Minimum Makespan Multi-vehicle Dial-a-Ride

Inge Li Gørtz*, Viswanath Nagarajan, and R. Ravi**

[1] Technical University of Denmark. E-mail: ilg@imm.dtu.dk
[2] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598. E-mail: viswanath@us.ibm.com
[3] Tepper School of Business, Carnegie Mellon University. E-mail: ravi@cmu.edu

Abstract. Dial-a-Ride problems consist of a set V of n vertices in a metric space (denoting travel time between vertices) and a set of m objects represented as source-destination pairs $\{(s_i, t_i)\}_{i=1}^m$, where each object requires to be moved from its source to destination vertex. In the *multi-vehicle Dial-a-Ride* problem, there are q vehicles each having capacity k and where each vehicle $j \in [q]$ has its own depot-vertex $r_j \in V$. A feasible schedule consists of a capacitated route for each vehicle (where vehicle j originates and ends at its depot r_j) that together move all objects from their sources to destinations. The objective is to find a feasible schedule that minimizes the maximum completion time (i.e. *makespan*) of vehicles, where the completion time of vehicle j is the time when it returns to its depot r_j at the end of its route. We consider the *preemptive* version of multi-vehicle Dial-a-Ride, where an object may be left at intermediate vertices and transported by more than one vehicle, while being moved from source to destination. Approximation algorithms for the single vehicle Dial-a-Ride problem ($q = 1$) have been considered in [3, 10].

Our main results are an $O(\log^3 n)$-approximation algorithm for *preemptive multi-vehicle Dial-a-Ride*, and an improved $O(\log t)$-approximation for its special case when there is no capacity constraint (here $t \leq n$ is the number of distinct depot-vertices). There is an $\Omega(\log^{1/4} n)$ hardness of approximation known [9] even for single vehicle capacitated preemptive Dial-a-Ride. We also obtain an improved constant factor approximation algorithm for the uncapacitated multi-vehicle problem on metrics induced by graphs excluding any fixed minor.

1 Introduction

The *multi-vehicle Dial-a-Ride* problem involves routing a set of m objects from their sources to respective destinations using a set of q vehicles starting at t distinct depot nodes in an n-node metric. Each vehicle has a *capacity* k which is the maximum number of objects it can carry at any time. Two versions arise based on whether or not the vehicle can use any node in the metric as a preemption (a.k.a. transshipment) point - we study the less-examined *preemptive version* in this paper. The objective in these problems is either the total completion time or the makespan (maximum completion time) over the q vehicles, and again we study the more challenging *makespan* version

* Sponsored in part by a grant from the Carlsberg Foundation.
** Supported in part by NSF grant CCF-0728841.

of the problem. Thus this paper studies the preemptive, capacitated minimum makespan multi-vehicle Dial-a-Ride problem.

While the multiple qualifications may make the problem appear contrived, this is exactly the problem that models courier or mail delivery over a day from several city depots: preemption is cheap and useful for packages, trucks are capacitated and the makespan reflects the daily working time limit for each truck. Despite its ubiquity, this problem has not been as well studied as other Dial-a-Ride versions. One reason from the empirical side is the difficulty in handling the possibility of preemptions in a clean mathematical programming model. On the theoretical side which is the focus of this paper, the difficulty of using preemption in a meaningful way in an approximation algorithm persists. It is further compounded by the hardness of the makespan objective.

The requirement in preemptive Dial-a-Ride, that preemptions are allowed at all vertices, may seem unrealistic. In practice, a subset P of the vertex-set V represents the vertices where preemption is permitted: the two extremes of this general problem are non-preemptive Dial-a-Ride ($P = \emptyset$) and preemptive Dial-a-Ride ($P = V$). However preemptive Dial-a-Ride is more generally applicable: specifically in situations where the preemption-points P form a *net* of the underlying metric (i.e. every vertex in V has a nearby preemption-point). I.e., approximation algorithms for preemptive Dial-a-Ride imply good approximations even in this general setting, wherein the precise guarantee depends on how well P covers V.

We note that although our model allows any number of preemptions and preemptions at all vertices our algorithms do not use this possibility to its full extent. Our algorithm for the capacitated case preempts each object at most once and our algorithm for the uncapacitated case only preempts objects at depot vertices.

The preemptive Dial-a-Ride problem has been considered earlier with a single vehicle, for which an $O(\log n)$ approximation [3] and an $\Omega(\log^{1/4-\epsilon} n)$ hardness of approximation (for any constant $\epsilon > 0$) [9] are known. Note that the completion time and makespan objectives coincide for this case.

Moving to multiple vehicles, the total completion time objective admits a straightforward $O(\log n)$ approximation along the lines of the single vehicle problem [3]: Using the FRT tree embedding [7], one can reduce to tree-metrics at the loss of an expected $O(\log n)$ factor, and there is a simple constant approximation for this problem on trees. The maximum completion time or makespan objective, which we consider in this paper turns out to be considerably harder. Due to non-linearity of the makespan objective, the above reduction to tree-metrics does not hold. Furthermore, the makespan objective does not appear easy to solve even on trees.

Unlike in the single-vehicle case, note that an object in multi-vehicle Dial-a-Ride may be transported by several vehicles one after the other. Hence it is important for the vehicle routes to be coordinated so that the objects trace valid paths from respective sources to destinations. For example, a vehicle may have to wait at a vertex for other vehicles carrying common objects to arrive. Interestingly, the multi-vehicle Dial-a-Ride problem captures aspects of both machine scheduling and network design problems.

