Routing a Fleet of Automated Vehicles in a Capacitated Transportation Network

Martin Schaefer, Michal Čáp, Jan Mrkos and Jiří Vokřínek
Computer Science Department, Faculty of Electrical Engineering,
Czech Technical University in Prague
Email: {name.surname}@fel.cvut.cz

***Abstract*— Routing of a fleet of automated unit-occupancy vehicles in a capacitated transportation network is an emerging problem that needs to be addressed to realize large-scale automated transportation systems. We adopt an existing network-flow-based model for the problem and present a new reformulation based on Dantzig-Wolfe decomposition. This reformulation allows us to apply the column generation solution technique which, in turn, enables us to solve large-scale problem instances with tens of thousands of requests on networks with thousands of links. We empirically compare our method to the state-of-the-art approach on several standard benchmark instances and find that the computational time of our solution approach scales qualitatively better in all tested problem instance parameters: namely, in the size of the transportation network, in the magnitude of demand intensity, and in the number of demand flows.**

I. Introduction

Advancements in robotics and automation enabled the realization of automated transportation systems, where large fleets of automated vehicles are used to transport material or people. Today, automated transportation systems are commonly deployed in industrial environments to move material between workstations or to fetch and store material in a warehouse. In future, automated transportation systems may be deployed in the urban environment to provide affordable on-demand mobility to people within mobility-as-a-service schemes. Some of these transportation systems are envisioned to employ tens of thousands of automated vehicles [1].

In automated transportation systems, the vehicles typically move in a transportation network, often modeled as a graph of links and intersections. Depending on a deployment scenario, a transportation network can be a road network, a network of guiding strips, or simply a network of virtual lanes. The links and intersections of the network are typically capacitated, i.e., there is a limited number of robots that can travel along a specific link or intersection in a fixed time interval. Consequently, when the automated transportation system consists of a large number of vehicles, it needs to coordinate the routes of individual vehicles in order to prevent the formation of congestion in the underlying transportation network.

The problem of fleet routing in a capacitated transportation network can be informally described as follows: Consider an automated transportation system consisting of a fleet of automated vehicles that jointly serve requests for transportation between two points at the transportation network. We assume that there is no other traffic competing for the capacity, i.e., the considered transportation network (that might be a reserved fraction of the overall network capacity) is exclusively occupied by the centrally-controlled fleet.

We model the transportation network by a directed graph, where the nodes represent junctions or parking places and the edges represent the transportation links. The road links have time-invariant capacities describing the maximal flow rate along the link measured in the number of vehicles per time unit. We assume that if the flow over the link is below its capacity, the congestion will not occur and consequently the transit times over the link are time-invariant and deterministic. The transportation demand is a collection of transportation requests for transportation of *payload*, which could be, e.g., a person, a storage shelf, some standard unit of material, etc. Each transportation request is specified by the earliest pick-up time, the origin node, and the desired destination node. Each vehicle can carry one unit of payload. A request is served when a vehicle picks up the payload at the origin node and subsequently drops-off the payload at the destination node. Note that vehicles need to be preserved between requests and thus they need to drive without payload from the destination node of one request to origin node of a potential subsequent request. The vehicles can only park at the dedicated parking nodes. The goal is to assign a route to each vehicle (a sequence of nodes to visit in the transportation network) so that each transportation request is serviced within provided time constraints, the link capacities are not exceeded, and at the same time, the desired performance criterion is optimized.

The problem of fleet routing in a capacitated network is related to several established research areas. First, the problem of multi-robot trajectory coordination is dealt with in the area of multi-robot path planning [2], [3], but due to inherent combinatorial complexity, the existing solution techniques typically target only small-scale scenarios with at most a few tens of robots. Second, the choice of the optimal vehicle-request assignment is central to vehicle routing problems. Yet, the existing models for VRP problems do not consider endogenous congestion effects, i.e., it is implicitly assumed that the size of the fleet is small compared to road capacities and consequently the contribution of fleet vehicles to congestion is negligible [4], [5]. Third, the choice of routes for a large number of vehicles in a road network is addressed in traffic assignment and route guidance subfields of transportation research [6], [7], [8]. Yet, these areas implicitly assume privately-owned vehicles, and thus they do not deal with the vehicle-request assignment problem. The closest work to ours is [9], where the authors propose a generalization of traffic assignment models that includes considerations for fleet-request assignment by incorporating vehicle rebalancing flow. The resulting linear programs for a system of non-trivial size, however, tend to be too large to be solvable with existing off-the-shelf LP solvers.

Our contribution is the application of the column generation approach to the problem of fleet routing in a capacitated transportation network. We manage this by reformulating the problem using the Dantzig-Wolfe decomposition. Our experimental analysis reveals that the solution approach based on column generation scales significantly better in relevant parameters of the problem than the solution approach proposed in the previous work.

