convert W^{OPT} into a walk that visits *only* terminal clients (and so a walk in G^T), of no greater cost, that covers ℓ^{OPT} terminals. Thus, the ℓ-MST approximation of Chaudhuri *et al.* [9] will find a tree of cost at most $3B + \epsilon$ ($5B + \epsilon$) in G_T that covers ℓ^{OPT} terminals.[8] From this and by construction, the tour O' must have cost at most $6B - 2S_{O'}$ ($10B - 4S_{O'}$).

The expected cost of the extra edges added in step 5 is $2S_{O'}$, so some offset R exists such that we pay at most this amount. Thus, we can cut up O' into smaller tours covering at most Q demand, with total cost $6B$ ($10B - 2S_{O'}$), yielding a $(1, 6+\epsilon)$-approximation to the 1-COP. For the U-COP, the cost of the extra tours in step 6 is also $2S_{O'}$, yielding a $(1, 10+\epsilon)$-approximation. Extending these results to the weighted case, we obtain a $(1-\epsilon', 6+\epsilon)$-approximation for 1-COP and $(1-\epsilon', 10+\epsilon)$-approximation for U-COP for any $\epsilon', \epsilon > 0$.

Proof (Proof of Theorems 1 and 2) Combining Theorem 4 with the $(1 - \epsilon', 6 + \epsilon)$-approximation for 1-COP yields a 25.49-approximation to the unit-demand MD-CkTRP; similarly, combining Theorem 4 with the $(1-\epsilon', 10+\epsilon)$-approximation for U-COP yields a 42.49-approximation to the MD-CkTRP ($\tau \approx 1.616$). □

5 Extensions to MD-CkTRP

We briefly consider two extensions to our problem - non-uniform vehicle capacities, and service delays. In the first case, suppose vehicle i has capacity Q_i. Adjust the definition of $\mathcal{W}_i$ to be all walks of capacity $\leq Q_i$ instead of Q; note that the approximation guarantees of our oracles do not depend on the capacity of the vehicle. Thus, our results extend to vehicles with non-uniform capacities.

To handle service delays, suppose each client c has a service time $\delta(c) \geq 0$, which adds to the time a vehicle must spend traversing its walk (we assume $\delta(r_i) = 0$ for each root r_i). We still wish to minimize the total latency of all clients visited. Define a new metric $d'(u,v) = d(u,v) + \frac{\delta(u)+\delta(v)}{2}$. Solve the MD-C$k$TRP for the new instance (with metric d'). The latency of each node u in the solution returned will be the sum of the edge-lengths, plus the sum of the delays of all the nodes visited before u, plus $\delta(u)/2$. Thus, at an extra loss of $+0.5$ in the approximation, the solution will be a solution for the corresponding problem with service delays.

[8] We omit the ϵ from the rest of the discussion for clarity.

6 Concluding Remarks

We presented a general framework to obtain a constant approximation algorithm for the capacitated multi-depot k-TRP, using bi-criteria approximation algorithms for orienteering style problems, giving the first constant approximations for MD-CkTRP. A consequence of this approach is if our oracles for single-depot (or multi-depot) orienteering are improved, we would have improved approximations for multi-depot (capacitated and uncapacitated) k-TRP. In particular, it seems that the results in Section 4 could be improved. One possible direction for that improvement would be to use ideas from [14] for maximum coverage with knapsack constraints with exponential number of sets. Each set corresponds to one tour of total demand Q and the constraint that the total lengths of all tours picked is no more than the given budget B.

Acknowledgements: We would like to thank anonymous referees for their suggestions and comments.

References

1. A. Ageev and M. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.
2. Kemal Altinkemer and Bezalel Gavish. Heursitics for unequal weight delivery problems with a fixed error guarantee. *Operations Research Letters*, 6(4):149–158, 1987.
3. Aaron Archer and Anna Blasiak. Improved approximation algorithms for the minimum latency problem via prize-collecting strolls. *21st ACM SODA*, pages 429–447, 2010.
4. Aaron Archer, Asaf Levin, and David P Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM Journal on Computing*, 37(5):1472–1498, 2008.
5. A. Blum, P. Chalasani, B. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. *26th ACM STOC*, pages 163–171, 1994.
6. Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.
7. B. Carr and S. Vempala. Randomized meta-rounding. *In Proceedings of STOC*, 2000.
8. Deeparnab Chakrabarty and Chaitanya Swamy. Facility location with client latencies: Linear-programming based techniques for minimum-latency problems. *15th IPCO*, pages 92–103, 2011.
9. Kamalika Chaudhuri, Godfrey Brighten, Satish Rao, and Kunal Talwar. Paths, trees, and minimum latency tours. *44th IEEE-FOCS*, pages 36–45, 2003.
10. Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. *Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques*, pages 72–83, 2004.
11. Jittat Fakcharoenphol, Chris Harrelson, and Satish Rao. The k-traveling repairman problem. *14th ACM-SIAM SODA*, pages 655–664, 2003.
12. Zachary Friggstad and Chaitanya Swamy. Approximation algorithms for regret-bounded vehicle routing and applications to distance-constrained vehicle routing. *In Proceedings of STOC*, pages 744–753, 2014.
13. Michel Goemans and Jon Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82(1-2):111–124, 1998.
14. Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.*, 40(1):56–79, 2015.