Results and Paper Outline We first consider the special case of multi-vehicle Dial-a-Ride (*uncapacitated* mDaR) where the vehicles have no capacity constraints (i.e. $k \geq m$). This problem is interesting in itself, and serves as a good starting point before we present the algorithm for the general case. The uncapacitated mDaR problem itself highlights differences from the single vehicle case: For example, in single vehicle Dial-a-Ride, preemption plays no role in the absence of capacity constraints; however in uncapacitated mDaR, an optimal non-preemptive schedule may take $\Omega(\sqrt{q})$ longer than the optimal preemptive schedule (see the full version of the paper). We prove the following theorem in Section 2.

Theorem 1 *There is an $O(\log t)$-approximation algorithm for uncapacitated preemptive* mDaR *obtaining a tour where objects are only preempted at depot vertices.*

The above algorithm has two main steps: the first one reduces the instance (at a constant factor loss in the performance guarantee) to one in which all demands are between depots (a "depot-demand" instance). In the second step, we use a *sparse spanner* on the demand graph to construct routes for moving objects across depots.

We also obtain an improved guarantee for the following special class of metrics using the notion of *sparse covers* in such metrics [14].

Theorem 2 *There is an $O(1)$-approximation algorithm for uncapacitated* mDaR *on metrics induced by graphs that exclude any fixed minor.*

In Section 3, we study the capacitated preemptive mDaR problem, and obtain our main result. Recall that there is an $\Omega(\log^{1/4-\epsilon} n)$ hardness of approximation for even single vehicle Dial-a-Ride [9]. A feasible solution to preemptive mDaR is said to be *1-preemptive* if every object is preempted at most once while being moved from its source to destination.

Theorem 3 *There is an $O(\log^3 n)$ approximation algorithm for preemptive* mDaR *obtaining a 1-preemptive tour.*

This algorithm has four key steps: (1) We *preprocess* the input so that demand points that are sufficiently far away from each other can be essentially decomposed into separate instances for the algorithm to handle independently. (2) We then solve a single-vehicle instance of the problem that obeys some additional bounded-delay property (Theorem 6) that we prove; This property combines ideas from algorithms for *light approximate shortest path trees* [13] and *capacitated vehicle routing* [11]. The bounded-delay property is useful in *randomly partitioning* the single vehicle solution among the q vehicles available to share this load. This random partitioning scheme is reminiscent of the work of Hochbaum-Maass [12], Baker [1] and Klein-Plotkin-Rao [14], in trying to average out the effect of the cutting in the objective function. (3) The partitioned segments of the single vehicle tour are assigned to the available vehicles; However, to check if this assignment is feasible we solve a matching problem that identifies cases when this load assignment must be *rebalanced*. This is perhaps the most interesting step in the algorithm since it identifies stronger lower bounds for subproblems where the current load assignment is not balanced. (4) We finish up by *recursing* on the load rebalanced subproblem; An interesting feature of the recursion is that the fraction of demands that are processed recursively is not a fixed value (as is more common in such recursive algorithms) but is a carefully chosen function of the number of vehicles on which these demands have to be served.

Due to lack of space some proofs are omitted from this paper. The proofs can be found in the full version.

Related Work Dial-a-Ride problems form an interesting subclass of Vehicle Routing Problems that are well studied in the operations research literature. Paepe et al. [5] provide a classification of Dial-a-Ride problems using a notation similar to that for scheduling and queuing problems: preemption is one aspect in this classification. Savelsberg and Sol [18] and Cordeau and Laporte [4] survey several variants of non-preemptive Dial-a-Ride problems that have been studied in the literature. Most Dial-a-Ride problems arising in practice involve making routing decisions for multiple vehicles.

Dial-a-Ride problems with transshipment (the preemptive version) have been studied in [15–17]. These papers consider a more general model where preemption is allowed only at a specified subset of vertices. Our model (and that of [3]) is the special case when every vertex can serve as a preemption point. It is clear that preemption only reduces the cost of serving demands: [17] studied the maximum decrease in the optimal cost upon introducing one preemption point. [15, 16] also model time-windows on the demands, and study heuristics and a column-generation based approach; they also describe applications (eg. courier service) that allow for preemptions. The *truck and trailer routing problem* has been studied in [2, 19]. Here a number of capacitated trucks and trailers are used to deliver all objects. Some customers are only accessible without the trailer. The trailers can be parked at any point accessible with a trailer and it is possible to shift demand loads between the truck and the trailer at the parking places.

For single vehicle Dial-a-Ride, the best known approximation guarantee for the preemptive version is $O(\log n)$ (Charikar and Raghavachari [3]), and an $\Omega(\log^{1/4-\epsilon} n)$ hardness of approximation (for any constant $\epsilon > 0$) is shown in Gørtz [9]. The non-preemptive version appears much harder and the best known approximation ratio is $\min\{\sqrt{k}\log n, \sqrt{n}\log^2 n\}$ (Charikar and Raghavachari [3], Gupta et al. [10]); however to the best of our knowledge, APX-hardness is the best lower bound. There are known instances of single vehicle Dial-a-Ride where the ratio between optimal non-preemptive and preemptive tours is $\Omega(\sqrt{n})$ in general metrics [3], and $\tilde{\Omega}(n^{1/8})$ in the Euclidean plane [10]. A 1.8-approximation is known for the $k = 1$ special case of single vehicle Dial-a-Ride (a.k.a. *stacker-crane* problem) [8].

The uncapacitated case of preemptive mDaR is also a generalization of a problem called *nurse-station-location* that was studied in Even et al. [6] (where a 4-approximation algorithm was given). Nurse-station-location is a special case of uncapacitated mDaR when each source-destination pair coincides on a single vertex. In this paper, we handle not only the case with arbitrary pairs (uncapacitated mDaR), but also the more general problem with finite capacity restriction.