This paper is organized as follows. The next section reviews literature in the related research domains. The section III introduces the fleet routing in a capacitated transportation network problem. The solution approaches to this problem are introduced in Section IV. In particular, we first introduce the standard edge-based formulation in Section IV-A and its improved variant with bundling in Section IV-B. Finally, Section V formulates our column generation approach. The empirical comparison follows in Section VI to conclude the paper in Section VII.

II. Related Work

The problem of fleet routing in a capacitated transportation network is related, but distinct to vehicle routing and traffic assignment problems:

The research field of vehicle routing problems (VRP) [10], [4] has focused on finding the optimal assignment of transportation requests to vehicles in a fleet and on determining the order in which the vehicle should visit the requests. In result, the majority of VRP algorithms require no information about the transportation network and instead the input is presented as a *customer-based graph* that contains links and corresponding travel time between each origin position, each destination position, and each vehicle position. However, in certain contexts, road network information can be leveraged, e.g., when there are multiple optimization objectives or when routing cleaning vehicles that need to cover the road segments of the network [5]. Yet, to our knowledge, none of the VRP formulations considers throughput constraints on the road links.

The road network is naturally considered in traffic assignment [6] problems. System-optimal traffic assignment methods aim to compute the coordinated routes for all vehicles in the system such that the average travel delay is minimized. Typically, the number of vehicles travelling through a network is huge and therefore, the vehicles that share the same origin node and the same destination node are aggregated and considered as one "vehicle flow" / "commodity". The limitation of these models is that they make no distinction between requests (passengers) and the vehicles that serve the requests. Note that in contrast to personal transportation by privately-owned vehicles, where each request (driver) is tied to its vehicle, in fleet routing problem we have a shared fleet of vehicles, and thus we can also optimize over the possible assignments of requests to vehicles.

Recently, Zhang et al. [9] generalized traffic assignment approach and formulated the problem of fleet routing in a capacitated network as a multi-commodity network-flow problem. In their model, all transportation requests starting in the same origin node and going to the same destination node are aggregated and considered as one "commodity" to be served by an appropriate number of vehicles. Empty "rebalancing" vehicles have to drive from nodes with a surplus of vehicles to nodes with a shortage of vehicles to ensure vehicle availability in the system and they are modeled as another, multi-source multi-destination commodity. The system is analyzed in steady-state, i.e., transportation requests for each origin-destination pair are assumed to be generated by a periodic process with given intensity. Subsequently, a linear program is solved to obtain minimum-cost flows for all such commodities that respect the capacity constraints of the road network. In the final step, vehicles are assigned routes that are consistent with the computed commodity flows.

The network-flow model proved useful for analytic exploration of the structural properties of the fleet routing problem in urban transportation networks [9]. However, the main drawback of the proposed network-flow approach is that the resulting linear programs are too large to be efficiently solvable with general solution methods. In fact, even small routing problems on graphs with dozens of nodes and dozens of commodities may take hours to solve. The size of the resulting linear program can be reduced by bundling commodities with the same destination [11]. Yet, even when commodity bundling is applied, large-scale fleet routing instances cannot be generally solved in practical time.

Solving the problem of fleet routing in capacitated networks optimally and efficiently, therefore, requires a different approach. Dantzig-Wolfe decomposition followed by column generation is a solution approach that can efficiently solve large-scale linear programs that possess favourable (specifically, block angular) structure [12]. It has proven as a successful solution methodology for various domains. In particular, it has been successfully applied to solve instances of min-cost multicommodity flow problems, e.g., [13], [14]. In this paper, we demonstrate that when the model proposed by [9] is appropriately reformulated, then the column generation approach can be exploited to find optimal solutions for non-trivial problem instances in practical time.

III. Problem Formulation

In this section, we formulate the fleet routing in a capacitated transportation network problem (FRCTN). Our formulation corresponds to the congestion-aware fleet routing model used in [9] with some changes in notation.

Consider a transportation system consisting of a fleet of single-occupancy vehicles that jointly serve periodic node-to-node transportation requests over a road network. We assume that there is no other traffic, i.e., the road network is exclusively occupied by the centrally-controlled fleet.

The road network is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, c, \delta)$, where $\mathcal{V}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. Nodes correspond to road junctions or parking places and edges to the road links. The road links have time-invariant capacities describing the maximal flow rate along the link measured in the number of vehicles per time unit. The capacity of a road link $(u, v) \in \mathcal{E}$ is denoted as $c(u, v) \in \mathbb{R}_{>0}$. The length of edge (u, v) is denoted $\delta_{(u,v)} \in \mathbb{R}_{\geq 0}$.

