Geometric Optimization for Routing with Floating Targets

Wei Zhang
Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hong Kong, wei.sz.zhang@connect.polyu.hk

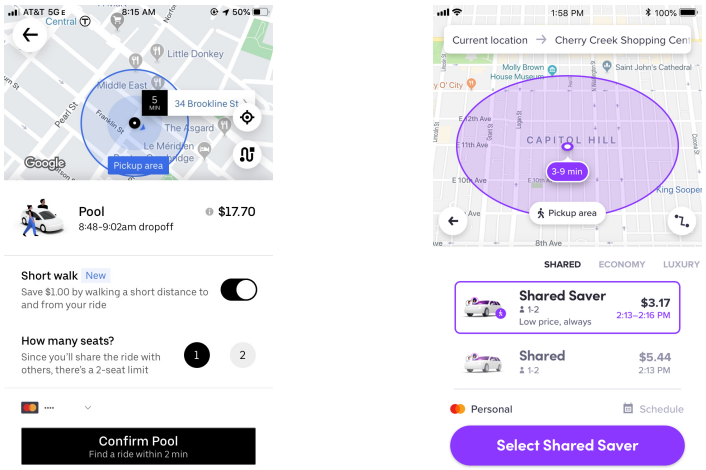
Alexandre Jacquillat
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, alexjacq@mit.edu

Kai Wang
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, chwangkai@gmail.com

Shuaian Wang
Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hong Kong, hans.wang@polyu.edu.hk

---

Recent industrial developments have led to a new feature in vehicle routing: floating targets. Instead of meeting at prespecified origins and destinations, customers and vehicles can now meet at other locations where pickups are more convenient and more efficient. Such operations are commonly found in ride-sharing (e.g., Uber Express Pool, Lyft Shared Saver, shown in Figure 1). They are also expected to lie at the core of future transportation systems (e.g., on-demand buses, urban aerial mobility, drone-based logistics). By leveraging information and communication technologies, operators can use floating targets to alleviate the costs of first-mile and last-mile transportation.



(a) Uber (b) Lyft

Figure 1 User interface of Uber Express Pool and Lyft Shared Saver (circles represent pickup areas).

At the core of these operations lies a conventional routing problem—how to visit a set of customers as efficiently as possible. But floating targets add one extra degree of freedom: instead of visiting customers in pre-determined stops, vehicles can optimize the stopping locations themselves—thus providing more flexibility in operations. This flexibility, however, comes at the cost of increasing computational complexity, thus motivating new modeling and algorithmic approaches for routing optimization with floating targets.

This paper addresses these questions by revisiting canonical routing problems to optimize routing operations in the presence of floating targets. We tackle the following two problems:

– *The Vehicle Routing Problem with Floating Targets (VRP–FT).* It takes as inputs a set of homogeneous vehicles with limited capacity and a set of customers to serve. It jointly optimizes the assignment of vehicles to customers, the sequence of customer visits for each vehicle, the time of each customer visit, and the location of each customer visit—in order to minimize travel times. Sample applications include company bus routing and aerial refueling.

– *The Dial-A-Ride problem with Floating Targets (DAR–FT).* It takes as inputs a set of customer requests, each with an origin, a destination, and a time window. It jointly optimizes the subset of customers to serve, the sequence of customer pickups and dropoffs, the time of each pickup and dropoff, and the location of each pickup—in order to maximize operating profits. Sample applications include ride-sharing and on-demand high-capacity vehicle routing.

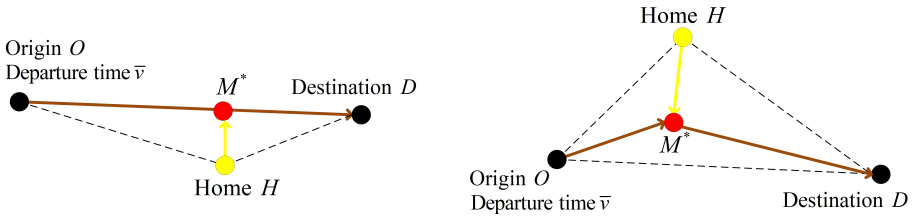
In these problems, floating targets introduce nonlinearities—stemming from Euclidean distances in the two-dimensional space. Gambella, Naoum-Sawaya, and Ghaddar (2018) tackled VRP–FT with mixed-integer second-order cone programming (MISOCP). To our knowledge, DAR–FT has not been defined yet. This paper proposes a novel approach to solve routing problems with floating targets, based on geometric optimization, and embeds it into dynamic programming algorithms to solve VRP–FT and DAR–FT. Theoretical and computational results show that this approach can solve large-scale instances optimally—much faster than state-of-the-art MISOCP benchmarks.

Specifically, this paper makes the following four contributions.

- We develop a new approach based on geometric optimization to determine the location and time of one pickup. This is referred to as the *Single Pickup Optimization with a Floating Target* (SPO–FT). Based on a geometric representation of the vehicle's and the customer's operations in the Euclidean space, we derive structural insights on the optimal pickup location—given the vehicle's and the customer's speeds and the customer's maximum walking distance.

