Large scale column generation for multi commodity flow problems: Application to transport optimization

Benedikt Lienkamp[a,*], Maximilian Schiffer[a,b]

[a] TUM School of Management, Technical University of Munich, Munich, Germany
benedikt.lienkamp@tum.de, schiffer@tum.de

[b] Munich Data Science Institute, Technical University of Munich, Munich, Germany

[*] Corresponding author

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

April 4, 2022

Keywords: multi commodity flow, column generation, time expanded networks

1 INTRODUCTION

According to the United Nations, the global urban population will increase by 900 million from 2020 to 2030, reaching a total of 5.2 billion (UN, 2018). This population increase in combination with already high congestion in cities all around the world calls for novel transportation concepts to keep up with future urban travel demand. The rise of autonomous vehicles and shared intermodal transportation modes offer an opportunity to mitigate the rising congestion by smarter route choice. To enable smart utilization of these novel transportation modes it is imperative to be able to optimize route planning in such a large system. In this context, we will focus on optimizing passenger flow in a capacitated network with fixed vehicle routes.

A number of approaches exist to optimize route planning in large systems. The concept of partially time-expanded networks has been successfully applied to service network design problems (Boland *et al.* , 2017) and allows for iterative refinement to obtain a reduced model size. Even though this concept can be used to solve service network design problems optimally, it is not directly applicable to solve large instances. Simulation-based models capture transportation systems with high accuracy, but are generally not amenable to efficient optimization. Network flow models are amenable to efficient optimization and capture transportation systems with high accuracy. Accordingly, they have for example been used to study the interaction of Autonomous Mobility-on-Demand (AMoD) with the public transportation system (PTS) (Salazar *et al.* , 2020) and the control of AMoD systems in congested road networks (Rossi *et al.* , 2018) on a mesoscopic level. To the best of our knowledge, so far no computational efficient algorithm exists that can be applied to optimize such network flow models for manifold passenger movements in very large systems.

Against this background we develop an algorithmic framework which uses a time expanded network as well as a spatial expansion to model multiple transportation modes. We define a minimum cost multi commodity flow problem (MCMCFP) to solve the routing of a set of travel requests optimally. The novelty in our approach lies in presenting a scalable column generation approach which solves the defined MCMCFP optimally. This column generation leverages a cyclic pricing approach in combination with a filtering mechanism to a improve computation time. Our preliminary results show that this algorithm is able to solve multi commodity flow problems with 3.137 trips in 23 seconds and with 31.371 trips in 9 minutes optimally.