We model requests using the notion of commodities. Commodity is an origin destination pair. The set of commodities is $\mathcal{M} = \{(s_i, g_i)\}_{i=1,...,M}$, where for the i-th request flow, $s_i \in \mathcal{V}$ is the origin of the commodity, $g_i \in \mathcal{V}$ is the destination of the commodity. The cardinality of the set $\mathcal{M}$ is denoted by M.

We model the system in steady state and assume that transportation requests are generated by a collection of periodic processes with time-invariant intensities. Specifically, the demand for transportation from node s_m to node g_m (that is, a commodity m) appears periodically with frequency d_m, measured in the number of requests per time unit.

The fleet size that is simultaneously needed in the steady state system is given by the sum of the transportation requests and overall imbalance both expressed in vehicles per time unit.

We distinguish two types of vehicle flows: demand flows f_m and rebalancing flows f_R. The demand flows correspond to parts of vehicle routes with passenger or material on board, and the rebalancing flows correspond to the parts of routes when vehicles drive empty.

The flow rate on the edge $(u, v) \in \mathcal{E}$ for the demand flow m is denoted as $f_m(u, v)$. The rebalancing flow rate on the edge (u, v) is denoted by $f_R(u, v)$. Additionally, for $\circ \in \mathcal{M} \cup \{R\}$, we define flow rate from a node v as $f^+_\circ(v) = \sum_{\{u|(v,u)\in\mathcal{E}\}} f_\circ(v, u)$ and similarly $f^-_\circ(v)$ as flow rate into node v.

The optimization criterion is the cost of fleet operation, i.e., the total distance traveled by the vehicles in the fleet:

$$\Gamma(f) := \sum_{(u,v)\in\mathcal{E}} \delta(u, v)\Big(\sum_{m\in\mathcal{M}} f_m(u, v) + f_R(u, v)\Big)$$

Apart from the edge capacities, the problem solution has to satisfy additional constraints: 1) no vehicles can enter or exit the system, 2) all demand has to be satisfied, and 3) the vehicles can not accumulate in any node of the system (rebalancing constraint).

The goal is to determine routes for the vehicles in the fleet through the road network such that the above constraints are satisfied.

Note that fleet routing in a capacitated transportation network is a relevant problem only for a system that serves transportation demand that is so large that the system becomes constrained by the road capacities. Therefore, we can safely assume that the fleet consists of a vast number of vehicles. For example, Fiedler et al. [15] shown that to serve transportation demand during the morning peak in Prague, one would need to coordinate more than 50 000 vehicles that operate simultaneously on the road.

IV. Solution Methods

In this section, we present the reference solution approach for fleet routing in a capacitated transportation network problem based on the edge-based formulation and its improved variant with commodity bundling. Next, we formulate our column generation approach forming the main contribution of this paper.

A. Edge-based LP Formulation

In this section, we present a solution method for FRCTN based on reformulation to multi-commodity min-cost flow problem. This method roughly corresponds to the solution approach introduced by Zhang et al. in [9].

In this approach, the transportation requests that have the same origin node and the destination node are considered as a demanded commodity flow. This, however, results in a number of commodities that is quadratic in the number of nodes of the road graph. In practice, to reduce the number of considered commodities, the travel requests that originate in one region and have their destination in another region can be aggregated and represented by one commodity with larger demand flow.

Now we are in the position to cast the problem of fleet routing in a capacitated transportation network as an instance of minimum cost multi-commodity flow problem:

Problem 1. *Edge-based LP formulation of FRCTN:*

$$\underset{\{f_m\}, f_R}{\arg\min} \sum_{(u,v)\in\mathcal{E}} \delta(u, v)\Big(\sum_{m\in\mathcal{M}} f_m(u, v) + f_R(u, v)\Big) \qquad \textit{s.t.}$$

$$f^+_m(u) = f^-_m(u) \qquad \forall u \in \mathcal{V} \setminus \{s_m, g_m\} \qquad \forall m \in \mathcal{M}, \quad (1)$$

$$f^+_m(s_m) = d_m \qquad \forall m \in \mathcal{M}, \quad (2)$$

$$f^-_m(g_m) = d_m \qquad \forall m \in \mathcal{M}, \quad (3)$$

$$f^-_R(v) + \sum_{\{m|g_m=v\}} d_m = f^+_R(v) + \sum_{\{m|s_m=v\}} d_m \qquad \forall v \in \mathcal{V}, \quad (4)$$

$$\sum_{m\in\mathcal{M}} f_m(u, v) + f_R(u, v) \leq c(u, v) \qquad \forall (u, v) \in \mathcal{E}. \quad (5)$$

The objective is to minimize the total distance traveled by the vehicles, i.e., the cost of the fleet operation. The constraints enforce that the demand flows are conserved (1), requests are satisfied (2, 3), rebalancing flows in the road network transform to demand flows, and vice versa, without loss[1](4), and the capacity of road links is not exceeded (5). The flow rates non-negativity is assumed implicitly.