Problem Definition and Preliminaries We represent a finite metric as (V, d) where V is the set of vertices and d is a symmetric distance function satisfying the triangle inequality. For subsets $A, B \subseteq V$ we denote by $d(A, B)$ the minimum distance between a vertex in A and another in B, so $d(A, B) = \min\{d(u, v) \mid u \in A, v \in B\}$. For a subset $E \subseteq \binom{V}{2}$ of edges, $d(E) := \sum_{e\in E} d_e$ denotes the total length of edges in E.

The *multi-vehicle Dial-a-Ride problem* (mDaR) consists of an n-vertex metric (V, d), m objects specified as source-destination pairs $\{s_i, t_i\}_{i=1}^m$, q vehicles having respective depot-vertices $\{r_j\}_{j=1}^q$, and a common vehicle capacity k. A feasible schedule is a set of q routes, one for each vehicle (where the route for vehicle $j \in [q]$ starts and ends at r_j), such that no vehicle carries more than k objects at any time and each object is moved from its source to destination. The completion time C_j of any vehicle $j \in [q]$ is the time when vehicle j returns to its depot r_j at the end of its route (the schedule is assumed to start at time 0). The objective in mDaR is to minimize the makespan, i.e., $\min \max_{j\in[q]} C_j$. We denote by $S := \{s_i \mid i \in [m]\}$ the set of sources, $T := \{t_i \mid i \in [m]\}$ the set of destinations, $R := \{r_j \mid j \in [q]\}$ the set of distinct depot-vertices, and $t := |R|$ the number of distinct depots. Unless mentioned otherwise, we only consider the *preemptive* version, where objects may be left at intermediate vertices while being moved from source to destination.

Single vehicle Dial-a-Ride. The following are lower bounds for the single vehicle problem: the minimum length TSP tour on the depot and all source/destination vertices (*Steiner* lower bound), and $\frac{\sum_{i=1}^{m} d(s_i,t_i)}{k}$ (*flow* lower bound). Charikar and Raghavachari [3] gave an $O(\log n)$ approximation algorithm for this problem based on the above lower bounds. Gupta et al. [10] showed that the single vehicle preemptive Dial-a-Ride problem always has a 1-preemptive tour of length $O(\log^2 n)$ times the Steiner and flow lower-bounds.

Lower bounds for mDaR. The quantity $\frac{\sum_{i=1}^{m} d(s_i,t_i)}{qk}$ is a lower bound similar to the flow bound for single vehicle Dial-a-Ride. Analogous to the Steiner lower bound above, is the optimal value of an induced *nurse-station-location* instance. In the nurse-station-location problem [6], we are given a metric (V, d), a set $\mathcal{T}$ of terminals and a multi-set $\{r_j\}_{j=1}^{q}$ of depot-vertices; the goal is to find a collection $\{F_j\}_{j=1}^{q}$ of trees that collectively contain all terminals $\mathcal{T}$ such that each tree F_j is rooted at vertex r_j and $\max_{j=1}^{q} d(F_j)$ is minimized. Even et al. [6] gave a 4-approximation algorithm for this problem. The optimal value of the nurse-station-location instance with depots $\{r_j\}_{j=1}^{q}$ (depots of vehicles in mDaR) and terminals $\mathcal{T} = S \cup T$ is a lower bound for mDaR. The following are some lower bounds implied by nurse-station-location: (a) $1/q$ times the minimum length forest that connects every vertex in $S \cup T$ to some depot vertex, (b) $\max_{i\in[m]} d(R, s_i)$, and (c) $\max_{i\in[m]} d(R, t_i)$. Finally, it is easy to see that $\max_{i\in[m]} d(s_i, t_i)$ is also a lower bound for mDaR.

2 Uncapacitated Preemptive mDaR

In this section we study the uncapacitated special case of preemptive mDaR, where vehicles have no capacity constraints (i.e., capacity $k \geq m$). We give an algorithm that achieves an $O(\log t)$ approximation ratio for this problem (recall $t \leq n$ is the number of distinct depots). Unlike in the single vehicle case, preemptive and non-preemptive versions of mDaR are very different even without capacity constraints (there exists an $\Omega(\sqrt{q})$ factor gap, where q is number of vehicles). The algorithm for uncapacitated preemptive mDaR proceeds in two stages. Given any instance, it is first reduced (at the loss of a constant factor) to a depot-demand instance, where all demands are between depot vertices. Then the depot-demand instance is solved using an $O(\log t)$ approximation algorithm.

Reduction to depot-demand instances We define *depot-demand instances* as those instances of uncapacitated mDaR where all demands are between depot vertices. Given any instance $\mathcal{I}$ of uncapacitated mDaR, the algorithm UncapMulti (given below) reduces $\mathcal{I}$ to a depot-demand instance.

Input: instance $\mathcal{I}$ of uncapacitated preemptive mDaR.