2 Methodology

We seek to find optimal routes for a set of passengers in a capacitated transportation system with fixed vehicle routes, e.g., bus and subway lines. An instance of our problem contains the schedule of the fixed vehicle routes in our system and a set of passengers. For each route driven by a vehicle, the vehicle schedule contains information about the stops of the route, arriving times at the stops and capacity of the vehicle performing the route. For each passenger, trip information about the origin/destination coordinates and the departure time is known. We want to find paths for all passengers, such that the sum over all passenger travel times is minimized and all vehicle capacities constraints are kept.
We model such a transportation system as a multilayered digraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with a set of temporal vertices $\mathcal{V}$ and a set of temporal arcs $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$. Let $\mathcal{R}$ be the set of all routes and $\mathcal{S}$ be the set of stops in our transportation network. The graph G contains a route layer $G_r = (V_r, A_r)$ for each route $r \in \mathcal{R}$ and a waiting layer $G_s = (V_s, A_s)$ for each stop $s \in \mathcal{S}$. The route layers $G_{\mathcal{R}} = \bigcup_{r\in\mathcal{R}} G_r$ contain temporal vertices and arcs which represent the routes driven in our transportation network. The waiting layers $G_{\mathcal{S}} = \bigcup_{s\in\mathcal{S}} G_s$ contain temporal vertices and arcs which represent passengers waiting at a stop. Additionally we add a set of transit arcs A_T and walking arcs A_W. Transit arcs can be used to switch between different routes in the transportation network by moving from a route layer G_r, to a waiting layer G_s or from a waiting layer $G_{s'}$ to a route layer $G_{r'}$, with $r, r' \in \mathcal{R}$ and $s, s' \in \mathcal{S}$. Walking arcs allow passengers to walk between stops. Here, temporal vertices of waiting layers G_s and $G_{s'}$ of different stops $s, s' \in S$ are connected. The capacity of the route arcs corresponds to the capacity of the vehicle operating the route. All other arcs are unbounded. Collecting all definitions, it holds that $\mathcal{V} = V_{\mathcal{R}} \cup V_{\mathcal{S}}$ and $\mathcal{A} = A_{\mathcal{R}} \cup A_{\mathcal{S}} \cup A_T \cup A_W$. This graph represents the temporal and spatial expansion of the transportation network. Figure 1 shows a simplified example of the network graph's structure.
To formulate the passenger flow problem as a MCMCFP, we add a source and a sink vertex for each passenger trip to the graph G. Hereby, we connect the source ($\mathcal{S}^+$) and sink vertices ($\mathcal{S}^-$) to vertices of the waiting layers $G_{\mathcal{S}}$. This extends the vertex set of G to $\mathcal{V} = V_{\mathcal{R}} \cup V_{\mathcal{S}} \cup V_{\mathcal{S}^+} \cup V_{\mathcal{S}^-}$ and the arc set to $\mathcal{A} = A_{\mathcal{R}} \cup A_{\mathcal{S}} \cup A_T \cup A_W \cup A_{\mathcal{S}^+} \cup A_{\mathcal{S}^-}$. Each passenger travel demand is now represented as an individual flow going from a source vertex $v_{s^+} \in V_{\mathcal{S}^+}$ to a sink vertex $v_{s^-} \in V_{\mathcal{S}^-}$ with flow demand one.
Let c_{ij} be the travel time and u_{ij} the capacity of arc $(i, j) \in \mathcal{A}$, d_i^p the vertex demand of vertex $i \in \mathcal{V}$ and passenger $p \in \mathcal{P}$ and x_{ij}^p the decision variable, whether passenger $p \in \mathcal{P}$ uses arc $(i, j) \in \mathcal{A}$. The vertex demand is defined as

$$d_i^p = \begin{cases} 1 & \text{if vertex } i \in \mathcal{V} \text{ is the origin of passenger } p \in \mathcal{P}, \\ -1 & \text{if vertex } i \in \mathcal{V} \text{ is the destination of passenger } p \in \mathcal{P}, \\ 0 & \text{otherwise} \end{cases}$$

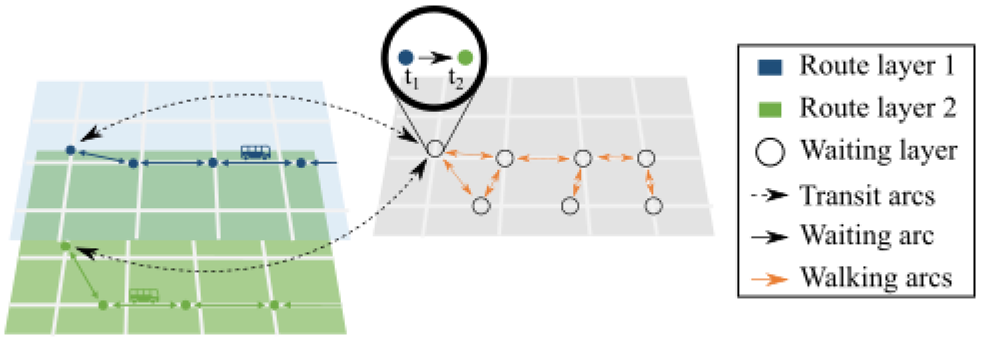


Figure 1 – *Scematic illustration of the multilayered network structure*

We can now formulate the continuous relaxation of finding the minimal sum of all passenger trip travel times as