[1]Note that rebalancing flows origin in demand destinations by dropping off transported commodity and sink by picking-up transported commodity.

B. Commodity Bundling

Rossi et al. [11] observed that the equivalent optimal solution could be obtained by a smaller linear program, where the demand flows with the same destination are bundled together, i.e., they are considered to be indistinguishable.

A bundled demand b is a set of demand commodities with the same destination node g_b, i.e., $m \in b \iff g_m = g_b$. We denote the set of all bundled demands as $\mathcal{B}$.

Using bundled demand, we define $f_b(u,v) := \sum_{m \in b} f_m(u,v)$, Functions f^+ and f^- are extended correspondingly.

For problems with one commodity for each origin-destination node pair, where $\mathcal{M} = \mathcal{V} \times \mathcal{V}$, bundling reduces the number of flow variables from $|\mathcal{V}|^2$ to $|\mathcal{V}|$. Authors of the technique report speed-ups of three orders of magnitude on the example problem compared to the edge-based formulation without bundling [11].

V. Proposed method with Column Generation

In this section, we introduce a solution method to FRCTN based on Dantzig-Wolfe decomposition and column generation [16]. The main idea of the approach is the following: First, we change the representation from the edge-based model to path-based model. In the path-based model, we introduce a variable that encodes how many vehicles travel along each possible (simple non-branching) path from source to sink. Clearly, the number of paths can be exponential in the size of the graph and thus it is not practically possible to enumerate all such paths. Therefore, we will use the column generation approach to add variables only when they need to enter the basis in order to improve the value of the objective function.

Path-based LP formulation for FRCTN can be developed as follows. First, we represent an origin-destination flow of a single commodity m as a collection $f_m = \{f_m(u,v)\}_{(u,v)\in\mathcal{E}}$, where $f_m(u,v)$ is the flow of commodity m over edge (u,v). For a flow f_m to be feasible it must satisfy the conditions (1), (2), (3) from Problem 1, i.e., the flow has to start at node s_m, end at node g_m, respect flow conservation constraints, and have the total intensity of d_m. Let F^m denote the set of all feasible flows for commodity m. Then, we can define $P^m = \{P^m_1, \ldots, P^m_{|P^m|}\}$ to be the set of all extreme points of set F^m. Or, in other words, P^m is a set of all paths (or more precisely trivial non-branching flows of intensity d_m) starting at s_m and ending at g_m. We can observe that any point $f \in F^m$ can be expressed as a convex combination of points from P^m, i.e.,

$$f = \sum_{i=1}^{|P^m|} \lambda_i P^m_i \text{ with } \sum_{i=1}^{|P^m|} \lambda_i = 1.$$

Then, with the notation $P^m_i(u,v)$ representing i-th flow intensity of commodity m on edge $(u,v) \in \mathcal{E}$, the path-based LP can be formulated as follows:

Problem 2. *Path-based LP formulation of FRCTN*

$$\underset{\{\lambda^m_p\}, f_R}{\arg\min} \sum_{(u,v)\in\mathcal{E}} \delta(u,v) \Big(\sum_{m\in M} \sum_{i=1}^{|P^m|} \lambda^m_i \cdot P^m_i(u,v) + f_R(u,v) \Big)$$

subject to

$$\sum_{m\in M} \sum_{i=1}^{|P^m|} \lambda^m_i \cdot P^m_i(u,v) + f_R(u,v) \le c(u,v) \quad \forall (u,v) \in \mathcal{E}, \tag{6}$$

$$\sum_{i=1}^{|P^m|} \lambda^m_i = 1 \qquad \forall m \in \mathcal{M}, \tag{7}$$

$$\lambda^m_i \ge 0 \qquad \forall m \in \mathcal{M} \forall i \in 1, \ldots, |P_m|, \tag{8}$$

$$f^-_R(v) + \sum_{\{m|g_m=v\}} d_m = f^+_R(v) + \sum_{\{m|s_m=v\}} d_m \qquad \forall v \in \mathcal{V}. \tag{9}$$

Analogously to Problem 1, the objective is to minimize the total distance travelled by the vehicles. The constraints enforce that the capacities of each link must not be exceeded (6), the flow must be a convex combination of extreme points (7, 8), and the vehicles must be balanced (9).

The sets $P^1, \ldots, P^m$ are in most cases too large to allow explicit construction of the above linear program. However, one may use the column generation technique to generate these sets iteratively.