1. Solve the nurse-station-location instance with depots $\{r_j\}_{j=1}^q$ and all sources/ destinations $S \cup T$ as terminals, using the 4-approximation algorithm [6]. Let $\{F_j\}_{j=1}^q$ be the resulting trees covering $S \cup T$ such that each tree F_j is rooted at depot r_j.
2. Define a depot-demand instance $\mathcal{J}$ of uncapacitated mDaR on the same metric and set of vehicles, where the demands are $\{(r_j, r_l) \mid s_i \in F_j \;\&\; t_i \in F_l,\ 1 \le i \le m\}$. For any object $i \in [m]$ let the *source depot* be the depot r_j for which $s_i \in F_j$ and the *destination depot* be the depot r_l for which $t_i \in F_l$.
3. **Output** the following schedule for $\mathcal{I}$:
   (a) Each vehicle $j \in [q]$ traverses tree F_j by an Euler tour, picks up all objects from sources in F_j and brings them to their source-depot r_j.
   (b) Vehicles implement a schedule for *depot-demand instance* $\mathcal{J}$, and all objects are moved from their source-depot to destination-depot (see Section 2).
   (c) Each vehicle $j \in [q]$ traverses tree F_j by an Euler tour, picks up all objects having destination-depot r_j and brings them to their destinations in F_j.

Note that objects only are preempted at depot vertices. We now argue that the reduction in UncapMulti only loses a constant approximation factor. Let B denote the optimal makespan of instance $\mathcal{I}$. Since the optimal value of the nurse-station-location instance solved in the first step of UncapMulti is a lower bound for $\mathcal{I}$, we have $\max_{j=1}^q d(F_j) \le 4B$.

Claim. The optimal makespan for the depot-demand instance $\mathcal{J}$ is at most $17B$.

Assuming a feasible schedule for $\mathcal{J}$, it is clear that the schedule returned by UncapMulti is feasible for the original instance $\mathcal{I}$. The first and third rounds in $\mathcal{I}$'s schedule require at most $8B$ time each. Thus an approximation ratio α for depot-demand instances implies an approximation ratio of $17\alpha + 16$ for general instances. Next we show an $O(\log t)$-approximation algorithm for depot-demand instances (here t is the number of depots), which implies Theorem 1.

Algorithm for depot-demand instances Let $\mathcal{J}$ be any depot-demand instance: note that the instance defined in the second step of UncapMulti is of this form. It suffices to restrict the algorithm to the induced metric (R, d) on only depot vertices, and use only one vehicle at each depot in R. Consider an undirected graph H consisting of vertex set R and edges corresponding to demands: there is an edge between vertices r and s iff there is an object going from *either* r to s *or* s to r. Note that the metric length of any edge in H is at most the optimal makespan $\tilde{B}$ of instance $\mathcal{J}$. In the schedule produced by our algorithm, vehicles will only use edges of H. Thus in order to obtain an $O(\log t)$ approximation, it suffices to show that each vehicle only traverses $O(\log t)$ edges. Based on this, we further reduce $\mathcal{J}$ to the following instance $\mathcal{H}$ of uncapacitated mDaR: the underlying metric is shortest paths in the graph H (on vertices R), with one vehicle at each R-vertex, and for every edge $(u, v) \in H$ there is a demand from u to v and one from v to u. Clearly any schedule for $\mathcal{H}$ having makespan β implies one for $\mathcal{J}$ of makespan $\beta \cdot \tilde{B}$. The next lemma implies an $O(\log |R|)$ approximation for depot-demand instances.

Lemma 4 *There exists a poly-time computable schedule for $\mathcal{H}$ with makespan $O(\log t)$, where $t = |R|$.*

Proof: Let $\alpha = \lceil \lg t \rceil + 1$. We first construct a *sparse spanner* A of H as follows: consider edges of H in an arbitrary order, and add an edge $(u, v) \in H$ to A iff the shortest path between u and v using current edges of A is more than 2α. It is clear from this construction that the girth of A (length of its shortest cycle) is at least 2α, and that for every edge $(u, v) \in H$, the shortest path between u and v in A is at most 2α.

We now assign each edge of A to one of its end-points such that each vertex is assigned at most two edges. Repeatedly pick any vertex v of degree at most two in A, assign its adjacent edges to v, and remove these edges and v from A. We claim that at the end of this procedure (when no vertex has degree at most 2), all edges of A would have been removed (i.e. assigned to some vertex). Suppose for a contradiction that this is not the case. Let $\tilde{A} \neq \phi$ be the remaining graph; note that $\tilde{A} \subseteq A$, so the girth of $\tilde{A}$ is at least 2α. Every vertex in $\tilde{A}$ has degree at least 3, and there is at least one such vertex w. Consider performing a breadth-first search in $\tilde{A}$ from w. Since the girth of $\tilde{A}$ is at least 2α, the first α levels of the breadth-first search is a tree. Furthermore every vertex has degree at least 3, so each vertex in the first $\alpha - 1$ levels has at least 2 children. This implies that $\tilde{A}$ has at least $1 + 2^{\alpha-1} > t$ vertices, which is a contradiction! For each vertex $v \in R$, let A_v denote the edges of A assigned to v by the above procedure; we argued that $\cup_{v \in R} A_v = A$, and $|A_v| \leq 2$ for all $v \in R$.

The schedule for $\mathcal{H}$ involves 2α rounds as follows. In each round, every vehicle $v \in R$ traverses the edges in A_v (in both directions) and returns to v. Since $|A_v| \leq 2$ for all vertices v, each round takes 4 units of time; so the makespan of this schedule is $8\alpha = O(\log t)$. The route followed by each object in this schedule is the shortest path from its source to destination in spanner A; note that the length of any such path is at most 2α. To see that this is indeed feasible, observe that every edge of A is traversed by some vehicle in each round. Hence in each round, every object traverses one edge along its shortest path (unless it is already at its destination). Thus after 2α rounds, all objects are at their destinations. ■

Tight example for uncapacitated mDaR lower bounds. We note that known lower bounds for uncapacitated preemptive mDaR are insufficient to obtain a sub-logarithmic approximation guarantee. The lower bounds we used in our algorithm are the following: $\max_{i \in [m]} d(s_i, t_i)$, and the optimal value of a nurse-station-location instance with depots $\{r_j\}_{j=1}^q$ and terminals $S \cup T$. We are not aware of any lower bounds stronger than these two bounds. There exist instances of uncapacitated mDaR where the optimal makespan is a factor $\Omega(\frac{\log t}{\log\log t})$ larger than both the above lower bounds.

