

Routing Optimization with Vehicle-Customer Coordination

In several transportation systems, vehicles can choose where to meet customers rather than stopping in fixed locations. This added flexibility, however, requires coordination between vehicles and customers that adds complexity to routing operations. This paper develops scalable algorithms to optimize these operations. First, we solve the one-stop subproblem in the ℓ_1 space and the ℓ_2 space, by leveraging the geometric structure of operations. Second, to solve a multi-stop problem, we embed the single-stop optimization into a tailored coordinate descent scheme, which, we prove, converges to a global optimum. Third, we develop a new algorithm for dial-a-ride problems, based on a subpath-based time-space network optimization combining set partitioning and time-space principles. Finally, we propose an online routing algorithm to support real-world ride-sharing operations with vehicle-customer coordination. Computational results show that our algorithm outperforms state-of-the-art benchmarks, yielding far superior solutions in shorter computational times, and can support real-time operations in very large-scale systems. From a practical standpoint, most of the benefits of vehicle-customer coordination stem from comprehensively re-optimizing “upstream” operations, as opposed to merely adjusting “downstream” stopping locations. Ultimately, vehicle-customer coordination provides win-win-win outcomes: higher profits, better customer service, and smaller environmental footprint.

Key words: Vehicle-customer coordination; Vehicle routing; Ride-sharing; Time-space network.

1. Introduction

Several transportation systems leverage connected technologies and digitization to coordinate vehicles’ and customers’ operations. In the most prominent example, ride-sharing providers have launched services where riders can walk to meet drivers in mutually convenient locations in exchange of a discount (see Figure 1). Similarly, company and school buses pick up riders in a few central locations prior to traveling to a common destination. Another example is aerial refueling, in which a tanker aircraft coordinates its operations with other aircraft to transfer fuel.

Vehicle-customer coordination provides an extra degree of freedom to enhance first- and last-mile transportation. This flexibility, however, comes with challenges. At the *downstream* level, the immediate question is how to optimize the timing and location of each stop. At the *upstream* level, service providers need to comprehensively re-optimize routing operations to take full advantage of vehicle-customer coordination—which customers to serve, with which vehicles, and in which sequence. In turn, vehicle-customer coordination requires new dedicated routing algorithms.

This problem falls into the broad umbrella of vehicle routing with capacitated vehicles and time windows—a challenging class of combinatorial optimization problems that lies at the core of modern transportation and logistics systems (see, e.g., Holland et al. 2017). Many variants have emerged to capture features of modern supply chains (e.g., consistency requirements, electric vehicles, multi-tier

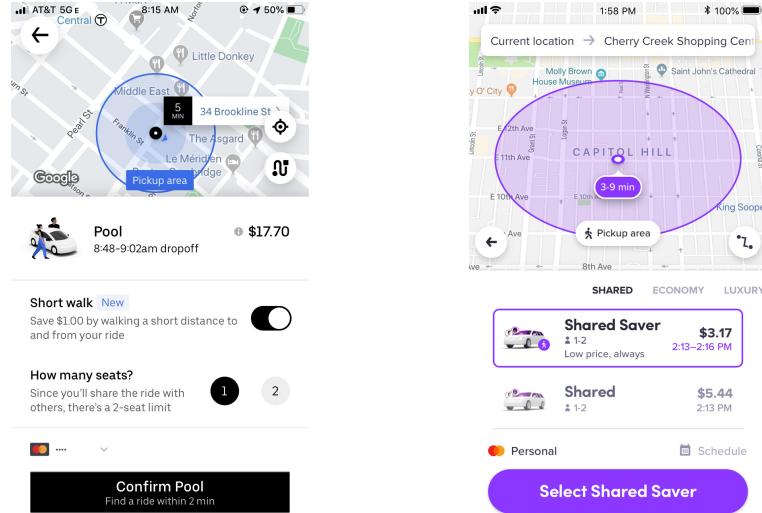


Figure 1 Interface of Uber Express Pool and Lyft Shared Saver (shaded areas represent possible pickup locations).

operations) but each comes with new computational challenges. The only attempt to date to tackle a routing problem with vehicle-customer coordination comes from Gambella et al. (2018), who solved a mixed-integer second-order cone optimization model in the Euclidean space via branch-and price. Yet, their algorithm falls short of the large-scale instances arising in practical applications.

In response, the first goal of this paper is to develop scalable algorithms to support dial-a-ride routing with capacitated vehicles, time windows, and with vehicle-customer coordination. This problem involves mixed-integer second-order cone optimization (MISOCO) in the Euclidean ℓ_2 space and mixed-integer linear optimization (MILO) in the Manhattan ℓ_1 space. Either way, off-the-shelf implementation only scales to small instances. Instead, we develop a decomposition algorithm that breaks down the problem into three steps, resulting in three methodological contributions:

- *Single Stop Optimization with Vehicle-Customer Coordination* (SSO–VCC, Section 3). We derive geometric insights on the optimal stopping location (between two fixed locations) as a function of the vehicle’s speed, the customer’s speed and the customer’s maximum walking distance. This reduces SSO–VCC from a three-dimensional problem (one temporal dimension and two spatial dimensions) to one-dimensional problems in the ℓ_2 space, or a closed-form system of linear equations in the ℓ_1 space. Either way, SSO–VCC can be solved very efficiently.
- *Multiple Stop Optimization with Vehicle-Customer Coordination* (MSO–VCC, Section 4). This problem involves optimizing the locations and times of multiple stops in a given sequence. We propose a tailored coordinate descent scheme that optimizes one stop at a time, thus decomposing MSO–VCC into a sequence of SSO–VCC problems. Our main result establishes that, in the ℓ_1 space and the ℓ_2 space, this algorithm terminates at a globally optimal solution of the MSO–VCC—a constrained and non-separable optimization problem.

– *Dial-A-Ride with Vehicle-Customer Coordination* (DAR–VCC, Section 5). We propose an original subpath-based time-space approach in which “empty vehicles” flow from node to node and arcs represent customer-serving trips (or subpaths). We generate candidate trips by embedding our MSO–VCC algorithm into a label-setting dynamic programming algorithm, and select trips via time-space integer optimization. This algorithm contributes to the dial-a-ride literature by combining set partitioning principles and time-space principles—namely, by using subpath-based variables as opposed to route-based variables or arc-based variables. Results show that this algorithm can handle instances with up to 50 customers in seconds and instances with up to 200 customers in minutes, significantly outperforming state-of-the-art MISOCO and MILO benchmarks—both solution quality and computational times.¹

Our second goal is to support real-world operations with vehicle-customer coordination, with a focus on “walking products” in ride-sharing (Figure 1). We propose an *Online Dial-A-Ride model with Vehicle-Customer Coordination* (O–DAR–VCC, Section 6) that captures dynamic customer arrivals as well as vehicles’ and customers’ operations in complex road networks featuring one-way streets, traffic congestion, and a discrete set of possible stopping locations (as opposed to continuous operations in the ℓ_1 space or the ℓ_2 space). Our online algorithm proceeds via batching and optimization, by dynamically optimizing, every 10 seconds, service to “new” customers, service to “backlogged” customers, and vehicle repositioning. Using real-world data, we show that our algorithm, combined with a history-based heuristic acceleration, consistently terminates in seconds for the full Manhattan taxi system with thousands of requests per hour—enabling its real-time implementation in very large-scale networks of operations.

From a practical standpoint, our results suggest that vehicle-customer coordination can provide significant improvements in routing operations, with an average profit increase of 6%. For a system of the size of Manhattan, this represents an estimated gain of \$100,000 daily, or \$35–40M annually. We also show that vehicle-customer coordination provides limited benefits at the “downstream” level by merely adjusting stopping locations; rather, most of the gains stem from comprehensively re-optimizing “upstream” operations in view of the added flexibility: which customers to serve, in which sequence and with which vehicles. By sharing the benefits equally between the operator and customers, we obtain a solution such that (i) the operator’s profit increases by 3%, (ii) 6% extra customers receive a service, (iii) 22% of customers walk before pickup and 7% walk after dropoff, with an average distance of 100 meters and a fare discount of 8%, and (iv) vehicle miles traveled are reduced by 15%. Ultimately, vehicle-customer coordination can lead to win-win-win outcomes: higher profits, better customer service, and smaller environmental footprint.

¹ We also solve the *Vehicle Routing Problem with Vehicle-Customer Coordination* (VRP–VCC) from Gambella et al. (2018) by embedding MSO–VCC into a dynamic programming algorithm. Our approach provides Pareto improvements as compared to the branch-and-price benchmark: higher-quality solutions in shorter runtimes (Appendix A).

2. Literature Review

Dial-a-ride. Dial-a-ride and pickup-and-delivery problems involve vehicle routing operations in which demand materializes from an origin to a destination (Psaraftis 1980, Savelsbergh and Sol 1995). Our dial-a-ride problem in this paper features capacitated vehicles and time windows. Exact solution approaches for this class of problems span three categories: branch-and-cut (Cordeau 2006), set partitioning methods (Desaulniers 2010, Dabia et al. 2013) and time-space methods (Chardaire et al. 2005, Baldacci et al. 2012, Dash et al. 2012). In this paper, we propose a new approach based on subpath-based variables, each corresponding to a sequence of pickups and dropoffs from an “empty point” (where a vehicle carries no passenger) to the next. This approach tightens direct branch-and-cut formulations by integrating vehicle capacities and time windows into the definition of subpath-based variables. As compared to arc-based time-space formulations, it induces sparsity by defining nodes as “empty points” and arcs as customer-carrying trips—bringing set partitioning principles into time-space methods. And as compared to set partitioning formulations, it also induces sparsity by using subpath-based variables (from empty point to empty point) as opposed to route-based variables (from origin to destination) and by connecting them via a time-space network optimization model—bringing time-space principles into set partitioning methods.