Let $(\{\lambda^{*m}_i\}, \{f^{*R}(u,v)\}, \mathbf{w}, \mathbf{y}) = \mathrm{RMP}(P^1, \ldots, P^m)$ denote the solution of Problem 2 with particular sets of extreme points $P^1, \ldots, P^m$, i.e., the solution to so-called restricted master problem (RMP). The solution process returns the optimal values of the optimization variables $\{\lambda^{*m}_i\}$ and $\{f^{*R}(u,v)\}$, and the dual values $\mathbf{w} \in \mathbb{R}^{|\mathcal{E}|}$ and $\mathbf{y} \in \mathbb{R}^{|\mathcal{M}|}$ corresponding to constraints (12) and (13) respectively.

Given the dual values, we solve so-called subproblem for each commodity m to determine if adding any extreme point (a new column) to P^m could improve the objective value of the reduced master problem. The subproblem for commodity m is defined as

$$\mathrm{SP}^m(\mathbf{w} = \{w(u,v)\}) :=$$
$$\underset{f\in F^m}{argmin} \sum_{(u,v)\in\mathcal{E}} (\delta(u,v) - w(u,v)) \cdot f(u,v) =$$
$$\underset{p\in P^m}{argmin} \sum_{(u,v)\in\mathcal{E}} (\delta(u,v) - w(u,v)) \cdot p(u,v).$$

Notice that to solve this optimization problem, it is sufficient to find a shortest path from s_m to g_m in a graph with the cost on an edge (u,v) defined as $\delta(u,v) - w(u,v)$. For a path p computed by a subproblem, $c(p)$ will denote its cost.

The pseudocode of the column generation algorithm is exposed in Algorithm 1. The algorithm starts with a single extreme point for each commodity that corresponds to the shortest path from s_m to g_m. Then, it solves the restricted master problem to determine the dual values on the capacity constraints of each edge. By subtracting the dual (which has non-positive value), the cost on road links where the capacity constraint is active is increased. Then, we solve subproblem

Algorithm 1: Column generation

$P^m \leftarrow \{\mathrm{SP}^m(\mathbf{0})\}, \quad \forall m \in \mathcal{M}$;
do
  $(\{\lambda_i^{*m}\}, \{f^{*R}(u,v)\}, \mathbf{w}, \mathbf{y}) = \mathrm{RMP}(P^1, \ldots, P^m)$;
  newcols $\leftarrow$ False;
  forall $m \in \mathcal{M}$ **do**
    $p \leftarrow \mathrm{SP}^m(\mathbf{w})$;
    if $c(p) - y^m < 0$ **then**
      $P^m \leftarrow P^m \cup \{p\}$;
      newcols $\leftarrow True$;
while newcols;
if $\mathrm{RMP}(P^1, \ldots, P^m)$ *is feasible* **then**
  return $\{f^m = \sum_{i=1}^{|P^m|} \lambda_i^{*m} P_i^m\}, f^{*R}$;
else
  return infeasible;

for each commodity to see if there is any commodity that can use an alternative cost-improving path. All such paths are then added to the set of extreme points. This process is iteratively repeated until no more improving paths can be found, which certifies that the current solution is optimal. The intermediate solutions are mostly infeasible (flows over capacity are handled by the capacity slacks) and typically only few last iterations are feasible but suboptimal until the optimal one is found.

The resulting steady-state flows then also reveal the number of vehicles needed to serve the assumed demand. The number of vehicles to serve a flow is given by the product of the flow intensity and the flow duration. The vehicles of all the flows then form a steady-state circulation in network with intensities matching the intensity of the demand requests.

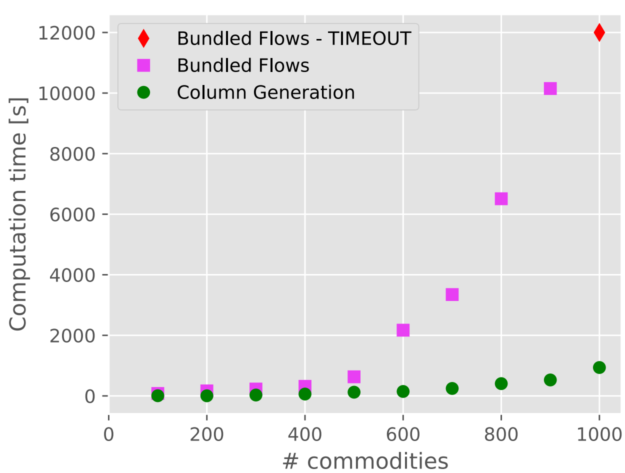

Fig. 1. Performance of bundled and CG methods on grid benchmark with 625 nodes, 2400 edges and varying number of commodities.