3 Preemptive multi-vehicle Dial-a-Ride

In this section we prove our main result: an $O(\log^2 m \cdot \log n)$ approximation algorithm for the preemptive mDaR problem. We first prove a new structure theorem on single-vehicle Dial-a-Ride tours (Subsection 3.1) that preempts each object at most once, and where the total time spent by objects in the vehicle is small. Obtaining such a single vehicle tour is crucial in our algorithm for preemptive mDaR, which appears in Section 3.2. The algorithm for mDaR relies on a partial coverage algorithm Partial that, given subsets Q of vehicles and D of demands, outputs a schedule for Q of near-optimal makespan that covers some *fraction* of demands in D. Algorithm Partial follows an interesting recursive framework where the fraction of satisfied demands is not a fixed value but some function of the number $|Q|$ of vehicles (Lemma 7). The main steps in Partial are as follows. **(1)** Obtain a *single-vehicle* tour satisfying 1-preemptive and bounded-delay properties (Theorem 6), **(2)** Randomly partition the single vehicle tour into $|Q|$ equally spaced pieces, **(3)** Solve a matching problem to assign *some* of these pieces to vehicles of Q that satisfy a subset of demands D, **(4)** A suitable fraction of the residual demands in D are covered recursively by unused vehicles of Q.

3.1 Capacitated Vehicle Routing with Bounded Delay

Before we present the structural result on Dial-a-Ride tours, we consider the classic *capacitated vehicle routing problem* [11] with an additional constraint on object 'delays'. In the capacitated vehicle routing problem (CVRP) we are given a metric (V, d), specified depot-vertex $r \in V$, and m objects each having source r and respective destinations $\{t_i\}_{i\in[m]}$. The goal is to compute a minimum length *non-preemptive* tour of a capacity k vehicle originating at r that moves all objects from r to their destinations. In *CVRP with bounded delay*, we are additionally given a *delay parameter* $\beta > 1$, and the goal is to find a minimum length capacitated non-preemptive tour serving all objects such that the time spent by each object $i \in [m]$ in the vehicle is at most $\beta \cdot d(r, t_i)$. The following are natural lower bounds [11], even without the bounded delay constraint: (i) the minimum length TSP tour on $\{r\} \cup \{t_i \mid i \in [m]\}$ (cf. Steiner lower bound), and (ii) the quantity $\frac{2}{k}\sum_{i=1}^{m} d(r, t_i)$ (cf. flow lower bound).

Theorem 5 *There is a* $(2.5 + \frac{3}{\beta-1})$ *approximation algorithm for CVRP with bounded delay, where* $\beta > 1$ *is the delay parameter. This guarantee is relative to the Steiner and flow lower bounds.*

We now consider the *single vehicle* preemptive Dial-a-Ride problem given by metric (V, d), set D of demands, and a vehicle of capacity k. We prove the following structural result which extends a result from [10].

Theorem 6 *There is a randomized poly-time computable 1-preemptive tour* τ *servicing* D *that satisfies the following conditions (where* LB_{pmt} *is the maximum of the Steiner and flow lower bounds):*

1. **Total length:** $d(\tau) \le O(\log^2 n) \cdot \mathsf{LB}_{pmt}$.
2. **Bounded delay:** $\sum_{i\in D} T_i \le O(\log n) \sum_{i\in D} d(s_i, t_i)$ *where* T_i *is the total time spent by object* $i \in D$ *in the vehicle under the schedule given by* τ.

3.2 Algorithm for preemptive mDaR

The algorithm first guesses the optimal makespan B of the given instance of preemptive mDaR (it suffices to know B within a constant factor for a polynomial-time algorithm). Let $\alpha = 1 - \frac{1}{1+\lg m}$. For any subset $Q \subseteq [q]$, we abuse the notation and use Q to denote both the set of vehicles Q and the multi-set of depots corresponding to vehicles Q.

We give an algorithm Partial that takes as input a tuple $\langle Q, D, B\rangle$ where $Q \subseteq [q]$ is a subset of vehicles, $D \subseteq [m]$ a subset of demands and $B \in \mathbb{R}_+$, with the *promise* that vehicles Q (originating at their respective depots) suffice to completely serve the demands D at a makespan of B. Given such a promise, Partial $\langle Q, D, B\rangle$ returns a schedule of makespan $O(\log n \log m) \cdot B$ that serves a good fraction of D. Algorithm Partial$\langle Q, D, B\rangle$ is given in below. We set parameter $\rho = \Theta(\log n \log m)$, the precise constant in the Θ-notation comes from the analysis.

Input: Vehicles $Q \subseteq [q]$, demands $D \subseteq [m]$, bound $B \geq 0$ *such that* Q can serve D at makespan B.

Preprocessing

1. If the minimum spanning tree (MST) on vertices Q contains an edge of length greater than $3B$, there is a non-trivial partition $\{Q_1, Q_2\}$ of Q with $d(Q_1, Q_2) > 3B$. For $j \in \{1, 2\}$, let $V_j = \{v \in V \mid d(Q_j, v) \leq B\}$ and D_j be all demands of D induced on V_j. Run in parallel the schedules from Partial$\langle Q_1, D_1, B\rangle$ and Partial$\langle Q_2, D_2, B\rangle$. Assume there is no such long edge in the following.