Ride-sharing. Extensive research in ride-sharing has focused on vehicle-customer matching (Agatz et al. 2011, Özkan and Ward 2020), spatial-temporal pricing (Bimpikis et al. 2019, Besbes et al. 2021, Hu et al. 2022), vehicle repositioning (Braverman et al. 2019, Balseiro et al. 2021), etc. Within this literature, our paper relates to graph-based routing optimization. Santi et al. (2014) use a maximum matching model in a ride-shareability network where arcs indicate rides that can be pooled. Vazifeh et al. (2018) use a minimum path cover model in a vehicle-shareability network, where arcs indicate rides that can be served by the same vehicle. Bertsimas et al. (2019) optimize taxi routing (without ride-pooling) in a network where arcs indicate consecutive trips. Our paper contributes to this literature with a new network representation, where nodes characterize “empty points” for vehicles, arcs characterize vehicle subpaths from one “empty point” to the next (each serving one or multiple customers), and vehicle-customer assignments are determined via time-space network optimization with flow balance, packing and budget constraints.

This approach connects to Alonso-Mora et al. (2017), who first generate a set of feasible trips for the vehicles, and then optimize the assignment of customer requests, vehicle trips and vehicles. Our approach follows a similar two-step procedure. However, we restrict trips to a sequence of pickups and dropoffs from an “empty point” to the next, as opposed to full vehicle routes. Our network optimization model then connects vehicle trips together to reconstruct full routes. As such, our approach relieves the burden of the trip-generation part, while retaining tractability in the subsequent vehicle-customer assignment optimization part (using a time-space network). Finally, our paper is the first to embed vehicle-customer coordination into on-demand ride-sharing.

Routing with flexible stops. Our flexible routing problem relates to body chasing, which seeks a distance-minimizing sequence of points, each in a given set. Friedman and Linial (1993) prove that the collection of convex sets in \mathbb{R}^2 is *chaseable*, i.e., the online-to-offline cost ratio is bounded. Bubeck et al. (2019) extend this result in higher dimensions. In this paper, we seek an exact algorithm to solve large-scale instances, as opposed to theoretical guarantees for online policies. Moreover, in our setting, feasible stopping regions depend on prior operations.

This links our problem to vehicle routing with non-stationary targets. Stieber et al. (2015) and Xia et al. (2019) optimize the VRP with customers moving along fixed trajectories. Stieber et al. (2015) and Ozbaygin et al. (2017) optimize the VRP with “roaming targets”, in which customers are located in different places at different times of the day. In our setting, customer operations are endogenous, in that the service provider coordinates vehicle and customer operations to design mutually convenient stopping locations. Another related problem is the horsefly routing problem to coordinate a primary vehicle (e.g., truck, ship) and a drone (Carlsson and Song 2017, Poikonen and Golden 2020). Our setting features moving customers as opposed to a moving depot.

Gambella et al. (2018) are the first to investigate the VRP with vehicle-customer coordination (or “floating targets”). They formulate a mixed-integer second-order cone optimization model in the Euclidean space, and solve it with a branch-and-price algorithm. Our paper builds upon their work, but exhibits several differences. This paper designs a new decomposition algorithm, which, as our results show, provides higher-quality solutions in shorter runtimes. Moreover, we consider a dial-a-ride problem with practical restrictions, such as time windows and discretized road networks, and design an online algorithm to support real-time ride-sharing with vehicle-customer coordination.

3. Single Stop Optimization with Vehicle-Customer Coordination

We consider operations in the two-dimensional Euclidean space, induced by the ℓ_2 norm $\|\cdot\|_2$, and the two-dimensional Manhattan space, induced by the ℓ_1 norm $\|\cdot\|_1$. Consider a vehicle traveling from an origin $O \in \mathbb{R}^2$ to a destination $D \in \mathbb{R}^2$, starting at time $\bar{v} \geq 0$. The vehicle needs to visit one customer, who starts in a home location $H \in \mathbb{R}^2$ at time 0. We denote by M the stopping location and by v_M the stopping time. The customer can walk up to a distance W from H to M . Let S and \bar{S} be the customer’s and vehicle’s speeds, with $\bar{S} > S$. SSO-VCC minimizes the time at which the vehicle reaches its destination, while ensuring that both the vehicle and the customer arrive at the stopping location by time v_M and that the customer does not walk more than W .

3.1. Preliminary results

Lemma 1 shows that, at the optimum, the vehicle does not wait for the customer at the stopping location (because it travels faster). All proofs from this section are reported in EC.1.

LEMMA 1. *The optimal solution of SSO–VCC satisfies $v_M = \bar{v} + \frac{\|M-O\|}{S}$.*

Based on Lemma 1, SSO–VCC can be formulated as follows:

$$v^* = \min_{M \in \mathbb{R}^2} \left\{ \bar{v} + \frac{\|M-O\| + \|D-M\|}{S} : \text{s.t. } \frac{\|M-H\|}{S} \leq \bar{v} + \frac{\|M-O\|}{S}, \|M-H\| \leq W \right\}. \quad (1)$$

Next, Lemma 2 introduces a key distinction between a *straight-path solution* (in which the customer walks to an “ideal” stopping location for the vehicle) versus a *detour solution*.

LEMMA 2. *The optimal solution v^* satisfies $v^* \geq \bar{v} + \frac{\|D-O\|}{S}$. Moreover, $v^* = \bar{v} + \frac{\|D-O\|}{S}$ if and only if there exists a feasible stopping location M on the straight line segment between O and D in the ℓ_2 space or in the rectangle formed by O and D in the ℓ_1 space.*

Finally, Lemma 3 shows that, in a detour solution, at least one of the two constraints in Equation (1) is binding: either the vehicle and the customer reach the stopping location M^* at the same time, or the customer travels until reaching the maximum walking distance.

LEMMA 3. *In a detour, the optimum M^* satisfies $\frac{\|M^*-H\|}{S} = \bar{v} + \frac{\|M^*-O\|}{S}$ or $\|M^*-H\| = W$.*

We now determine the optimal solution in the Euclidean space (Section 3.2) and the Manhattan space (Section 3.3). In each case, we first examine whether a straight path exists (in which case it is optimal); otherwise, we derive structural geometric insights to find the optimal stopping location.

3.2. SSO–VCC in the Euclidean space

Synchronous operations. Let us first assume that the vehicle and the customer leave at the same time ($\bar{v} = 0$). Without loss of generality, O is located in $(0, 0)$, H in $(d, 0)$ (with $d = \|H-O\|_2$), and D in (X, Y) . Lemma 4 shows that the feasible region of Equation (1) is the overlap of two non-concentric disks: (i) a *vehicle pickup disk* \mathcal{D}^V of center $(d_V, 0)$ and radius r_V , whose center reflects that vehicle travels faster than the customer ($d_V > d$); and (ii) a *customer walking disk* \mathcal{D}^C , with center H and radius W . We denote the corresponding circles by \mathcal{C}^V and \mathcal{C}^C .

LEMMA 4. *In the Euclidean space, the constraint $\frac{\|M-H\|_2}{S} \leq \frac{\|M-O\|_2}{S}$ is equivalent to $M \in \mathcal{D}^V = \{(\chi, \gamma) \in \mathbb{R}^2 : (\chi - d_V)^2 + \gamma^2 \leq r_V^2\}$, where $d_V = d \times \frac{\bar{S}^2}{\bar{S}^2 - S^2} > d$ and $r_V = d \times \frac{\bar{S}S}{\bar{S}^2 - S^2}$. The constraint $\|M-H\|_2 \leq W$ is equivalent to $M \in \mathcal{D}^C = \{(\chi, \gamma) \in \mathbb{R}^2 : (\chi - d)^2 + \gamma^2 \leq W^2\}$.*

Figure 2 illustrates SSO–VCC operations. If the destination D is in the region \mathcal{R}^S (shown in blue) defined by the two tangent lines to $\mathcal{D}^V \cap \mathcal{D}^C$ starting at the origin $O = (0, 0)$, there exists a straight path solution. Otherwise, the vehicle needs to make a detour. Per Lemma 3, the optimal stopping location lies on an arc of the vehicle pickup circle, denoted by $\mathcal{A}^V = \mathcal{C}^V \cap \mathcal{D}^C$ (Figure 2a), or on an arc of the customer walking circle, denoted by $\mathcal{A}^C = \mathcal{C}^C \cap \mathcal{D}^V$ (Figure 2b).

In polar coordinates, we denote by θ (resp. α) the angle related to arc \mathcal{A}^V (resp. \mathcal{A}^C), shown in Figure 2a (resp. Figure 2b). Proposition 1 elicits the optimal solution, by making use of functions $f(\theta)$ and $h(\alpha)$ and interval restrictions of the form $\theta \in [\theta_L^M, \theta_U^M] \cap [\theta_L^I, \theta_U^I]$ and $\alpha \in [\alpha_L^N, \alpha_U^N] \cap [\alpha_L^I, \alpha_U^I]$.

