Proactive Vehicle Dispatching in Large-Scale Ride-Sourcing Systems

Amirmahdi Tafreshian[1], Mojtaba Abdolmaleki[1], Neda Masoud[1], Huizhu Wang[2]

[1]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI

[2]Ford Motor Company, Greater Detroit Area, MI

Keywords: Proactive optimization, Vehicle dispatching, On-demand dial-a-ride problem, Large-scale optimization, Offline and online optimization framework

---

1 Introduction

The problem of dispatching a fleet of vehicles to serve trip requests can be mathematically formulated as a dial-a-ride problem. Although well-studied, the dial-a-ride problem has been traditionally used for dispatching vehicles in para-transit systems, where trip requests are known well ahead of time. With on-demand transportation alternatives gaining more popularity, the real-time application of dial-a-ride problem is attracting more interest. However, the computational complexity of this problem and the fact that the size of the dial-a-ride problem grows exponentially with the number of requests renders the current solution methodologies inadequate for online applications.

In this research, we study a dynamic dial-a-ride problem for an on-demand ride-sourcing system that provides ride-share services to a set of passengers during time horizon $\mathcal{T} = \{1, ..., T\}$ on a (directed) road network. We assume that the shortest-path travel times and distances between stations are stored in matrices τ and ρ. Passengers dynamically enter the system during the study time horizon and provide information regrading their origin and destination stations, and earliest departure time from their trip origin. A constant parameter, denoted by ω, is introduced to account for the maximum pick-up waiting time of every passenger at their origin station. Furthermore, in order to maintain a certain level of service, another constant parameter, Ω, is introduced to limit the detour of every passenger from their shortest path. Passengers announce their trips at short notice prior to the start of their trip and expect a response immediately. From historical data and/or survey responses, the expected number of requests for every origin-destination pair that enter the system at each time interval is available in $\phi = [(e, o, d)]$.

We have a fleet of K homogeneous vehicles with capacity of C that operate during the time horizon. Every vehicle starts/ends empty in the beginning/end of the horizon from/at any station. As a result of having discrete sets of time and space, we can represent the time-expanded network in our study with a directed acyclic graph $\mathcal{G}$. The route of every vehicle in the study time horizon can be represented by a path in this graph that connects a node with time 0 to a node with time T. Finally we assume that vehicles can wait at any station for any number of time steps if necessary. Finally, without loss of generality we assume that the ride-sourcing system aims to maximize the total shortest-path driving distances of customers. Since fare of customers are usually proportionate to the distance of their trips, this objective function is aligned with the purpose of ride-sourcing companies in maximizing profit.

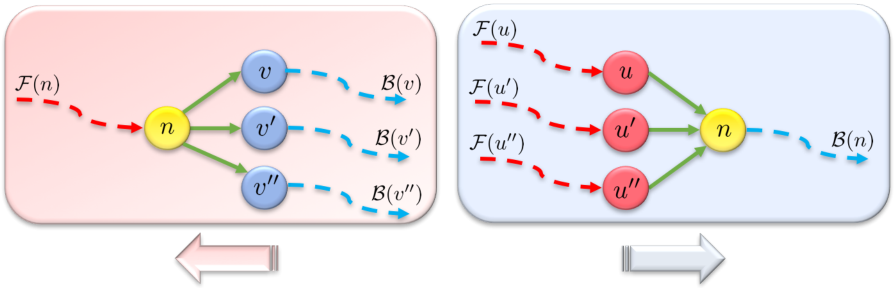
This research introduces a general framework that provides near-optimal insertion methodologies using system-level information, in real time. As such, our framework shifts much of the computational burden of the optimization problems that need to be solved into an offline setting, thereby addressing the on-demand requests with fast and high quality solutions. Our research contributes to the literature in the following ways: *i*) We propose a new local search algorithm for generating a pool of useful routes in the offline mode; *ii*) we introduce an efficient min-cost flow problem to find the best subset of customers served by a fixed route; and *iii*) we propose an online framework that exploits the information from historical data to deploy vehicles proactively and change their routes if necessary in real time.

2 Solution Methodology

Our methodology can be clearly divided into two distinct, yet related, parts, which are implemented in offline and online phases. The offline portion of the solution methodology aims to sequentially generate a pool of routes for vehicles $k = 1, ..., K$. Thus, for each vehicle k, we initially generate a base route $\mathcal{P}(k, 1)$ based on average demand. In order to find the base routes asynchronously, we introduce a clustering/ re-clustering step to distribute/redistribute the average demand among vehicles. Next, we reduce time expanded network around each base route and generate $M - 1$ alternative routes under different realizations of demand requests. In the online setting, the vehicles will be proactively routed through the base routes. In case of having accurate demand forecasts, we expect that the base routes readily serve a good portion of passengers en-route. However, the alternative routes in our pool can be used at any time to change the route of a vehicle toward more promising directions in real-time. At every time step θ, we run a cost-benefit analysis among feasible alternative routes by computing profit based on candidate realized customers and predicted future customers discounted by a factor of β.