Random partitioning

2. Obtain single-vehicle 1-preemptive tour τ using capacity k and serving demands D (Theorem 6).
3. Choose a uniformly random offset $\eta \in [0, 2\rho B]$ and cut edges of tour τ at distances $\{2p\rho B + \eta \mid p = 1, 2, \cdots\}$ along the tour to obtain a set $\mathcal{P}$ of pieces of τ.
4. C'' is the set of objects $i \in D$ such that i is carried by the vehicle in τ over some edge that is cut in Step (3); and $C' := D \setminus C''$. Ignore the cut objects C'' in the rest of the algorithm.

Load rebalancing

5. Construct a bipartite graph H with vertex sets $\mathcal{P}$ and Q and an edge between piece $P \in \mathcal{P}$ and depot $f \in Q$ iff $d(f, P) \leq 2B$. For any subset $A \subseteq \mathcal{P}$, $\Gamma(A) \subseteq Q$ denotes the neighborhood of A in graph H. Let $\mathcal{S} \subseteq \mathcal{P}$ be any *maximal* set that satisfies $|\Gamma(\mathcal{S})| \leq \frac{|\mathcal{S}|}{2}$.
6. Compute a *2-matching* $\pi : \mathcal{P} \setminus \mathcal{S} \to Q \setminus \Gamma(\mathcal{S})$, i.e., a function such that the number of pieces mapping to any $f \in Q \setminus \Gamma(\mathcal{S})$ is $|\pi^{-1}(f)| \leq 2$.

Recursion

7. Define $C_1 := \{i \in C' \mid \text{ either } s_i \in \mathcal{S} \text{ or } t_i \in \mathcal{S}\}$; and $C_2 := C' \setminus C_1$.
8. Run in parallel the *recursive* schedule Partial$\langle \Gamma(\mathcal{S}), C_1, B\rangle$ for C_1 and the following for C_2:
   (a) Each vehicle $f \in Q \setminus \Gamma(\mathcal{S})$ traverses the pieces $\pi^{-1}(f)$, moving all C_2-objects in them from their source to preemption-vertex, and returns to its depot.
   (b) Each vehicle $f \in Q \setminus \Gamma(\mathcal{S})$ again traverses the pieces $\pi^{-1}(f)$, this time moving all C_2-objects in them from their preemption-vertex to destination, and returns to its depot.

Output: A schedule for vehicles Q of makespan $(16+16\rho) \cdot B$ that serves an $\alpha^{\lg \min\{|Q|, 2m\}}$ fraction of D.

Lemma 7 *If there exists a schedule for vehicles Q covering all demands D, having makespan at most B, then* Partial *invoked on $\langle Q, D, B\rangle$ returns a schedule of vehicles Q of makespan at most $(16 + 16\rho) \cdot B$ that covers at least an $\alpha^{\lg z}$ fraction of D (here $z := \min\{|Q|, 2m\} \leq 2m$).*

The final algorithm invokes Partial iteratively until all demands are covered: each time with the entire set $[q]$ of vehicles, all uncovered demands, and bound B. If $D \subseteq [m]$ is the set of uncovered demands at any iteration, Lemma 7 implies that $\mathsf{Partial}\langle[q], D, B\rangle$ returns a schedule of makespan $O(\log m \log n) \cdot B$ that serves at least $\frac{1}{4}|D|$ demands. Hence a standard set-cover analysis implies that all demands will be covered in $O(\log m)$ rounds, resulting in a makespan of $O(\log^2 m \log n) \cdot B$.

It remains to prove Lemma 7. We proceed by induction on the number $|Q|$ of vehicles. The base case $|Q| = 1$ reduces to the single vehicle preemptive Dial-a-Ride, where we can serve *all* the demands D in a 1-preemptive fashion at makespan $O(\log^2 n) \cdot B$ using the algorithm from [10]. In the rest of this section, we prove the inductive step.

Preprocessing. Suppose Step (1) applies. Note that $d(V_1, V_2) > B$ and hence there is no demand with source in one of $\{V_1, V_2\}$ and destination in the other. So demands D_1 and D_2 partition D. Furthermore in the optimal schedule, vehicles Q_j (any $j = 1, 2$) only visit vertices in V_j (otherwise the makespan would be greater than B). Thus the two recursive calls to $\mathsf{Partial}$ satisfy the assumption: there is some schedule of vehicles Q_j serving D_j having makespan B. Inductively, the schedule returned by $\mathsf{Partial}$ for each $j = 1, 2$ has makespan at most $(16+16\rho)\cdot B$ and covers at least $\alpha^{\lg c} \cdot |D_j|$ demands from D_j, where $c \leq \min\{|Q| - 1, 2m\} \leq z$. The schedules returned by the two recursive calls to $\mathsf{Partial}$ can clearly be run in parallel and this covers at least $\alpha^{\lg z}(|D_1| + |D_2|)$ demands, i.e. an $\alpha^{\lg z}$ fraction of D. So we have the desired performance in this case.