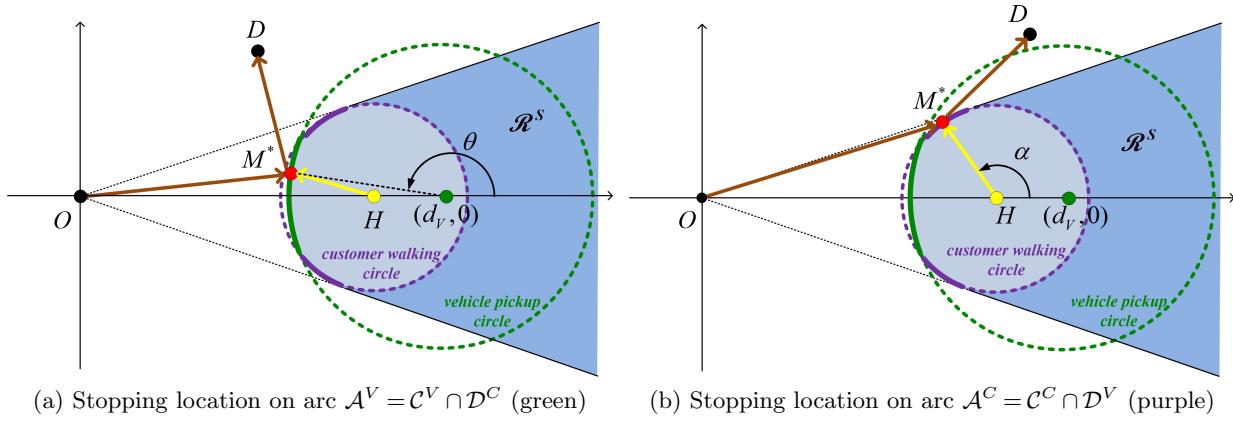


Figure 2 Geometric representation of detour solutions in synchronous SSO–VCC with the Euclidean distance.

PROPOSITION 1. *There exist coordinates (X_P, Y_P) such that the straight-path region \mathcal{R}^S becomes:*

$$\mathcal{R}^S = (\mathcal{D}^V \cap \mathcal{D}^C) \cup \left\{ (X, Y) \in \mathbb{R}^2 : X > X_P \text{ and } \left| \frac{Y}{X} \right| \leq \frac{Y_P}{X_P} \right\}.$$

- If $D \in \mathcal{R}^S$, all stopping locations at the intersection of $[OD]$, \mathcal{D}^V and \mathcal{D}^C are optimal.
- If $D \notin \mathcal{R}^S$, there exist angles $\theta_L^M, \theta_U^M, \theta_L^I, \theta_U^I, \alpha_L^N, \alpha_U^N, \alpha_L^I$, and α_U^I such that:

$$v^* = \frac{1}{S} \min \left\{ \min_{\theta \in [\theta_L^M, \theta_U^M] \cap [\theta_L^I, \theta_U^I]} f(\theta), \min_{\alpha \in [\alpha_L^N, \alpha_U^N] \cap [\alpha_L^I, \alpha_U^I]} h(\alpha) \right\}, \text{ where:}$$

$$f(\theta) = \sqrt{r_V^2 + d_V^2 + 2d_V r_V \cos(\theta)} + \sqrt{r_V^2 + (d_V - X)^2 + Y^2 + 2r_V(d_V - X) \cos(\theta) - 2r_V Y \sin(\theta)},$$

$$h(\alpha) = \sqrt{W^2 + d^2 + 2dW \cos(\alpha)} + \sqrt{W^2 + (d - X)^2 + Y^2 + 2W(d - X) \cos(\alpha) - 2WY \sin(\alpha)}.$$

The functions f and h are differentiable and unimodal. The optimal stopping location is given by $M^* = (d_V + r_V \cos(\theta^*), r_V \sin(\theta^*))$ or $M^* = (d + W \cos(\alpha^*), W \sin(\alpha^*))$.

Proposition 1 reduces SSO–VCC from a three-dimensional problem to two one-dimensional problems. Each one involves minimizing a unimodal function, and can thus be solved efficiently using ternary search. Ultimately, a solution is found in $\mathcal{O}(\log(\frac{1}{\varepsilon}))$, where ε denotes the tolerance.

Asynchronous operations. If the vehicle departs after the customer ($\bar{v} > 0$), Equation (1) can no longer be simplified as in Lemma 4. We divide Equation (1) by splitting the constraint $\frac{\|M-H\|_2}{S} \leq \bar{v} + \frac{\|M-O\|_2}{S}$ based on whether the customer can arrive at the stop prior to the vehicle's departure time \bar{v} (Equation (2)) or not (Equation (3)). Namely, $v^* = \min\{v^a, v^b\}$, where:

$$v^a = \bar{v} + \min_{M \in \mathbb{R}^2} \left\{ \frac{\|M-O\|_2 + \|D-M\|_2}{S} : \frac{\|M-H\|_2}{S} \leq \bar{v}, \|M-H\|_2 \leq W \right\}, \quad (2)$$

$$v^b = \bar{v} + \min_{M \in \mathbb{R}^2} \left\{ \frac{\|M-O\|_2 + \|D-M\|_2}{S} : \bar{v} < \frac{\|M-H\|_2}{S} \leq \bar{v} + \frac{\|M-O\|_2}{S}, \|M-H\|_2 \leq W \right\}. \quad (3)$$

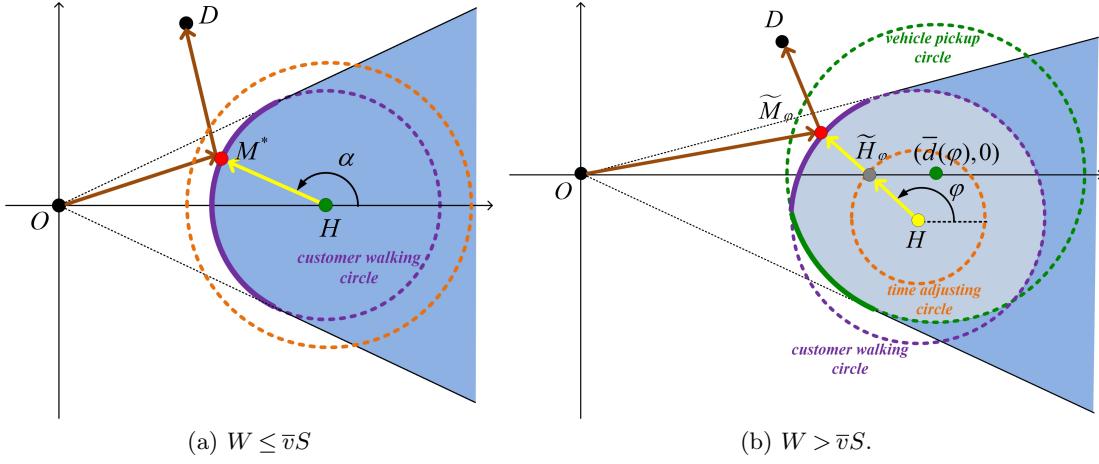


Figure 3 Geometric representation of asynchronous SSO–VCC with the Euclidean distance.

If $W \leq \bar{v}S$, Equation (2) is only constrained by $\|M - H\|_2 \leq W$; Equation (3) is infeasible. The optimum is obtained as in Proposition 1, with only the customer walking disk \mathcal{D}^C (Figure 3a).

If $W > \bar{v}S$, Equation (2) is only constrained by $\|M - H\|_2 \leq \bar{v}S$. Its optimal solution is obtained as before, by replacing the customer walking disk by a *time adjusting disk*, \mathcal{D}^T , with center H and radius $\bar{v}S$. If there exists a feasible straight-path solution within \mathcal{D}^T , it is optimal, i.e., $v^* = v^a = \bar{v} + \frac{\|D - O\|_2}{\bar{S}}$. Otherwise, we need to solve Equation (3) by searching for the optimal angle $\varphi \in [0, 2\pi]$ specifying the customer's walking direction from time 0 to \bar{v} , and then solving a synchronous SSO–VCC. Proposition 2 simplifies the problem, by showing that the customer follows the same walking direction between time 0 and \bar{v} and between \bar{v} and v_M (see Figure 3b).

PROPOSITION 2. *For any $\varphi \in [0, 2\pi]$, let $\tilde{H}_\varphi = (\bar{d} + \bar{v}S \cos(\varphi), \bar{v}S \sin(\varphi))$ be the customer's location at time \bar{v} . Let $\mathcal{D}^\varphi = \left\{ M \in \mathbb{R}^2 : \frac{\|M - \tilde{H}_\varphi\|_2}{\bar{S}} \leq \frac{\|M - O\|_2}{\bar{S}} \right\}$, so that $\mathcal{D}^\varphi \cap \mathcal{D}^C$ is the feasible region. Let \tilde{M}_φ be the intersection of ray $\overrightarrow{H\tilde{H}_\varphi}$ with $\mathcal{D}^\varphi \cap \mathcal{D}^C$. We have:*

$$v^b = \bar{v} + \min_{\varphi \in [\varphi_L, \varphi_U]} \left(\frac{\|\tilde{M}_\varphi - O\|_2 + \|D - \tilde{M}_\varphi\|_2}{\bar{S}} \right).$$

It is thus sufficient to search for $\varphi \in [\varphi_L, \varphi_U]$ alone, rather than jointly searching for $\varphi \in [\varphi_L, \varphi_U]$ and for $M \in \mathbb{R}^2$. We can thus apply a traverse search over $[\varphi_L, \varphi_U]$. For any φ , we derive the intersection between the ray $\overrightarrow{H\tilde{H}_\varphi}$ and the boundary of $\mathcal{D}^\varphi \cap \mathcal{D}^C$, and apply results from synchronous operations (Proposition 1). This algorithm terminates in $\mathcal{O}(\frac{1}{\epsilon})$. Ultimately, we have reduced SSO–VCC (in the Euclidean space) from a three-dimensional problem to simple one-dimensional problems.