VI. Empirical evaluation

In this section, we compare the scalability of edge-based formulation enhanced by commodity bundling (Section IV-B) with the proposed method based on column-generation (Section V). The improved variant with commodity bundling (Section IV-B) dominates the basic edge-based formulation (Section IV-A). Therefore we consider only the improved variant in the evaluation.

Since both compared methods produce the optimal solution, we do not explicitly compare the quality of results in terms of achieving an optimal solution but the running time and ability to solve the benchmark instance in defined time.

A. Benchmark instances

Since there are no standard benchmarks for FRCTN, we compared the two algorithms on graphs and with corresponding transportation demands from standard benchmark instances for a related multi-commodity min-cost flow problem. This benchmark set was originally presented at [17] and the instances are available for download from `http://www.di.unipi.it/optimize/Data/MMCF.html`.

The benchmark set contains grid and planar networks. We tested grid networks with 80 to 3480 edges and 50 to 6000 demand commodities. The planar networks have 150 to 4388 edges and 92 to 12756 demand commodities. For more details of the instances see Table I.

B. Implementation details

We implemented both solution methods in Python. The linear programs were modeled using PuLP and solved by COIN-OR LP Solver (CLP). We obtained the presented results on a machine with AMD Phenom II X4 945 processor and 32 GB RAM.

Further, the both approaches use the active set strategy [18] to reduce the number of capacity constraints. The idea behind the technique is that in an optimal solution to the flow problem, only a subset of edges will have their capacity constraints active. Hence, it would be enough only to include the constraints associated with these edges in the problem. However, because we can not know beforehand which constraints are going to be active, we add them iteratively during restricted master problem creation. In the first iteration, we solve the flow problem without capacity constraints. Next, the algorithm adds the capacity constraints to the edges where flow exceeds capacity. Then, it solves the flow problem with the added constraints. This process is repeated until no new constraints are added. Then column generation procedure reads the values of dual variables and continues by solving the subproblems.

The subproblems in the column generation method are solved using the Dijkstra's algorithm implementation from NetworkX library.

C. Results

To compare the scalability of the edge-based formulation with bundling and our approach based on column generation, we run both algorithms on each problem instances from the

Graph	# Nodes	# Edges	# Commodities	Demand	Time - Bundled [s]	Time CG [s]	CG iterations	% of CG active sets
Grid	25	80	50	2833	**0.6**	1.2	6	23.8
Grid	25	80	100	5642	**0.7**	2.9	7	57.5
Grid	100	360	50	2856	4.2	**2.9**	5	7.8
Grid	100	360	100	5484	8.7	**5.2**	6	17.5
Grid	225	840	100	5595	26.0	**17.6**	9	7.9
Grid	225	840	200	11261	66.3	**46.1**	14	26.5
Grid	400	1520	400	22349	331.9	**139.0**	20	15.9
Grid	625	2400	500	27782	4,530.6	**389.8**	21	23.5
Grid	625	2400	1000	56347	T/O	**956.2**	21	29.3
Grid	625	2400	2000	111950	T/O	**1743.4**	19	28.0
Grid	625	2400	4000	224787	T/O	**2267.7**	19	19.3
Grid	900	3480	6000	330946	T/O	**2302.9**	15	11.8
Planar	30	150	92	704	**1.1**	2.4	5	21.3
Planar	50	250	267	1756	**4.0**	10.0	7	27.2
Planar	80	440	543	3612	**14.9**	46.0	10	40.5
Planar	100	532	1085	6987	**23.9**	90.0	11	26.9
Planar	150	850	2239	17876	**180.2**	577.8	14	54.9
Planar	300	1680	3584	23354	599.1	**577.0**	11	14.2
Planar	500	2842	3525	24750	1,236.2	**454.6**	9	4.3
Planar	800	4388	12756	69879	T/O	**7780.6**	17	6.4

TABLE I

COMPARISON OF PERFORMANCE OF BUNDLED AND COLUMN GENERATION ALGORITHMS ON VARIOUS GRID AND PLANAR BENCHMARKS.