Random partitioning. The harder part of the analysis is when Step (1) does not apply: so the MST length on Q is at most $3|Q| \cdot B$. Note that when the depots Q are contracted to a single vertex, the MST on the end-points of D plus the contracted depot-vertex has length at most $|Q| \cdot B$ (the optimal makespan schedule induces such a tree). Thus the MST on the depots Q along with end-points of D has length at most $4|Q| \cdot B$. Based on the assumption in Lemma 7 and the flow lower bound, we have $\sum_{i \in D} d(s_i, t_i) \leq k|Q| \cdot B$. It follows that for the single vehicle Dial-a-Ride instance solved in Step (2), the Steiner and flow lower-bounds (denoted LB_{pmt} in Theorem 6) are $O(1) \cdot |Q|B$. Theorem 6 now implies that τ is a 1-preemptive tour τ servicing D, of length at most $O(\log^2 n)|Q| \cdot B$ such that $\sum_{i \in D} T_i \leq O(\log n) \cdot |D|B$, where T_i denotes the total time spent in the vehicle by demand $i \in D$. The bound on the delay uses the fact that $\max_{i=1}^m d(s_i, t_i) \leq B$.

Choosing a large enough constant corresponding to $\rho = \Theta(\log n \log m)$, the length of τ is upper bounded by $\rho|Q| \cdot B$ (since $n \leq 2m$). So the cutting procedure in Step (3) results in at most $|Q|$ pieces of τ, each of length at most $2\rho B$. The objects $i \in C''$ (as defined in Step (4)) are called *cut objects*. We restrict our attention to the other objects $C' = D \setminus C''$ that are not 'cut'. For each object $i \in C'$, the path traced by it (under single vehicle tour τ) from its source s_i to preemption-point and the path (under τ) from its preemption-point to t_i are each completely contained in pieces of $\mathcal{P}$. Figure 1 gives an example of objects in C' and C'', and the cutting procedure.

Claim. The expected number of objects in C'' is at most $\sum_{i \in D} \frac{T_i}{2\rho B} \leq O(\frac{1}{\log m}) \cdot |D|$.

We can derandomize Step (3) and pick the best offset η (there are at most polynomially many combinatorially distinct offsets). Claim 3.2 implies (again choosing large enough constant in $\rho = \Theta(\log n \log m)$) that $|C'| \geq (1 - \frac{1}{2 \lg m})|D| \geq \alpha \cdot |D|$ demands are *not cut*. From now on we only consider the set C' of uncut demands. Let $\mathcal{P}$ denote the pieces obtained by cutting τ as above, recall $|\mathcal{P}| \leq |Q|$. A piece $P \in \mathcal{P}$ is said to be non-trivial if the vehicle in the 1-preemptive tour τ carries some C'-object while traversing P. Note that the number of non-trivial pieces in $\mathcal{P}$ is at most

$2|C'| \leq 2m$: each C'-object appears in at most 2 pieces, one where it is moved from source to preemption-vertex and other from preemption-vertex to destination. Retain only the non-trivial pieces in $\mathcal{P}$; so $|\mathcal{P}| \leq \min\{|Q|, 2m\} = z$. The pieces in $\mathcal{P}$ may not be one-one assignable to the depots since the algorithm has not taken the depot locations into account. We determine which pieces may be assigned to depots by considering a matching problem between $\mathcal{P}$ and the depots in Step (5) and (6).

Load rebalancing. The bipartite graph H (defined in Step (5)) represents which pieces and depots may be assigned to each other. Piece $P \in \mathcal{P}$ and depot $f \in Q$ are assignable iff $d(f, P) \leq 2B$, and in this case graph H contains an edge (P, f). We claim that corresponding to the 'maximal contracting' set $\mathcal{S}$ (defined in Step (5)), the 2-matching π (in Step (6)) is guaranteed to exist. Note that $|\Gamma(\mathcal{S})| \leq \frac{|\mathcal{S}|}{2}$, but $|\Gamma(\mathcal{T})| > \frac{|\mathcal{T}|}{2}$ for all $\mathcal{T} \supset \mathcal{S}$. For any $T' \subseteq \mathcal{P} \setminus \mathcal{S}$, let $\tilde{\Gamma}(T')$ denote the neighborhood of T' in $Q \setminus \Gamma(\mathcal{S})$. The maximality of $\mathcal{S}$ implies: for any non-empty $T' \subseteq \mathcal{P} \setminus \mathcal{S}$, $\frac{|\mathcal{S}|}{2} + \frac{|T'|}{2} = \frac{|\mathcal{S} \cup T'|}{2} < |\Gamma(\mathcal{S} \cup T')| = |\Gamma(\mathcal{S})| + |\tilde{\Gamma}(T')|$, i.e. $|\tilde{\Gamma}(T')| \geq \frac{|T'|}{2}$. Hence by *Hall's condition*, there is a 2-matching $\pi : \mathcal{P} \setminus \mathcal{S} \to Q \setminus \Gamma(\mathcal{S})$. The set $\mathcal{S}$ and 2-matching π can be easily computed in polynomial time.

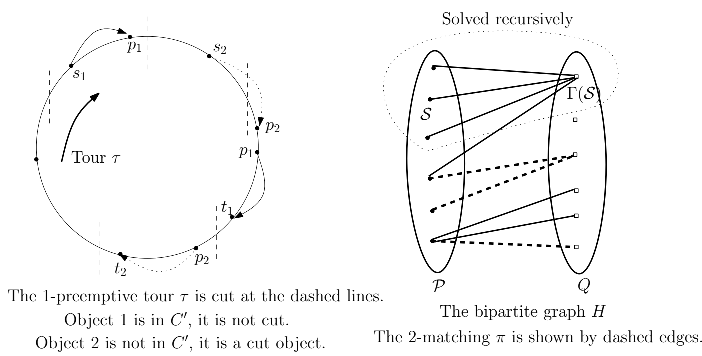

Fig. 1. Cutting and patching steps in algorithm Partial.