3.3. SSO–VCC in the Manhattan space

In the Manhattan space, the feasible region of Equation (1) is a polygon—the intersection of \mathcal{G}^V induced by $\frac{\|M - H\|_1}{\bar{S}} \leq \bar{v} + \frac{\|M - O\|_1}{\bar{S}}$, and \mathcal{G}^C induced by $\|M - H\|_1 \leq W$. If the feasible region $\mathcal{G}^V \cap \mathcal{G}^C$

is overlapping with the rectangle defined by O and D , denoted by $O \square D$, a straight-path solution exists and is thus optimal (Figure 4a). Otherwise, the optimal solution M^* lies on $\mathcal{S}_I^V = \mathcal{B}^V \cap \mathcal{G}^C$ or on $\mathcal{S}_I^C = \mathcal{B}^C \cap \mathcal{G}^V$, where \mathcal{B}^V and \mathcal{B}^C denote the boundaries of \mathcal{G}^V and \mathcal{G}^C (Figure 4b, Proposition 3). The SSO–VCC in the Manhattan space thus reduces to a system of linear equations.

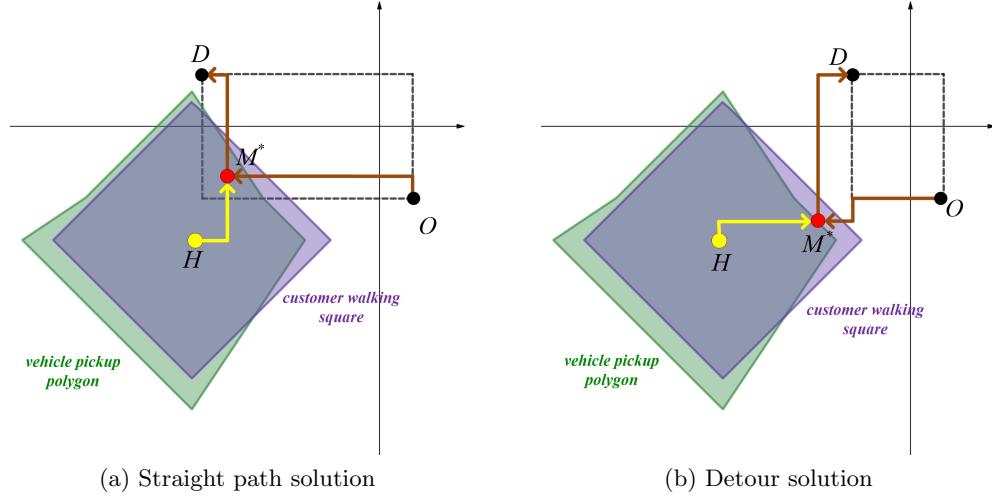


Figure 4 Geometric representation of SSO–VCC with the Manhattan distance.

PROPOSITION 3. *The optimal solution of Equation (1) satisfies $v^* = \frac{\|D-O\|_1}{\bar{S}} + \frac{2L}{\bar{S}}$, where L denotes the shortest Manhattan distance between $\mathcal{S}_I^V \cup \mathcal{S}_I^C$ and $O \square D$.*

Whenever SSO–VCC admits multiple optima, we choose one that does not lie on the “border” of the rectangle $O \square D$, if possible. This is stated in Remark 1, which will carry great importance in Section 4 (to circumvent the fact that the ℓ_1 norm is not differentiable in zero).

REMARK 1. Let $O = (x_O, y_O)$ and $D = (x_D, y_D)$. Let OPT be the set of optimal solutions, and $\mathcal{I} = \{(\chi, \gamma) : \chi \neq x_O, \chi \neq x_D, \gamma \neq y_O, \gamma \neq y_D\}$. If $OPT \cap \mathcal{I} \neq \emptyset$, we choose M^* in $OPT \cap \mathcal{I}$.

4. Multiple Stop Optimization with Vehicle-Customer Coordination

Consider a vehicle starting in origin $O \in \mathbb{R}^2$ at time $\bar{v} \geq 0$, visiting K customers in a given sequence, and ending in destination $D \in \mathbb{R}^2$. We index the visits by $k \in \{0, \dots, K+1\}$, where $k=0$ and $k=K+1$ correspond to O and D , respectively. Each customer $k=1, \dots, K$ is ready to depart at time 0 from a home location $H_k \in \mathbb{R}^2$. Let W_k and S_k be their maximum walking distance and speed, respectively. Let \bar{S} be the vehicle’s speed, with $\bar{S} > S_k$ for all $k=1, \dots, K$.

We denote by $v_k \geq 0$ and $M_k \in \mathbb{R}^2$ the time and location of visit $k=0, \dots, K+1$. The MSO–VCC minimizes the time at which the vehicle reaches its destination D . It is formulated via second-order cone optimization (in the Euclidean space) or linear optimization (in the Manhattan space):

$$[\text{MSO–VCC}] \quad \min v_{K+1}, \tag{4}$$

$$\text{s.t. } v_0 = \bar{v}, \quad M_0 = O, \quad M_{K+1} = D, \tag{5}$$

$$v_k \geq v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}} \quad \forall k = 1, \dots, K+1, \quad (6)$$

$$v_k \geq \frac{\|M_k - H_k\|}{S_k} \quad \forall k = 1, \dots, K, \quad (7)$$

$$\|M_k - H_k\| \leq W_k \quad \forall k = 1, \dots, K, \quad (8)$$

$$v_k \geq 0, \quad M_k \in \mathbb{R}^2 \quad \forall k = 0, \dots, K+1. \quad (9)$$

Analogous to Lemma 1, Lemma 5 shows that Equation (6) is binding at the optimum, i.e., the vehicle does not wait for any customer at the stopping location. However, the other SSO–VCC results do not have counterparts because of the interdependencies across the customer sequence.

LEMMA 5. *The MSO–VCC optimum satisfies $v_k = v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}}$, $\forall k = 1, \dots, K+1$.*

Exact MSO–VCC Algorithm. We make use of our efficient SSO–VCC procedure, by decomposing the multi-stop problem into a sequence of single-stop problems. This idea follows coordinate descent principles. However, MSO–VCC is a constrained and non-separable convex optimization problem, in that the feasible region of visit k depends on the vehicle’s operations through visit $k-1$. As a result, coordinate descent is not guaranteed to maintain feasibility, let alone global optimality. Below, we design a tailored coordinate descent scheme that leverages the structure of the MSO–VCC problem to ensure feasibility and global optimality upon convergence.

Note, first, that MSO–VCC can be re-written as a continuous-space dynamic program:

$$\Pi_{k-1}(M_{k-1}, v_{k-1}) = \min_{M_k \in \mathbb{F}_k(M_{k-1}, v_{k-1})} \left\{ \frac{\|M_k - M_{k-1}\|}{\bar{S}} + \Pi_k \left(M_k, v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}} \right) \right\}, \quad (10)$$

where $\Pi_k(M_k, v_k)$ denotes the minimum travel time from customer k (in location M_k at time v_k) to the destination D , and $\mathbb{F}_k(M_{k-1}, v_{k-1})$ denotes the feasible region for the location of visit k , i.e., $\mathbb{F}_k(M_{k-1}, v_{k-1}) = \left\{ M_k \in \mathbb{R}^2 : v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}} \geq \frac{\|M_k - H_k\|}{S_k}, \|M_k - H_k\| \leq W_k \right\}$.

As such, MSO–VCC could be solved with dynamic programming algorithms. These algorithms, however, would require to discretize the state space, with a potential loss of optimality. Instead, we design an exact decomposition algorithm to solve MSO–VCC optimally.

Following this reformulation, our algorithm will proceed backward, starting by optimizing visit K and moving all the way back to the first visit. Any time we update visit k , however, the times and locations of the subsequent visits may no longer be optimal—in fact, they may no longer be feasible. Our algorithm will thus combine a backward induction outer loop (to optimize each customer visit $k = K, \dots, 1$) and a forward induction inner loop (to maintain feasibility and optimality of the “tail” $k+1, \dots, K$). As Theorem 1 shows, the algorithm returns a global optimum of MSO–VCC.

THEOREM 1. *Let us assume that the norm $\|\cdot\|$ is differentiable (e.g., the ℓ_2 -norm) or the ℓ_1 -norm. Given any initial feasible solution $(M_1^{(0)}, \dots, M_K^{(0)})$, MSO–VCC can be solved to optimality by solving SSO–VCC problems iteratively.*

SKETCH OF THE PROOF (FOR DETAILS, SEE EC.2.1). We proceed by induction over K . If $K = 1$, MSO–VCC is exactly SSO–VCC. We assume that the statement holds for $K - 1$ visits. By induction, we invoke SSO–VCC to derive, from any state (M_1, v_1) , the subsequent $K - 1$ stopping locations, denoted by $\Xi^*(M_1, v_1) = (M_2^*(M_1, v_1), \dots, M_K^*(M_1, v_1))$. Let $\Phi(M_1, M_2, \dots, M_K) = \sum_{k=1}^K \frac{\|M_{k+1} - M_k\|}{S}$ denote the travel time from M_1 to the destination $M_{K+1} = D$ through M_2, \dots, M_K . We can re-formulate MSO–VCC as follows:

$$\min_{M_1 \in \mathbb{F}_1(O, \bar{v})} \left\{ \bar{v} + \frac{\|M_1 - O\|}{S} + \Phi(M_1, \Xi^*(M_1, v_1)) \right\}, \quad \text{where } v_1 = \bar{v} + \frac{\|M_1 - O\|}{S}.$$

Let $M_1^{(l)}$ and $v_1^{(l)}$ denote the location and time of visit 1 in iteration l . At each iteration $l = 1, 2, \dots$, we update the locations and times of visits $2, \dots, K$ as a function of $M_1^{(l-1)}$ and $v_1^{(l-1)}$. We then re-optimize the location and time of visit 1, given the locations and times of all subsequent visits. This is written as follows:

$$M_1^{(l)} \in \arg \min_{M_1 \in \mathbb{F}_1(O, \bar{v})} \left\{ \bar{v} + \frac{\|M_1 - O\|}{S} + \Phi(M_1, \Xi^*(M_1^{(l-1)}, v_1^{(l-1)})) \right\}, \quad \text{and } v_1^{(l)} = \bar{v} + \frac{\|M_1^{(l)} - O\|}{S}. \quad (11)$$

The algorithm terminates when $M_1^{(l+1)} = M_1^{(l)}$ (which implies that $M_k^{(l+1)} = M_k^{(l)}$ for all $k = 1, \dots, K$).