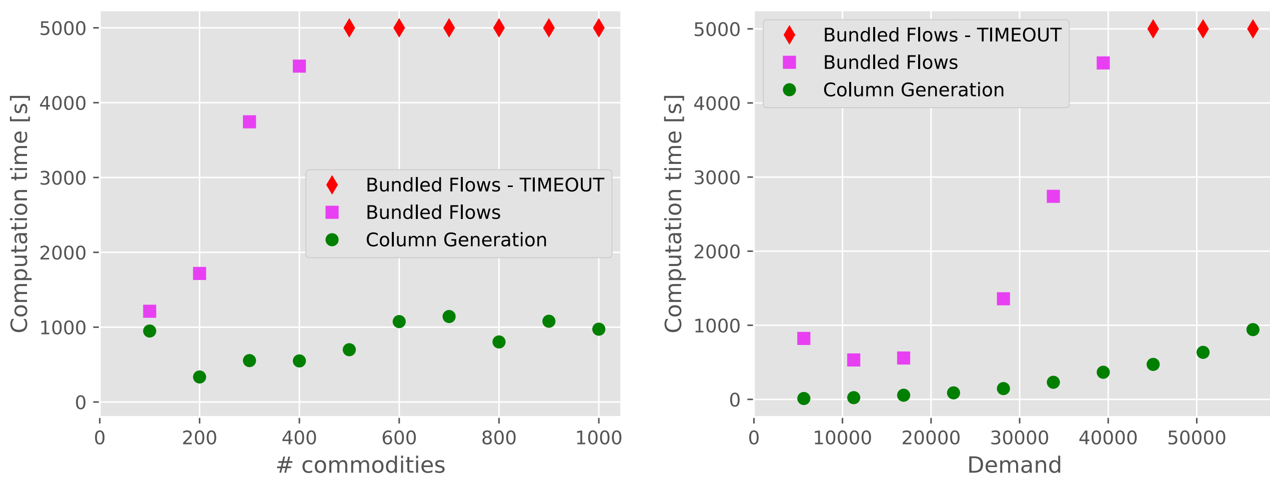

Fig. 2. Performance of bundled and CG methods on grid benchmark with 625 nodes, 2400 edges. In the plot on the left, the total demand was kept between between 51 and 57 thousand while varying the number of commodities. In the plot on the right, the total demand was varied and number of commodities was kept at 1000.

benchmark set with runtime limit of 9000 seconds. For each algorithm, We record the runtime needed to find an optimal solution. For column generation approach, we also keep track of the number of the outer iterations of the algorithm 1 (*CG iterations*) and of the percentage of capacity constraints that are in active set after the final iteration of the algorithm (*% of CG active sets*). The results are summarized in Table I.

The table shows that for very small instances, the edge-based approach with bundling outperforms column generation method. The fixed overhead is caused by the repeated problem construction and solver calls for each CG iteration. However, for larger instances, the CG method outperforms the edge-based approach with bundling. In fact, for larger graphs with thousands of nodes and edges, the edge-based formulation with bundling failed to return a solution within the given runtime limit.

To obtain further insights into the scalability in relation to the structure and size of the demand, we took the grid instance with 625 nodes, 2400 edges and 1000 commodities and generated a series of instances with $100, 200, \ldots, 1000$ commodities by selecting an appropriate subset of commodities. Note that the total size of demand scales linearly with the number of commodities. The results of this experiment[2] are shown in Figure 1. We can see that the proposed column generation method scales significantly better with the number of commodities than the edge-based method with bundling.

Varying the number of commodities also varies the total amount of demand, which strongly influences the average saturation of the network. To isolate the effects of increased

[2]Runtime limit was extended to 12,000 s

network saturation from the effect of larger problem size, we run two additional experiments. We again take the grid instance with 625 nodes, 2400 edges and 1000 commodities. In the first experiment, we vary the number of commodities from 100 to 1000, but re-scale the demand intensities accordingly so that the total demand remains approximately constant (at approx. 56 000). In the second experiment, we keep 1000 commodities but vary their intensities so that the total demand varies from 5 000 to 56 347 (the full demand of the original benchmark instance). The results of the two experiments are shown in Figure 2. The runtime limits are set to 5000 s for this experiment.

Comparing the plots in Figures 1 and 2, we can see that the column generation approach scales much better than the edge-based method with bundling in both the number of commodities and the total size of demand. Overall, the proposed method path-based formulation of FRCTN with column generation significantly outperforms the bundling approach, the state-of-the-art method for the FRCTN.

VII. Conclusion

The advancements in robot autonomy enabled deployment of large scale automated transportation systems. We study the problem of a routing of a fleet of automated vehicles in a transportation network with constrained link capacities. As a starting point, we use an existing edge-based network-flow formulation of the problem that, however, cannot be efficiently solved using general solution methods. To address this limitation, we proposed a path-based reformulation based on Dantzig-Wolfe decomposition and empirically demonstrated that by using this reformulation, it is possible to solve large problem instances with thousands of demand commodities on a network with thousands of edges. Our experimental analysis revealed that the computational time needed to find optimal solution with the proposed solution methods scales qualitatively better than the state-of-the-art solution method in three relevant problem instance parameters: in network size, in demand magnitude, and also in the number of demand commodities.