Recursion. In Step (7), demands C' are further partitioned into two sets: C_1 consists of objects that are *either* picked-up *or* dropped-off in some piece of $\mathcal{S}$; and C_2-objects are picked-up *and* dropped-off in pieces of $\mathcal{P} \setminus \mathcal{S}$. The vehicles $\Gamma(\mathcal{S})$ suffice to serve all C_1 objects, as shown below.

Claim. There exists a schedule of vehicles $\Gamma(\mathcal{S})$ serving C_1, with makespan B.

In the final schedule, a *large fraction* of C_1 demands are served by vehicles $\Gamma(\mathcal{S})$, and *all* the C_2 demands are served by vehicles $Q \setminus \Gamma(\mathcal{S})$. Figure 1 shows an example of this partition.

Serving C_1 demands. Based on Claim 3.2, the recursive call Partial $\langle \Gamma(\mathcal{S}), C_1, B \rangle$ (made in Step (8)) satisfies the assumption required in Lemma 7. Since $|\Gamma(\mathcal{S})| \leq \frac{|\mathcal{P}|}{2} \leq \frac{|Q|}{2} < |Q|$, we obtain inductively that Partial $\langle \Gamma(\mathcal{S}), C_1, B \rangle$ returns a schedule of makespan $(16 + 16\rho) \cdot B$ covering at least $\alpha^{\lg y} \cdot |C_1|$ demands of C_1, where $y = \min\{|\Gamma(\mathcal{S})|, 2m\}$. Note that $y \leq |\Gamma(\mathcal{S})| \leq |\mathcal{P}|/2 \leq z/2$ (as $|\mathcal{P}| \leq z$), which implies that at least $\alpha^{\lg z - 1}|C_1|$ demands are covered.

Serving C_2 demands. These are served by vehicles $Q \setminus \Gamma(\mathcal{S})$ using the 2-matching π, in two rounds as specified in Step (8). This suffices to serve all objects in C_2 since for any $i \in C_2$, the paths traversed by object i under τ, namely $s_i \rightsquigarrow p_i$ (its preemption-point) and $p_i \rightsquigarrow t_i$ are contained in pieces of $\mathcal{P} \setminus \mathcal{S}$. Furthermore, since $|\pi^{-1}(f)| \leq 2$ for all $f \in Q \setminus \Gamma(\mathcal{S})$, the distance traveled by vehicle f in one round is at most $2 \cdot 2(2B + 2\rho B)$. So the time taken by this schedule is at most $2 \cdot 4(2B + 2\rho B) = (16 + 16\rho) \cdot B$.

The schedule of vehicles $\Gamma(\mathcal{S})$ (serving C_1) and vehicles $Q \setminus \Gamma(\mathcal{S})$ (serving C_2) can clearly be run in parallel. This takes time $(16 + 16\rho) \cdot B$ and covers in total at least $|C_2| + \alpha^{\lg z - 1}|C_1| \geq \alpha^{\lg z - 1}|C'| \geq \alpha^{\lg z}|D|$ demands of D. This proves the inductive step of Lemma 7.

Using Lemma 7 repeatedly as mentioned earlier, we obtain an $O(\log^2 m \cdot \log n)$ approximation algorithm for capacitated preemptive mDaR.

References

1. B. S. Baker. Approximation Algorithms for NP-Complete Problems on Planar Graphs. *J. ACM*, 41:153–180, 1994.
2. I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computer & Operations Research*, 29:469–488, 2002.
3. M. Charikar and B. Raghavachari. The Finite Capacity Dial-A-Ride Problem. In *FOCS*, pages 458–467, 1998.
4. J.-F. Cordeau and G. Laporte. The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. *4OR: A Quarterly Journal of Operations Research*, 1(2), 2003.
5. W. E. de Paepe, J. K. Lenstra, J. Sgall, R. A. Sitters, and L. Stougie. Computer-Aided Complexity Classification of Dial-a-Ride Problems. *Informs J. Comp.*, 16(2):120–132, 2004.
6. G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha. Covering Graphs Using Trees and Stars. In *APPROX-RANDOM*, pages 24–35, 2003.
7. J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC*, pages 448–455, 2003.
8. G. N. Frederickson, M. S. Hecht, and C. E. Kim. Approximation algorithms for some routing problems. *SIAM J. Comput.*, 7(2):178–193, 1978.
9. I. L. Gørtz. Hardness of Preemptive Finite Capacity Dial-a-Ride. In *APPROX-RANDOM*, pages 200–211, 2006.
10. A. Gupta, M. Hajiaghayi, V. Nagarajan, and R. Ravi. Dial a Ride from k-forest. In *ESA*, pages 241–252, 2007.
11. M. Haimovich and A. H. G. R. Kan. Bounds and heuristics for capacitated routing problems. *Math. Oper. Res.*, 10:527–542, 1985.
12. D. Hochbaum and W. Maass. Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI. *J. ACM*, 32:130–136, 1985.
13. S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning and shortest path trees. In *SODA*, pages 243–250, 1993.
14. P. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *STOC*, pages 682–690, 1993.
15. S. Mitrović-Minić and G. Laporte. The Pickup and Delivery Problem with Time Windows and Transshipment. *INFOR Inf. Syst. Oper. Res.*, 44:217–227, 2006.
16. C. Mues and S. Pickl. Transshipment and time windows in vehicle routing. In *8th Int. Symp. on Parallel Architectures, Algorithms and Networks*, pages 113–119, 2005.
17. Y. Nakao and H. Nagamochi. Worst case analysis for pickup and delivery problems with transfer. *IEICE Trans. on Fund. of Electronics, Comm. and Computer Sci.*, E91-A(9), 2008.
18. M. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29:17–29, 1995.
19. S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computer & Operations Research*, 33:894–909, 2006.