The proof then proceeds in four steps:

1. *The algorithm only involves SSO–VCC solutions.* The proof reduces each step of the algorithm to a single-stop instance, by optimizing $M_1^{(l)}$ and $v_1^{(l)}$ between the origin O and “destination” $M_2^{(l-1)}$.
2. *The algorithm does not increase costs at each iteration.* The proof constructs a solution going through $M_1^{(l)}$ at time $v_1^{(l)}$ with at most the same cost as the previous one, going through $M_1^{(l-1)}$ at $v_1^{(l-1)}$.
3. *The algorithm terminates at the optimum if $\|\cdot\|$ is differentiable.* The proof leverages the differentiability of the norm and the problem’s convexity to show optimality.
4. *The algorithm terminates at the optimum if $\|\cdot\|$ is the ℓ_1 norm.* The proof introduces infinitesimally small perturbations to circumvent the fact that the ℓ_1 norm is not differentiable in 0. \square

At each iteration, the algorithm from Theorem 1 optimizes, using SSO–VCC, the location and time of one visit (e.g., M_k and v_k) given the previous location (M_{k-1}), the next location (M_{k+1}), and the time of the previous visit (v_{k-1}). Viewing MSO–VCC as a dynamic program (Equation (10)), the algorithm can be interpreted as a backward scheme that iteratively re-optimizes each decision ($k = K, \dots, 1$), while ensuring the optimality of all subsequent decisions ($k + 1, \dots, K$). Using this interpretation, Theorem 1 guarantees global optimality without resorting to state space discretization. Viewing MSO–VCC as a convex optimization problem (Equations (4)–(9)), the algorithm can be interpreted as a tailored coordinate descent scheme—that is, a sequence of decisions to re-optimize iteratively while guaranteeing feasibility and global optimality upon convergence.

Implementation and warm start. Algorithm 1 shows the pseudo-code of the algorithm given in Theorem 1, where $\text{SSO–VCC}(O, D, \bar{v})$ denote the SSO–VCC solution from origin O to destination D , starting at time \bar{v} (Section 3). To initialize the algorithm, one could start from each customer’s home location, i.e., $M_k^{(0)} = H_k$. However, the algorithm can be time-consuming, by constantly re-optimizing the “tail” ($k + 1, \dots, K$) even with poor solutions for the first visits ($1, \dots, k - 1$). We

Algorithm 1 Exact MSO–VCC algorithm.

Input: Sequence of customers $k = 1, \dots, K$; home locations H_k ; speed S_k ; maximum walking distance W_k ; vehicle origin O and destination D ; departure time \bar{v} ; speed \bar{S} ; tolerance $\varepsilon = 0.001$.

- 1: **MSO–VCC Warm Start**
- 2: Initialize stopping locations $M_k = H_k, \forall k = 1, \dots, K$, $M_0 = O$ and $M_{K+1} = D$.
- 3: Calculate arrival times $v_k, \forall k = 1, \dots, K+1$; Set earliest arrival time $v_{K+1}^{\text{new}} = +\infty$
- 4: **while** $v_{K+1}^{\text{new}} > v_{K+1} + \varepsilon$ **do** ▷ Stop when v_{K+1}^{new} can no longer be improved
- 5: $v_{K+1}^{\text{new}} \leftarrow v_{K+1}$ and $M_k^{\text{new}} \leftarrow M_k, \forall k = 1, \dots, K$ ▷ Record incumbent solution
- 6: **for** $k \leftarrow 1$ to K **do**
- 7: $M_k \leftarrow \text{SSO–VCC}(M_{k-1}, M_{k+1}, v_{k-1}), v_k \leftarrow v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}}$ ▷ Update stop k
- 8: **end for**
- 9: $v_{K+1} \leftarrow v_K + \frac{\|D - M_K\|}{\bar{S}}$
- 10: **end while**
- 11: **Exact MSO–VCC optimization**
- 12: $k \leftarrow K$
- 13: **while** $k \neq 0$ **do** ▷ Stop when the first visit can no longer be optimized
- 14: $\bar{M} \leftarrow \text{SSO–VCC}(M_{k-1}, M_{k+1}, v_{k-1})$ ▷ Update stopping location based on SSO–VCC
- 15: **if** $\|M_k - \bar{M}\| \leq \varepsilon$ **then** ▷ Optimality proved for visit k
- 16: $k \leftarrow k - 1$ ▷ Backward induction
- 17: **else**
- 18: $M_k \leftarrow \bar{M}$ and $v_k \leftarrow v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}}$ ▷ Update visit time and location
- 19: **if** $(k = K)$, **then** $k \leftarrow k - 1$, and $v_{K+1} \leftarrow v_K + \frac{\|D - M_K\|}{\bar{S}}$ ▷ Update solution
- 20: **else** $k \leftarrow k + 1$ ▷ Check optimality of M_k , by re-optimizing M_{k+1}, \dots, M_K
- 21: **end if**
- 22: **end while**

therefore propose a warm-start acceleration—a forward heuristic that optimizes one visit at a time, also using SSO–VCC, until no improvement is obtained through the full sequence. By proceeding forward, the heuristic constantly maintains feasibility but may not reach the global optimum. We thus apply, upon termination, the main procedure from Theorem 1 to guarantee optimality.

Discussion. The MSO–VCC can be formulated as a continuous-state dynamic program or a constrained convex optimization problem—two notoriously challenging classes of problems. We have developed a tailored coordinate descent algorithm that solves it to global optimality. In EC.2.2, we report computational results showing that the algorithm outperforms direct implementation via second-order cone optimization (in the Euclidean space) or linear optimization (in the Manhattan space), by a factor of 10–50. Moreover, our warm-start procedure accelerates the algorithm by a factor of 10 in the Euclidean space and a factor of 100–1,000 in the Manhattan space.

In Section 6, we will consider operations in a discretized space, motivated by real-world ride-sharing. In EC.2.3, we show that our MSO–VCC algorithm still terminates finitely in that case, albeit not necessarily at a global optimum. Computational results show that it is still one to two orders of magnitude faster than direct implementation, and that it returns strong solutions in practice—within 1–2% of the optimum without our warm start and at the optimum with the warm start. We therefore leverage the MSO–VCC algorithm in the remainder of this paper as the core optimization engine into broader algorithms toward large-scale routing optimization with vehicle–customer coordination—in the continuous space (Section 5) and in the discrete space (Section 6).

5. Dial-A-Ride with Vehicle-Customer Coordination

The DAR–VCC is a dial-a-ride problem with capacitated vehicles and time windows, with added flexibility due to vehicle–customer coordination. On the demand side, the problem takes as inputs n customer requests, stored in a set of pickups $\mathcal{P} = \{1, \dots, n\}$ and a set of dropoffs $\mathcal{D} = \{n+1, \dots, 2n\}$. For each $j \in \mathcal{P}$, q_j customers demand transportation from a “home” origin $H_j \in \mathbb{R}^2$ to a “goal” destination $G_{j+n} \in \mathbb{R}^2$ by a deadline $\bar{T}_j \geq 0$. Let W_j be their maximum walking distance, S_j their speed, and g_j the fare. By convention, each dropoff has a negative number of customers, that is, $q_{j+n} = -q_j$. Vehicle–customer coordination results in an endogenous location for each pickup. In contrast, we assume that dropoff locations are fixed (we relax this assumption in Section 6).

On the supply side, the problem considers a set of capacitated vehicles \mathcal{V} , each starting in origin $O \in \mathbb{R}^2$ at time 0 and needing to reach a destination $D \in \mathbb{R}^2$ by time \bar{T} (if it serves at least one customer). Vehicles travel at speed \bar{S} ($\bar{S} > S_j$ for all $j \in \mathcal{P}$), carry up to Q passengers, and incur a unit cost c per unit of time. The DAR–VCC optimizes service to customers to maximize profits.

The DAR–VCC is a challenging combinatorial problem, combining vehicle routing (which customers to serve, with which vehicle, in which sequence) and vehicle–customer coordination (where and when to pick up customers). We formalize the problem via mixed-integer optimization in Section 5.1; however, direct implementation does not scale beyond small-scale instances.

To solve it, we propose a new subpath-based time-space algorithm in Section 5.2 by decomposing the problem into “subpaths”, defined as customer-carrying trips between an “empty point” (where the vehicle carries no passenger) to the next. The proposed algorithm combines a label-setting dynamic programming algorithm (to generate subpaths), our MSO–VCC algorithm (to optimize the time and location of each stop within each subpath), and a time-space network optimization model (to combine subpaths into vehicle routes from origin to destination). As such, this approach contributes to the dial-a-ride literature by combining principles of set partitioning formulations and time-space network models—namely, by involving subpath-based variables as opposed to route-based variables or arc-based variables. Our results in Section 5.3 will show that this algorithm can tackle large-scale instances that would be otherwise intractable.

A special case of the DAR–VCC is the Vehicle Routing Problem with Vehicle-Customer Coordination (VRP–VCC), in which the operator minimizes travel times to serve a set of customers. We treat this problem in Appendix A, with details in EC.3. In particular, we find that our algorithm provides higher-quality solutions in shorter runtimes than the one from Gambella et al. (2018).