In future work, we plan to evaluate the efficiency of the proposed modeling approach in the large-scale multi-robot simulation. Moreover, we plan to relax the steady-state assumption and generalize the framework to more accurately model environments with time-dependent or non-periodic demand.

Acknowledgments

This research was supported by the Czech Science Foundation (grant No. 18-23623S) and OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16˙019/0000765 ”Research Center for Informatics”.

References

[1] R. Zhang, K. Spieser, E. Frazzoli, and M. Pavone, “Models, algorithms, and evaluation for autonomous mobility-on-demand systems,” in *2015 American Control Conference (ACC)*. IEEE, 7 2015, pp. 2573–2587. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7171122

[2] M. Cap, P. Novak, A. Kleiner, and M. Selecky, “Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 835–849, 7 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7138650/

[3] T. Standley and R. Korf, “Complete algorithms for cooperative pathfinding problems,” *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume One*, pp. 668–673, 2011. [Online]. Available: https://dl.acm.org/citation.cfm?id=2283396.2283505

[4] P. Toth and D. Vigo, Eds., *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 11 2014. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9781611973594

[5] H. Ben Ticha, N. Absi, D. Feillet, and A. Quilliot, “Vehicle routing problems with road-network information: State of the art,” *Networks*, vol. 72, no. 3, pp. 393–406, 10 2018. [Online]. Available: http://doi.wiley.com/10.1002/net.21808

[6] O. Jahn, R. H. Mhring, A. S. Schulz, and N. E. Stier-Moses, “System-Optimal Routing of Traffic Flows with User Constraints in Networks with Congestion,” *Operations Research*, vol. 53, no. 4, pp. 600–616, 2005.

[7] E. Angelelli, I. Arsik, V. Morandi, M. Savelsbergh, and M. Speranza, “Proactive route guidance to avoid congestion,” *Transportation Research Part B: Methodological*, vol. 94, pp. 1–21, 12 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261515300357?via%3Dihub

[8] E. Angelelli, V. Morandi, and M. Speranza, “Congestion avoiding heuristic path generation for the proactive route guidance,” *Computers & Operations Research*, vol. 99, pp. 234–248, 11 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054818301928

[9] R. Zhang, F. Rossi, and M. Pavone, “Routing Autonomous Vehicles in Congested Transportation Networks: Structural Properties and Coordination Algorithms,” in *Robotics: Science and Systems XII*, vol. 42, no. 7. Robotics: Science and Systems Foundation, 3 2016, pp. 1427–1442. [Online]. Available: http://www.roboticsproceedings.org/rss12/p32.pdfhttp://dx.doi.org/10.15607/RSS.2016.XII.032

[10] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2012.08.015

[11] F. Rossi, R. Iglesias, M. Alizadeh, and M. Pavone, “On the interaction between Autonomous Mobility-on-Demand systems and the power network: models and coordination algorithms,” in *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, 6 2018. [Online]. Available: http://www.roboticsproceedings.org/rss14/p37.pdf

[12] J. Gondzio, P. González-Brevis, and P. Munari, “Large-scale optimization with the primal-dual column generation method,” *Mathematical Programming Computation*, vol. 8, no. 1, pp. 47–82, 3 2016. [Online]. Available: http://link.springer.com/10.1007/s12532-015-0090-6

[13] W. DAI, J. ZHANG, and X. SUN, “On solving multi-commodity flow problems: An experimental evaluation,” *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1481–1492, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.cja.2017.05.012

[14] J. Rios and K. Ross, “Massively Parallel Dantzig-Wolfe Decomposition Applied to Traffic Flow Scheduling,” *Journal of Aerospace Computing, Information, and Communication*, vol. 7, no. 1, pp. 32–45, 1 2010. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.45606

[15] D. Fiedler, M. Cap, and M. Certicky, “Impact of mobility-on-demand on traffic congestion: Simulation-based study,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 10 2017, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/8317830/

[16] G. B. Dantzig and P. Wolfe, “Decomposition Principle for Linear Programs,” *Operations Research*, vol. 8, no. 1, pp. 101–111, 2 1960. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/opre.8.1.101

[17] T. Larsson and D. Yuan, “An Augmented Lagrangian Algorithm for Large Scale Multicommodity Routing,” *Computational Optimization and Applications*, vol. 27, no. 2, pp. 187–215, 2 2004. [Online]. Available: http://link.springer.com/10.1023/B:COAP.0000008652.29295.eb

[18] F. Babonneau, O. du Merle, and J.-P. Vial, "Solving Large-Scale Linear Multicommodity Flow Problems with an Active Set Strategy and Proximal-ACCPM," *Operations Research*, vol. 54, no. 1, pp. 184–197, 2 2006. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/opre.1050.0262