$$\begin{aligned}
\min_x \quad & \sum_{p\in\mathcal{P}} \sum_{(i,j)\in\mathcal{A}} c_{ij}\; x_{ij}^p && && (1a)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{N}^+(i)} x_{ij}^p - \sum_{j\in\mathcal{N}^-(i)} x_{ji}^p = d_i^p, && i\in\mathcal{V}, p\in\mathcal{P} && (1b)\\
& \sum_{p\in\mathcal{P}} x_{ij}^p \le u_{ij}, && (i,j)\in\mathcal{A} && (1c)\\
& x_{ij}^p \ge 0, && (i,j)\in\mathcal{A}, p\in\mathcal{P} && (1d)
\end{aligned} \qquad \text{(LP 1)}$$

Here, the objective function models the sum over all passenger travel times. The constraint (1b) ensures flow conservation in the network with $\mathcal{N}^+(i)/\mathcal{N}^-(i)$ being the outgoing/ingoing neighbourhood of $i \in \mathcal{V}$. The remaining constraints enforce the capacity constraints of all vehicles and ensure positive passenger flows.
Since this LP 1 is not scalable for larger sets of passengers because of the exponential growth of the model formulation, we reformulate it to apply a column generation approach.

$$\begin{aligned}
\min_\lambda \quad & c^T Y\lambda && (2a)\\
\text{s.t.} \quad & Y\lambda \le u && (2b)\\
& \Lambda\lambda = 1_{|\mathcal{P}|} && (2c)\\
& \lambda \ge 0_{|\mathcal{P}|} && (2d)
\end{aligned} \qquad \text{(LP 2)}$$

In this LP 2 all possible flows for the passengers are stored as columns in the matrix $Y \in \mathbb{R}^{|\mathcal{A}|\times n}$, where n is the number of all possible flows. The vector $u \in \mathbb{R}^{|\mathcal{A}|}$ is the capacity vector of all arcs $a \in \mathcal{A}$ and the matrix $\Lambda \in \mathbb{R}^{|\mathcal{P}|\times n}$ is the incidence matrix between the flows of Y and the passengers $p \in \mathcal{P}$. This means that $\Lambda(i,j) = 1$, if the flow of the j-*th* column of Y corresponds to passenger i and 0 otherwise. The decision variable $\lambda \in \mathbb{R}^n$ is used to select a convex combination of all possible flows for each passenger. A convex formulation of valid passenger flows of passenger $p \in \mathcal{P}$ yields a new valid flow for passenger p.
We add dummy variables with high objective costs for each passenger, initiate the column generation algorithm without knowing any columns of matrix Y and iteratively add new columns to the linear program by solving the pricing problem LP 3 for each passenger $p \in \mathcal{P}$.

$$\begin{aligned}
\min_{X^p} \quad & (c - w^*)^T X^p - \alpha_p^* && (3a)\\
\text{s.t.} \quad & X^p \le u && (3b)\\
& BX^p = d^p && (3c)\\
& X^p \ge 0 && (3d)
\end{aligned} \qquad \text{(LP 3)}$$

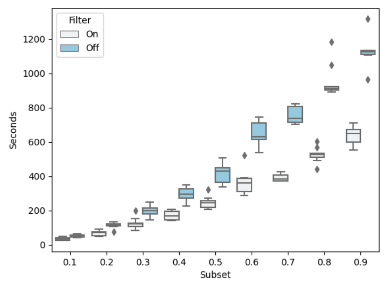
Here, w^* is the vector of dual variables for the capacity constraints (2b), and α^* is the vector of dual variables for the convexity constraint (2c). The matrix $B \in \mathcal{R}^{|\mathcal{V}|\times|\mathcal{A}|}$ is the vertex-edge incidence matrix of graph G. This pricing problem can be solved as a shortest path problem if we neglect constraint (3b), which is already enforced in constraint (2b) of the restricted master problem. We solve these shortest path problems via the A^* algorithm.