5.1. Mixed-integer optimization formulation

Each vehicle makes up to $2n$ stops, indexed in $\mathcal{N} = \{1, \dots, 2n\}$. By convention, $i = 0$ and $i = 2n + 1$ correspond to the origin O and the destination D , respectively. We define the following variables:

$$x_{rij} = \begin{cases} 1 & \text{if pickup or dropoff } j \in \mathcal{P} \cup \mathcal{D} \text{ is performed in stop } i \in \mathcal{N} \text{ of vehicle } r \in \mathcal{V}, \\ 0 & \text{otherwise.} \end{cases}$$

Let $M_j \in \mathbb{R}^2$ denote the pickup location of customer $j \in \mathcal{P}$, and $M_{ri}^V \in \mathbb{R}^2$ the location of vehicle $r \in \mathcal{V}$ in stop $i \in \mathcal{N} \cup \{0, 2n + 1\}$. Let v_{ri} be the time when vehicle $r \in \mathcal{V}$ gets to stop $i \in \mathcal{N} \cup \{0, 2n + 1\}$, and Q_{ri}^P the number of passengers in vehicle $r \in \mathcal{V}$ from stop $i \in \mathcal{N} \cup \{0\}$ to stop $i + 1$.

We formulate DAR–VCC via mixed-integer second-order cone optimization (in the ℓ_2 space) or mixed-integer linear optimization (in the ℓ_1 space). In EC.4.1, we set the “big- M ” coefficients N^W , N_j^T and N_j^V , and add valid inequalities to strengthen the formulation.

$$[\text{DAR–VCC}] \quad \max \sum_{r \in \mathcal{V}} \sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - \sum_{r \in \mathcal{V}} c \cdot v_{r,2n+1}, \quad (12)$$

$$\text{s.t. } M_{r0}^V = O \quad \forall r \in \mathcal{V}, \quad (13)$$

$$M_{r,2n+1}^V = \sum_{j \in \mathcal{P}} x_{r,1,j} D + \left(1 - \sum_{j \in \mathcal{P}} x_{r,1,j}\right) O \quad \forall r \in \mathcal{V}, \quad (14)$$

$$\|M_{ri}^V - M_j\| \leq N^W (1 - x_{rij}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \quad (15)$$

$$\|M_{ri}^V - G_j\| \leq N^W (1 - x_{rij}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{D}, \forall r \in \mathcal{V}, \quad (16)$$

$$v_{ri} \geq v_{r,i-1} + \frac{\|M_{ri}^V - M_{r,i-1}^V\|}{\bar{S}} \quad \forall i \in \mathcal{N} \cup \{2n + 1\}, \forall r \in \mathcal{V}, \quad (17)$$

$$v_{ri} \geq \frac{\|M_j - H_j\|}{S_j} - N_j^T (1 - x_{rij}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \quad (18)$$

$$\|M_j - H_j\| \leq W_j \quad \forall j \in \mathcal{P}, \quad (19)$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} \geq \sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{r,i+1,j} \quad \forall i \in \mathcal{N} \setminus \{2n\}, \forall r \in \mathcal{V}, \quad (20)$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} \leq 1 \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \quad (21)$$

$$\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} \leq 1 \quad \forall j \in \mathcal{P} \cup \mathcal{D}, \quad (22)$$

$$\sum_{i \in \mathcal{N}} x_{rij} = \sum_{i \in \mathcal{N}} x_{r,i,j+n} \quad \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \quad (23)$$

$$\sum_{i \in \mathcal{N}} i x_{rij} \leq \sum_{i \in \mathcal{N}} i x_{r,i,j+n} \quad \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \quad (24)$$

$$Q_{ri}^P = Q_{r,i-1}^P + \sum_{j \in \mathcal{P} \cup \mathcal{D}} q_j x_{rij} \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \quad (25)$$

$$Q_{ri}^P \leq Q \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \quad (26)$$

$$\bar{T}_j \geq v_{ri} - N_j^V (1 - x_{r,i,j+n}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \quad (27)$$

$$\bar{T} \geq v_{r,2n+1}, \quad \forall r \in \mathcal{V}, \quad (28)$$

$$x_{rij} \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P} \cup \mathcal{D}, \forall r \in \mathcal{V}, \quad (29)$$

$$Q_{ri}^P \geq 0 \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \quad (30)$$

$$v_{ri} \geq 0 \quad \forall i \in \mathcal{N} \cup \{2n+1\}, \forall r \in \mathcal{V}, \quad (31)$$

$$M_{ri}^V \in \mathbb{R}^2 \quad \forall i \in \mathcal{N} \cup \{2n+1\}, \forall r \in \mathcal{V}, \quad (32)$$

$$M_j \in \mathbb{R}^2 \quad \forall j \in \mathcal{P}. \quad (33)$$

Equation (12) maximizes profit. Equations (13) and (14) indicate that each vehicle travels from O to D if it is utilized. Equations (15)–(18) coordinate vehicle and customer operations in space and time. Equation (19) enforces customers' maximum walking distance. Equation (20) orders stops so that stop i is visited if stop $i+1$ is visited. Per Equations (21) and (22), at most one pickup/dropoff is performed at each stop, and each pickup/dropoff is performed at most once. Equations (23) and (24) indicate that each customer that is picked up is dropped off later on. Equations (25) and (26) apply vehicle capacities. Equations (27) and (28) enforce customer deadlines. Equations (29)–(33) define the domains of all variables.

CPLEX-based benchmarks. DAR–VCC can be solved directly using MISOCO or MILO solvers. However, routing problems with capacitated vehicles and time windows are known to induce weak mixed-integer optimization formulations due to the big-M constraints (Equations (25) and (27)). In addition, our vehicle–customer coordination structure leads to additional big-M constraints linking location, sequencing and timing variables (Equations (15), (16) and (18)). As results will show, direct implementation can only handle small-scale instances.

We also implement a heuristic, referred to as “CPLEX+H”, which one could easily implement in the absence of our tailored algorithm. It decomposes the problem by adding one customer request at a time. Let \mathcal{Q} denote the subset of customers that receive a service. Starting with $\mathcal{Q} = \emptyset$, we solve the DAR–VCC iteratively, with the subset \mathcal{Q} plus one request $j = 1, \dots, n$. We update \mathcal{Q} at each iteration, until the profit does not improve over the full sequence (see EC.4.2 for details).

5.2. Subpath-based time-space network optimization

Our DAR–VCC algorithm decomposes the problem into subpaths, defined as sequences of pickups and dropoffs from one “empty point” to the next (see Figure 5). We combine the subpaths into vehicle routes from origin to destination via a time-space network optimization model.

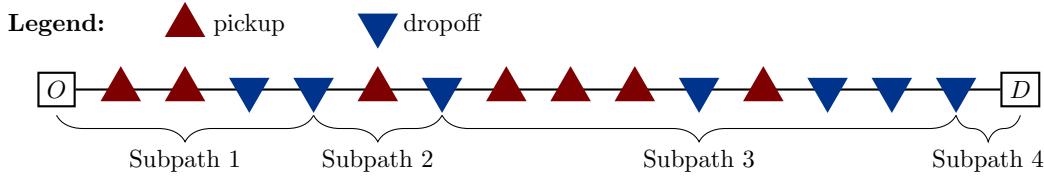


Figure 5 A vehicle path combines several customer-serving trips (subpaths) via time-space network optimization.

Specifically, let \mathcal{S} denote a set of time-space nodes for “empty vehicles” to flow, indexed by s and associated with a location $L(s)$ and time $t(s)$. We define one time-space node s^O at the vehicle’s origin, one time-space node s^D at the vehicle’s destination, and one set of time-space nodes at each customer’s dropoff location (with times spanning the customer’s earliest and latest possible arrival times). Let \mathcal{U} denote a set of time-space arcs, each joining two time-space nodes $s \in \mathcal{N}$ and $s' \in \mathcal{N}$ to indicate a trip (or subpath) where the vehicle starts empty in location $L(s)$ at time $t(s)$, picks up and drops off customers (without ever being empty), and performs its last dropoff in location $L(s')$ at time $t(s')$. Let \mathcal{U}_s^+ and \mathcal{U}_s^- store the time-space arcs starting from and ending in $s \in \mathcal{S}$, respectively. For each $u \in \mathcal{U}$, a^{uj} denotes a binary indicator equal to 1 if customer j is served by arc $u \in \mathcal{U}$ and 0 otherwise, and g^u denotes the profit contribution of the trip.

We generate the set of time-space arcs \mathcal{U} via a label-setting dynamic programming algorithm. The algorithm starts from each time-space node $s \in \mathcal{S} \setminus \{s^D\}$. It extends subpaths to pickups and dropoffs iteratively, while removing infeasible and dominated ones. To capture vehicle-customer coordination, we embed our MSO–VCC optimization (Algorithm 1) to determine, for each subpath, the optimal pickup locations and travel times. For details, we refer to EC.4.3.

Our subpath-based time-space network optimization relies on the following decision variables:

$$z^u = \begin{cases} 1 & \text{if one of the vehicles traverses time-space arc } u \in \mathcal{U} \\ 0 & \text{otherwise} \end{cases}$$

Λ = number of vehicles used to serve customers.