  Figure 2 illustrates a critical distinction at the core of our geometric analysis. The optimal solution of SPO–FT can fall in two categories: a *straight path* solution, in which the customer walks to an "ideal" pickup location M^* for the vehicle, and a *detour* solution, in which the vehicle's travel time increases to pick up the customer. In our geometric optimization procedure,

we first determine whether a straight path exists (in which case it is optimal). Otherwise, we derive the optimal pickup location M^* within the triangle formed by the vehicle's origin O, the customer's home H, and the vehicle's destination D. Our main result is that the optimal location can be solved exactly by finding the optimal angle $\widehat{HOM}^*$, that is, the angle between the vehicle's directions without and with floating targets. These insights reduce the SPO–FT from a three-dimensional optimization problem (one temporal dimension and two spatial dimensions) to simple one-dimensional problems (one angular dimension)—which can be solved computationally efficiently in polynomial time.



(a) Straight path (b) Detour

Figure 2 Geometric representation of the straight path and detour solutions.

- Using our geometric optimization approach, we develop an exact decomposition algorithm to optimize when and where to visit a set of customers in a fixed sequence. This is referred to as the *Multiple Pickup Optimization with Floating Targets* (MPO–FT). Our algorithm decomposes the MPO–FT by optimizing the time and location of *each pickup iteratively* one at a time, using SPO–FT. Our main theoretical result is that this algorithm improves the solution at each iteration, and converges to the MPO–FT optimum. We also propose a warm-start acceleration using a forward decomposition heuristic. Computational results suggest that our solution approach can solve MPO–FT 50–100 times faster than SOCP solvers.
- We design exact and approximate dynamic programming algorithms to solve the Vehicle Routing Problem with Floating Targets (VRP–FT). These algorithms optimize the sequence of customers visited by each vehicle—evaluating each sequence with MPO–FT. Ultimately, they determine which customers each vehicle will serve and in which sequence. We compare our solution approach to the branch-and-price (B&P) algorithm from Gambella, Naoum-Sawaya, and Ghaddar (2018). Results show that our algorithm provides Pareto improvements, yielding higher-quality solutions in shorter computational times. Sample results are reported in Table 1. The B&P solves instances with up to 16 customers to optimality; in these instances, our algorithm also returns the optimal solution, but reduces computational times by at a significant margin. Among the seven larger-scale instances that B&P does not solve to optimality, our algorithm returns a much better solution in four instances, and always terminates much faster.

- We formalize the Dial-A-Ride problem with Floating Targets (DAR–FT), and design exact and approximate dynamic programming algorithms to solve it. These algorithms optimize the sequence of pickups between consecutive dropoffs—evaluating each sequence with MPO–FT. Ultimately, they determine which customers to serve and in which sequence. We compare our solution approach to direct implementations of DAR–FT as a MISOCP using CPLEX. Again, we find that our algorithm provides Pareto improvements, yielding higher-quality solutions in shorter computational times. Sample results are reported in Table 2. Direct MISOCP implementation of the DAR–FT does not scale to even medium-sized instances. Indeed, CPLEX only solves instances with up to 8 customers to optimality with 2.5 hours. In contrast, our algorithm solves instances with up to 25 customers to optimality in seconds. Moreover, our algorithm can tackle large-scale instances with up to 200 customers.

	Gambella et al. (2018)		Our algorithm			
Cust.	Sol.	CPU(s)	Sol.	CPU(s)	CPU(%)	Sol. (%)
10	62.14	70	62.14	9	−87%	0%
10	71.71	161	71.71	1	−99%	0%
12	91.36	1,209	91.36	35	−97%	0%
12	80.08	532	80.08	48	−91%	0%
14	97.62	5,131	97.62	1,817	−65%	0%
14	80.18	377	80.18	304	−19%	0%
16	181.58	>7,200	87.36	1,436	-	−52%
16	75.85	6,635	75.85	5,165	−22%	0%
18	217.47	>7,200	115.15	5,500	-	−47%
18	80.68	>7,200	80.54	4,321	-	0%
20	193.88	>7,200	88.83	6,492	-	−54%
20	216.63	>7,200	110.08	1,345	-	−49%
20	128.35	>7,200	127.80	4,041	-	0%

Table 1: Sample resuts of VRP–FT.

	MISOCP		Our algorithm	
Cust.	Sol.	CPU (s)	Sol.	CPU (s)
5	10.2	13	10.2	<1
7	6.5	3,382	6.5	<1
8	16.1	1,692	16.1	<1
10	4.8	>9,000	4.8	<1
15	5.1	>9,000	25.2	1
20	4.7	>9,000	14.5	<1
25	8.1	>9,000	16.9	2
50	-	-	21.5	279
75	-	-	21.7	212
100	-	-	22.6	1,419
150	-	-	20.2	1,330
175	-	-	23.1	5,054
200	-	-	23.9	2,245

Table 2: Sample resuts of DAR–FT.

From a practical standpoint, our results reveal that floating targets, by themselves, do not necessarily reduce costs or increase profits. This is rather surprising because floating targets make operations more flexible. But they also complicate the underlying matching and routing problems. As it turns out, the net effects can be negative: solving large-scale problems with floating targets using off-the-shelf optimization methods can lead to a deteriorated solution. Instead, practical benefits arise from floating targets coupled with tailored optimization methods such as the one developed in this paper—thus highlighting the joint role of business and technical developments.

References

Gambella C, Naoum-Sawaya J, Ghaddar B, 2018 *The vehicle routing problem with floating targets: Formulation and solution approaches. INFORMS Journal on Computing* 30(3):554–569.