15. M. Haimovich and A. H. G Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985.
16. Jens Lysgaard and Sanne Wohlk. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *European Journal of Operational Research*, 236(3):800–810, 2014.
17. Ian Post and Chaitanya Swamy. Linear-programming based approximation algorithms for multi-vehicle minimum latency problems. *26th ACM-SIAM SODA*, pages 512–531, 2015.
18. Juan Carlos Rivera, H. Murat Afsar, and Christian Prins. A multistart iterated local search for the multitrip cumulative capacitated vehicle routing problem. *Computational Optimization and Applications*, 61(1):159–187, 2015.
19. René Sitters. The minimum latency problem is NP-hard for weighted trees. *IPCO*, 2337:230–239, 2002.
20. René Sitters. Polynomial time approximation schemes for the travelling repairman and other minimum latency problems. *25th ACM-SIAM SODA*, 2014.

A Corrected Analysis of MD-kTRP Algorithm from [10]

The analysis given in Section 3.1 does not appear to improve significantly over the analysis given in [10], despite the increased complexity, since they claim a 12-approximation for the uncapacitated multi-depot k-TRP using ℓ-MST as a subroutine (we achieve 11.89). We rederive the approximation ratio for their algorithm using the same ideas and tools in this section, and show that it is in fact a 24-approximation. The difference arises from a small miscalculation made in their paper.

The algorithm of [10] is similar to the one presented in Section 3. Computations are done in phases, and in each phase j we are given a set of uncovered clients C_j^u and a budget 2^j. We use a subroutine $\mathcal{C}$ to find a tour of cost $\leq 2^j$ rooted at each depot, and which cumulatively cover as many clients in C_j^u as possible.

The algorithm for phase j is the following:

function Do-Phase(j)
  for $p = 1, 2$ **do**
    Run $\mathcal{C}(C_j^u, 2^j)$ with clients C_j^u and budget 2^j.
    Append the returned tours to our solution.
    Remove all covered clients from C_j^u.
  end for
end function

We define $\mathcal{C}$ similar to how we did in Section 3, with the exception that we no longer need to specially define the sets $\mathcal{W}_i$; since we are dealing with the uncapacitated problem, $\mathcal{W}_i$ becomes all possible walks from r_i. Thus, our oracle for the MCG instance can be the ℓ-MST oracle described in Section 3.2 (also what is used in [10]).

For the purpose of analysis, we require that $j \geq 1$. As in Section 3.1, we will define n_j to be the number of clients we do not cover by the end of phase j, and similarly n_j^{OPT} will be the number of clients in a fixed optimal solution which have latency $\geq 2^j$. We define $n_{j\leq 0}$ and $n_{j\leq 0}^{OPT}$ to be n.

We can bound what we cover in each iteration, and what we have left to cover after each iteration, using the following two lemmas, which are proven in a similar way to Lemma 3 and Lemma 4. We only need to take into account that in each iteration we find 2 tours per depot instead of one, and explicitly use the ℓ-MST based uncapacitated oracle:

Lemma 9 (Lemma 4 in [10]) *At the end of phase j, we have covered at least $\frac{3}{4}|O_j - C_{j-1}^v|$ clients.*

Lemma 10 (Lemma 5 in [10]) $n_j \leq \frac{1}{4} n_{j-1} + \frac{3}{4} n_j^{OPT}$.

In our solution, any client covered in phase j has latency bounded by $4\sum_{i\leq j} 2^i$, since we must traverse any tour we buy in a previous iteration as well as the one we bought in phase j.[9] Summing this for all clients and using the definition of n_j, we can bound the total latency for our solution by

$$4\sum_{j\geq 1} 2^j n_{j-1}. \tag{LAT-UB}$$

This bound is primarily where we differ from [10]. The bound they present is $4\sum_j 2^j n_j$, using an equivalent definition of n_j. This implies that each client has latency bounded by $4\sum_{i<j} 2^i$, which does not include the tours bought when the client was visited.

The cost of an optimum solution is bounded from below by

$$\sum_{j\geq 0} 2^j (n_j^{OPT} - n_{j+1}^{OPT}),$$

which by re-arranging the summation and noting that $2^j - 2^{j-1} = 2^{j-1}$, is equivalent to

$$n_0^{OPT} + \sum_{j\geq 1} 2^{j-1} n_j^{OPT}. \tag{OPT-LB}$$

We can now use Lemma 10 to bound the cost of our solution against the optimum solution. Each step is derived by re-arranging sums.

$$\begin{aligned}
\sum_{j\geq 1} 2^j n_{j-1} &\leq \frac{1}{4}\sum_{j\geq 1} 2^j n_{j-2} + \frac{3}{4}\sum_{j\geq 1} 2^j n_{j-1}^{OPT} \\
&= \frac{1}{2}\sum_{j\geq 0} 2^j n_{j-1} + \frac{3}{4}\sum_{j\geq 0} 2^{j+1} n_j^{OPT} \\
&= \frac{1}{2}\left(n + \sum_{j\geq 1} 2^j n_{j-1}\right) + \frac{3}{4}\left(4(\text{OPT-LB}) - 2n\right) \\
&\leq \frac{1}{2}\sum_{j\geq 1} 2^j n_{j-1} + 3(\text{OPT-LB}) \\
\implies (\text{LAT-UB}) &\leq 24(\text{OPT-LB}).
\end{aligned}$$

We thus have a 24-approximation to the uncapacitated multi-depot k-TRP.

[9] We drop the ϵ here for convenience, but it can be also dropped from the final approximation ratio due to arguments in [9].