The formulation maximizes profit, subject to flow balance, packing and budget constraints:

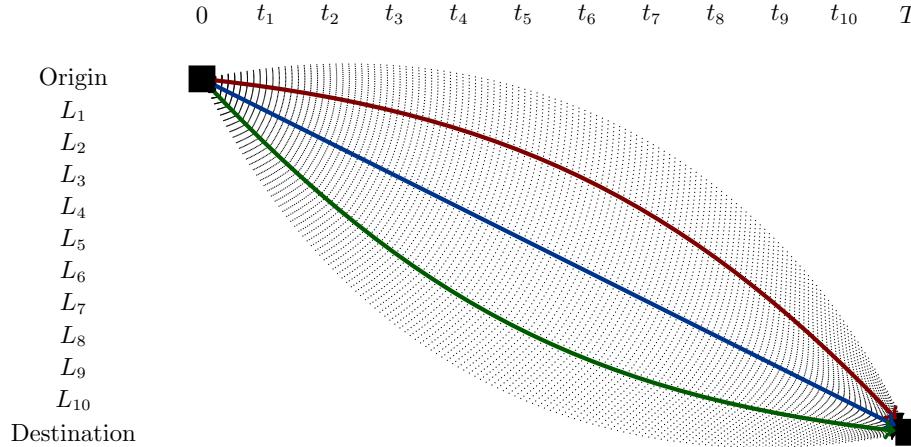
$$\max_{u \in \mathcal{U}} \sum g^u z^u, \quad (34)$$

$$\text{s.t. } \sum_{u \in \mathcal{U}_s^+} z^u - \sum_{u \in \mathcal{U}_s^-} z^u = \begin{cases} 0 & \forall s \neq s^O, s^D \\ \Lambda & \text{for } s = s^O \\ -\Lambda & \text{for } s = s^D \end{cases} \quad (35)$$

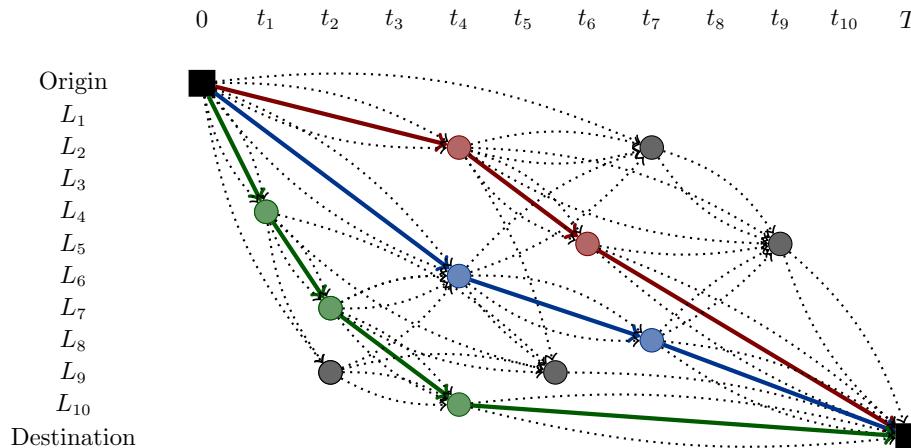
$$\sum_{u \in \mathcal{U}} a^{uj} z^u \leq 1, \quad \forall j \in \mathcal{P}, \quad (36)$$

$$0 \leq \Lambda \leq |\mathcal{V}|, \quad (37)$$

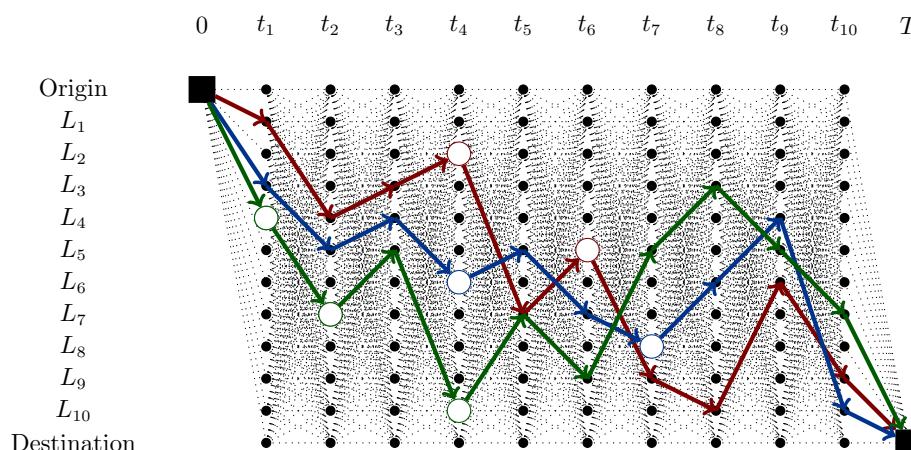
$$z^u \in \{0, 1\}, \quad \forall u \in \mathcal{U}. \quad (38)$$



(a) Set partitioning formulation: exponential number of route-based variables from origin to destination



(b) Subpath-based formulation: moderate number of variables in a sparse time-space network of “empty points”



(c) Arc-based formulation: large number of variables in dense time-space networks

Figure 6 Schematic representation of route-based, subpath-based and arc-based formulations.

As compared to the mixed-integer optimization formulation in Equations (12)–(33), our subpath-based time-space model (Equations (34)–(38)) yields a tighter formulation by avoiding big-M constraints. As compared to set partitioning formulations, it induces sparsity by decomposing the problem into subpath-based variables from one empty point to the next, as opposed to route-based variables from origin to destination—therefore alleviating the need for column generation algorithms. And as compared to arc-based formulations, it also induces sparsity by defining subpath-based variables (as opposed to more granular arc-based variables) in a single time-space network (as opposed to one time-space network for each vehicle and one for each customer to coordinate operations). Viewed through this lens, our subpath-based algorithm provides a new sparse and tight approach to dial-a-ride that combines set partitioning principles and time-space principles (see Figure 6). Its success hinges on the fact that vehicles are regularly empty between customer-serving trips, which is often the case in practice—as indicated by our results in Sections 5.3 and 6.2.

Due to the exactness of the MSO–VCC algorithm in the continuous Euclidean or Manhattan space (Theorem 1), our DAR–VCC algorithm solves the subpath-based time-space network formulation (Equations (34)–(38)) to optimality. In the absence of time windows, this formulation is equivalent to the mixed-integer formulation (Equations (12)–(33)) as time discretization becomes infinitesimally granular (see EC.4.4). However, it may introduce discretization errors with time windows. This is a common issue in time-space optimization, which has led to dedicated research (Boland et al. 2017, Boland and Savelsbergh 2019). Yet, our subpath-based time-space formulation introduces smaller approximation errors than traditional arc-based time-space formulations because discretization is only applied at “empty points” (i.e., dropoff locations when vehicles become empty) as opposed to all pickup and dropoff locations. Moreover, as our results will show, our DAR–VCC algorithm will consistently yield optimal solutions in small-scale instances that can be solved directly with CPLEX—and, as mentioned earlier, will scale to much larger instances.

5.3. Computational results

We generate synthetic data to implement our DAR–VCC algorithm. We consider n single-customer requests, with origins and destinations uniformly sampled in an 8km-by-3km rectangle centered at $(0, 0)$. All vehicles start in $(-2\text{km}, 0)$ and end in $(2\text{km}, 0)$, with a one-hour time window. Customers’ speeds are uniformly sampled between 3.2 and 5.4 km/h, and each vehicle’s speed is set to 20 km/h. We assume a maximum walking distance of 300 m. We set a fare of \$0.86/km times the direct distance $\|G_{j+n} - H_j\|$, and a vehicle cost of \$3.2/h. We vary the number of requests n (from 5 to 200), the vehicles’ capacity Q (from 3 to 6), and the customers’ “relative deadline” μ , defined as the maximum percent-wise increase from the ideal trip with no wait, no walk and no detour (from 20% to 50%). We generate five random instances (made available online) in each setting, referred

to as “ $p_n_Q_\mu$ ”. For all runs, we use a Xeon E5-2680 processor running at 2.80 GHz. Results are shown in Table 1 in the continuous two-dimensional Euclidean space—we establish the robustness of our findings in EC.4.5 in the continuous Manhattan space and in the discretized space.

Table 1 Average computational results for DAR–VCC (3 vehicles).

Instances	CPLEX					CPLEX+H			Our algorithm	
	Sol.	UB	Opt. gap	Gap vs. alg.	CPU (s)	Sol.	Gap vs. alg.	CPU (s)	Sol.	CPU (s)
p_5_3_0.5	18.3	18.3	0%	0%	66	18.3	0%	110	18.3	<1
p_6_3_0.3	11.9	11.9	0%	0%	1,060	11.9	0%	71	11.9	<1
p_7_4_0.3	10.7	20.6	92%	58%	>3,600	16.9	0%	>3,600	16.9	<1
p_8_4_0.4	9.6	28.6	198%	159%	>3,600	18.9	32%	>3,600	24.9	<1
p_9_5_0.5	9.9	28.9	193%	147%	>3,600	15.8	54%	2,236	24.4	<1
p_10_6_0.5	2.2	29.3	1,236%	707%	>3,600	14.0	26%	>3,600	17.7	<1
p_12_4_0.3	5.8	40.7	597%	313%	>3,600	18.4	31%	>3,600	24.1	<1
p_15_5_0.5	6.4	56.3	787%	568%	>3,600	18.6	128%	>3,600	42.4	<1
p_20_3_0.5	11.9	66.6	462%	172%	>3,600	12.5	158%	>3,600	32.2	<1
p_25_3_0.2	-	-	-	-	-	19.7	19%	>3,600	23.5	<1
p_25_5_0.4	4.6	94.4	1,936%	901%	>3,600	20.2	130%	>3,600	46.4	<1
p_50_4_0.2	-	-	-	-	-	15.4	98%	>3,600	30.4	<1
p_50_6_0.2	-	-	-	-	-	12.3	147%	>3,600	30.4	2
p_50_4_0.4	-	-	-	-	-	15.9	253%	>3,600	56.0	40
p_50_6_0.4	-	-	-	-	-	17.9	253%	>3,600	63.2	340
p_75_6_0.4	-	-	-	-	-	13.6	359%	>3,600	62.6	246
p_75_6_0.5	-	-	-	-	-	12.5	510%	>3,600	76.6	2,655
p_100_4_0.2	-	-	-	-	-	15.9	162%	>3,600	41.7	12
p_100_4_0.3	-	-	-	-	-	15.8	266%	>3,600	57.8	180
p_100_4_0.4	-	-	-	-	-	16.9	292%	>3,600	66.2	1,852
p_125_5_0.2	-	-	-	-	-	14.8	235%	>3,600	49.5	126
p_125_5_0.4	-	-	-	-	-	12.4	488%	>3,600	73.1	2,530
p_150_4_0.3	-	-	-	-	-	13.4	330%	>3,600	57.6	390
p_150_6_0.3	-	-	-	-	-	15.6	315%	>3,600	64.6	2,300
p_175_5_0.2	-	-	-	-	-	16.4	247%	>3,600	56.9	678
p_175_4_0.4	-	-	-	-	-	13.4	397%	>3,600	66.7	4,227
p_200_3_0.3	-	-	-	-	-	13.9	256%	>3,600	49.5	411
p_200_4_0.3	-	-	-	-	-	15.1	317%	>3,600	63.1	2,843

“>3,600” means that the instance cannot be solved to optimality within one hour.