To reduce the number of pricing problems called in each iteration, we add a filtering mechanism to the algorithm, which decides whether the pricing problem of passenger $p \in \mathcal{P}$ should be solved. Here, we find all trips, which use or could use saturated arcs and resolve their pricing problems. If no new paths are found, we recalculate all pricing problems to ensure optimality.
To ensure integral solutions in LP 2, we propose a branch-and-price (B&P) approach. Here, our B&P algorithm solves LP 2 at each vertex of the branch-and-bound (B&B) tree. In the vast majority of tested instances, integral solutions where found in the root node of the B&B tree

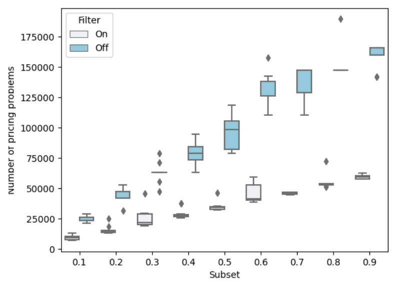
or solving the last updated RMP of our algorithm with λ fixed as an integral variable returned solutions with an optimality gap of under 0.05%. In the case of no integral solution in the root node, we use the just described approach to find an upper bound and perform a depth first search approach on our B&B tree.

3 Preliminary results

We test our algorithm on a case study for the city of Munich, where the network graph consists of 1008 stops and 362 routes, which results in a network graph $G = (\mathcal{V}, \mathcal{A})$ with $|\mathcal{V}| = 145.119$ and $|\mathcal{A}| = 510.224$. The total trip demand set consists of 31.371 travel requests. The passenger data is generated by the travel demand modelling tool MITO (Moeckel *et al.* , 2019) through a Monte-Carlo sampling followed by a nested mode choice model. Figure 2(a) shows the computation time for subsets of all travel requests ranging from 10-90% with and without the filter. All subset sizes have been run on 10 different instances. Our algorithm is able to solve multi commodity flow problems with 2.370 trips in 23 seconds and with 21.326 trips in 10 minutes. Additionally, one can see the stable computation time on the different subsets, which shows the generalizability of our algorithm. Figure 2(b) analogously shows the effectiveness of our filter with respect to the number of solved pricing problems. Solving these instances with a simple LP (see LP 1) without a column generation approach is not possible, since even for the smallest instance we need to define nearly 500 million constraints, such that a standard desktop computer with 16GB of RAM runs out of memory, even for initializing the model. Our code was written in Python 3.9 using the callable library of Gurobi 9.1 to solve the RMP.



(a) *Distribution of computation time*



(b) *Number of pricing problems solved*

Figure 2 – *Computational results*

Acknowledgements: This work was supported by the German Research Foundation (DFG) under grant no. 458548755.

References

Boland, N., Hewitt, M., Marshall, L., & Savelsbergh, M. 2017. The continuous-time service network design problem. *Operations Research*, **65**(5), 1303–1321.

Moeckel, R., Kuehnel, N., Llorca, C., Moreno, A. T., & Rayaprolu, H. 2019. Microscopic Travel demand modeling: using the agility of agent-based modeling without the complexity of activity-based models. *In: Annual Meeting of the Transportation Research Board.*

Rossi, F., Zhang, R., Hindy, Y., & Pavone, M. 2018. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. *Autonomous Robots*, **42**(7), 1427–1442.

Salazar, M., Lanzetti, N., Rossi, N., Schiffer, M., & Pavone, M. 2020. Intermodal Autonomous Mobility-on-Demand. *IEEE Transactions on Intelligent Transportation Systems*, **21**(9), 3946–3960.

UN, DESA. 2018. Revision of world urbanization prospects. *New York: United Nations Department of Economic and Social Affairs.*