2.1 Route Generation.

For generating a route given a demand table, we introduce a local search that starts with generating random partial routes to every node in $\mathcal{G}$. Next, we iterate between a backward and forward local search until convergence. Let $\mathcal{F}(n)$ and $\mathcal{B}(n)$ be partial routes until and from node n, respectively. At every node $n \in \mathcal{G}$ in the backward/forward local search, we decide about its immediate successor/predecessor node (see Figure 1). We proved that this algorithm converges to a local optimal in finite number of iterations.



(a) Backward **(b)** Forward

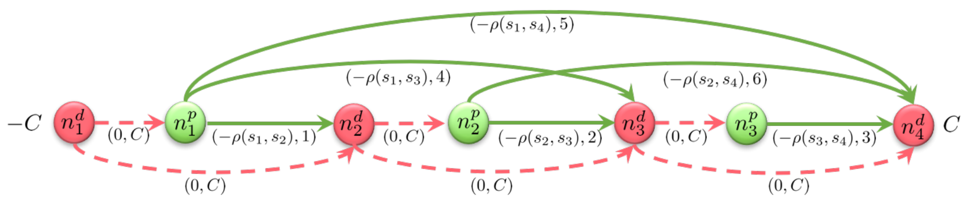
Figure 1: Local Search Algorithm

2.2 Optimal Served Trip

At every single step of backward/forward local search, we have a complete route $\mathcal{X}$. Thus, given a demand table (see Figure 2(a)), we can find the optimal set of trips that can be served by a vehicle on route $\mathcal{X}$ efficiently using a min-cost flow problem (see Figure 2(b)).

	n_2	n_3	n_4
n_1	1	4	5
n_2		2	6
n_3			3

(a) Demand



(b) Min-cost Flow problem

Figure 2: Find optimal served trip by route $\mathcal{X} = \{n_1, n_2, n_3, n_4\}$

3 Numerical Experiments

To showcase the performance of our framework, we consider a simulated ridesharing data set over a well-known Nguyen-Dupuis network, which consists of 13 nodes, 38 links, and 13 OD pairs. A fleet of $K = 50$ homogeneous vehicles with capacity $C = 10$ is available. We assume a study time horizon of $T = 60$ minutes, with time steps of 1 minute. We also set the waiting time ω and detour budget time Ω to 5 and 10 minutes, respectively. We compare the performance of our algorithm with two myopic benchmark methods proposed by [2] and [1], respectively referred to as Insertion and Assignment.

In these experiments, we define two extra parameters μ and σ to generate scenarios where the actual demand is a shifted (by μ) or scaled (by σ) version of values in ϕ. Table 1 summarizes the result of online experiments. For each method, we report the average Revenue ($\overline{\mathcal{Z}}$) and matching rate ($\overline{\mathcal{M}}$) over 20 simulation runs. This table shows that our proposed method significantly overcomes both the benchmark methods. Even in cases where the the demand forecast is far from the actual demand, our proposed method outperform these myopic methods. In this table, the best value of β is embolden for each scenario. The right choice of value for β depends on the performance of our demand forecast. In general, high values of β provide higher revenues when the average of expected demand is close to the average of realized one (scenarios 1 and 3). However, as mean of demand forecast gets farther from the mean of realized demand, we must choose lower values of β.

Table 1: The results of different methods on the performance measures averaged over simulation runs in the Nguyen-Dupuis case study

$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$	$\boldsymbol{\beta}$	**Our Method**		**Insertion**		**Assignment**	
			$\overline{\mathcal{Z}}_1$ (miles)	$\overline{\mathcal{M}}_1$ (%)	$\overline{\mathcal{Z}}_2$ (miles)	$\overline{\mathcal{M}}_2$ (%)	$\overline{\mathcal{Z}}_3$ (miles)	$\overline{\mathcal{M}}_3$ (%)
0	0	0.95	2748.90	74	2003.29	57	1914.25	54
		0.50	2741.19	74				
		0.25	2725.04	74				
5	0	0.95	3729.16	56	2997.80	47	2835.92	44
		0.50	3713.06	56				
		0.25	3677.31	57				
0	5	0.95	2847.92	67	2242.68	55	2137.66	52
		0.50	2842.04	67				
		0.25	2823.32	67				

4 Conclusion

It can be easily shown that our methodology can be simply adapted to special cases where there exist a set of heterogeneous origins and destinations as well as capacities for vehicles. Also, note that the proposed methodology is in independent of choice of objective function.

Acknowledgment

This work was supported by a research grant from the Ford Motor Company.

References

[1] Alonso-Mora, J., S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences 114*(3), 462–467.

[2] Ma, S., Y. Zheng, and O. Wolfson (2013). T-share: A large-scale dynamic taxi ridesharing service. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 410–421. IEEE.