Opt. gap: (CPLEX upper bound – CPLEX solution)/CPLEX solution.

Gap vs. alg.: (Solution with our algorithm – benchmark)/benchmark, where “benchmark” refers CPLEX or CPLEX+H.

Note that, due to the combinatorial complexity of routing optimization and the added complexity of vehicle-customer coordination, direct MISOCO implementation does not scale to even medium-sized instances. Indeed, CPLEX can only solve instances with up to 6–8 customers, returns a very loose optimality gap with 10–25 customers, and does not even find a feasible solution with more than 25 customers. CPLEX+H achieves better scalability: it terminates faster in small-scale instances and consistently returns feasible solutions—albeit, rarely terminating in 1 hour.

We now turn to our main observation: our algorithm significantly dominates both benchmarks, both in terms of solution time and solution quality. It returns the optimal solution in less than 1 second in all instances with up to 50 customers, whereas CPLEX and CPLEX+H can require over an hour. Moreover, it can handle instances with up to 200 customers, in 5–70 minutes (we establish the robustness of these results with respect to the maximum walking distance in EC.4.6). As compared to CPLEX and CPLEX+H, our algorithm improves profits—by a wide margin—in all

instances with more than 6–7 customers. In summary, our algorithm yields a Pareto improvement as compared to both benchmarks: far superior solutions in shorter computational times.

Note, also, that the complexity of the problem does not solely increase with the number of customers, but also with vehicle capacities and customer deadlines. Indeed, both parameters lead to more candidate subpaths in the label-setting algorithm. Among these, vehicle capacities are not the most binding constraints—the average subpath pools around two customers. In contrast, customer deadlines have a stronger impact both on the optimal profit and on computational complexity.

5.4. Practical results

Table 2 evaluates the benefits of our approach against three benchmarks:

- “No VCC”: dial-a-ride optimization without vehicle-customer coordination ($W_j = 0$ for all j).
- “Fix Seq. + VCC”: we optimize vehicle paths without vehicle-customer coordination; fix the sequence of pickups and dropoffs for each vehicle; and re-optimize DAR–VCC operations. This is a benchmark where an operator revisits operations at the “downstream” level, by leveraging vehicle-customer coordination to re-optimize stopping times and locations (all else equal).
- “Fix Cust. + VCC”: we optimize operations without vehicle-customer coordination; fix the set of served customers; and re-optimize DAR–VCC operations. This is a benchmark where an operator revisits operations at the “intermediate” level, by re-optimizing vehicle-customer assignments, customer sequences, and stopping times and locations.

In comparison, our algorithm leverages vehicle-customer coordination to re-optimize overall operations: which customers to serve, with which vehicles, in which sequence, where and when.

Benefit of re-optimization. The first and main observation is that re-optimizing operations with vehicle-customer coordination can provide significant benefits. Indeed, the “Fix Seq. + VCC” solution improves profits by less than 1%, as compared to “No VCC”, whereas “Fix Cust. + VCC” improves profits by up to 5% and our solution consistently improves profits by 5–35%. These results suggest that vehicle-customer coordination provides some benefits at the “downstream” level, by enabling more convenient pickup times and locations, but these benefits remain limited. In comparison, vehicle-customer coordination yields much more significant gains when accompanied by a comprehensive re-optimization of overall routing operations. In other words, these results underscore the need to support vehicle-customer coordination—and similar operating innovations—with dedicated optimization models and algorithms (as done in this paper).

Sensitivity. Figure 7 shows the sensitivity of the benefits of vehicle-customer coordination (that is, the relative profit increase from our DAR–VCC solution, as compared to “No VCC”) to the model parameters. Note that increasing the maximum walking distance from 0 to 100 meters has a slightly smaller impact than increasing it from 100 to 200 meters. This, again, underscores the benefits of “upstream” optimization, in that a small walking distance creates opportunities to save

Table 2 Impact of re-optimization on DAR–VCC profits ($W_j = 0.2$).

Instances	\mathcal{V}	No VCC		Fix Seq. + VCC		Fix Cust. + VCC		Our algorithm	
		Profit	% change	Profit	% change	Profit	% change	Profit	% change
p-50_4_0.4	1	16.1	(base)	16.1	(+0.4%)	16.1	(+0.4%)	19.1	(+18.9%)
	3	45.1	(base)	45.3	(+0.5%)	45.3	(+0.5%)	53.6	(+18.7%)
	5	70.3	(base)	70.5	(+0.4%)	70.9	(+0.9%)	85.4	(+21.5%)
	10	109.8	(base)	110.4	(+0.5%)	111.8	(+1.8%)	136.0	(+23.9%)
	15	133.1	(base)	133.1	(+0.0%)	137.3	(+3.2%)	152.2	(+14.4%)
	20	143.2	(base)	143.4	(+0.1%)	149.4	(+4.3%)	152.6	(+6.6%)
	25	144.2	(base)	144.5	(+0.2%)	151.4	(+5.0%)	152.6	(+5.8%)
	30	144.2	(base)	144.5	(+0.2%)	151.4	(+5.0%)	152.6	(+5.8%)
p-100_4_0.4	1	17.9	(base)	18.0	(+0.4%)	18.0	(+0.4%)	21.4	(+19.4%)
	3	48.7	(base)	48.9	(+0.4%)	49.0	(+0.6%)	62.6	(+28.7%)
	5	78.8	(base)	79.1	(+0.4%)	79.2	(+0.5%)	99.0	(+25.6%)
	10	146.3	(base)	146.8	(+0.4%)	147.2	(+0.7%)	174.8	(+19.5%)
	15	196.0	(base)	196.9	(+0.5%)	198.9	(+1.5%)	235.7	(+20.2%)
	20	233.2	(base)	234.4	(+0.5%)	238.6	(+2.3%)	279.5	(+19.9%)
	25	255.0	(base)	256.5	(+0.6%)	262.8	(+3.0%)	302.9	(+18.8%)
	30	270.2	(base)	271.9	(+0.6%)	281.0	(+4.0%)	309.6	(+14.6%)
p-150_4_0.4	1	17.4	(base)	17.5	(+0.4%)	17.5	(+0.6%)	23.4	(+34.4%)
	3	50.9	(base)	51.1	(+0.5%)	51.1	(+0.6%)	67.5	(+32.7%)
	5	82.0	(base)	82.4	(+0.5%)	82.4	(+0.5%)	105.8	(+29.0%)
	10	149.3	(base)	150.0	(+0.4%)	150.1	(+0.5%)	188.0	(+25.9%)
	15	201.6	(base)	201.8	(+0.1%)	204.8	(+1.6%)	260.0	(+28.9%)
	20	247.6	(base)	248.0	(+0.2%)	253.3	(+2.3%)	321.3	(+29.8%)
	25	284.0	(base)	284.7	(+0.3%)	292.7	(+3.1%)	369.2	(+30.0%)
	30	308.8	(base)	310.0	(+0.4%)	321.0	(+4.0%)	401.7	(+30.1%)

costs by adjusting stopping locations whereas a larger walking distance adds flexibility that enables the operator to serve more customers. Nonetheless, as the maximum walking distance increases further, the benefits of vehicle-customer coordination are marginally decreasing (Figure 7a). In our example, allowing customers to walk 200 meters can increase profits by 32%, whereas doubling that limit to 400 meters boosts the profit increase to only 42%. From a practical standpoint, this suggests that even limited flexibility via vehicle-customer coordination can result in significant improvements in operating profitability.

Next, the benefits of vehicle-customer coordination are strongest with an “intermediate” number of vehicles (Figure 7b). At one extreme, when demand far exceeds supply, the vehicles can select the “best” customers, leading to high utilization even without vehicle-customer coordination. At the other extreme, supply is sufficient to handle demand, so vehicle-customer coordination leads to “downstream” cost savings but limited re-optimization opportunities. In-between, vehicle-customer coordination provides much-needed flexibility, enabling vehicles to serve more customers within their deadlines. Therefore, the operating flexibility from vehicle-customer coordination is most beneficial for “balanced” systems in terms of demand and supply. For similar reasons, the percent-wise benefits of vehicle-customer coordination are higher for tighter deadlines (Figure 7c). Again, vehicles can serve demand effectively with looser deadlines, so vehicle-customer coordination enables cost savings at the “downstream” level but may not significantly alter which customers receive a service. In contrast, with tight deadlines, the spatial-temporal patterns of customer demand

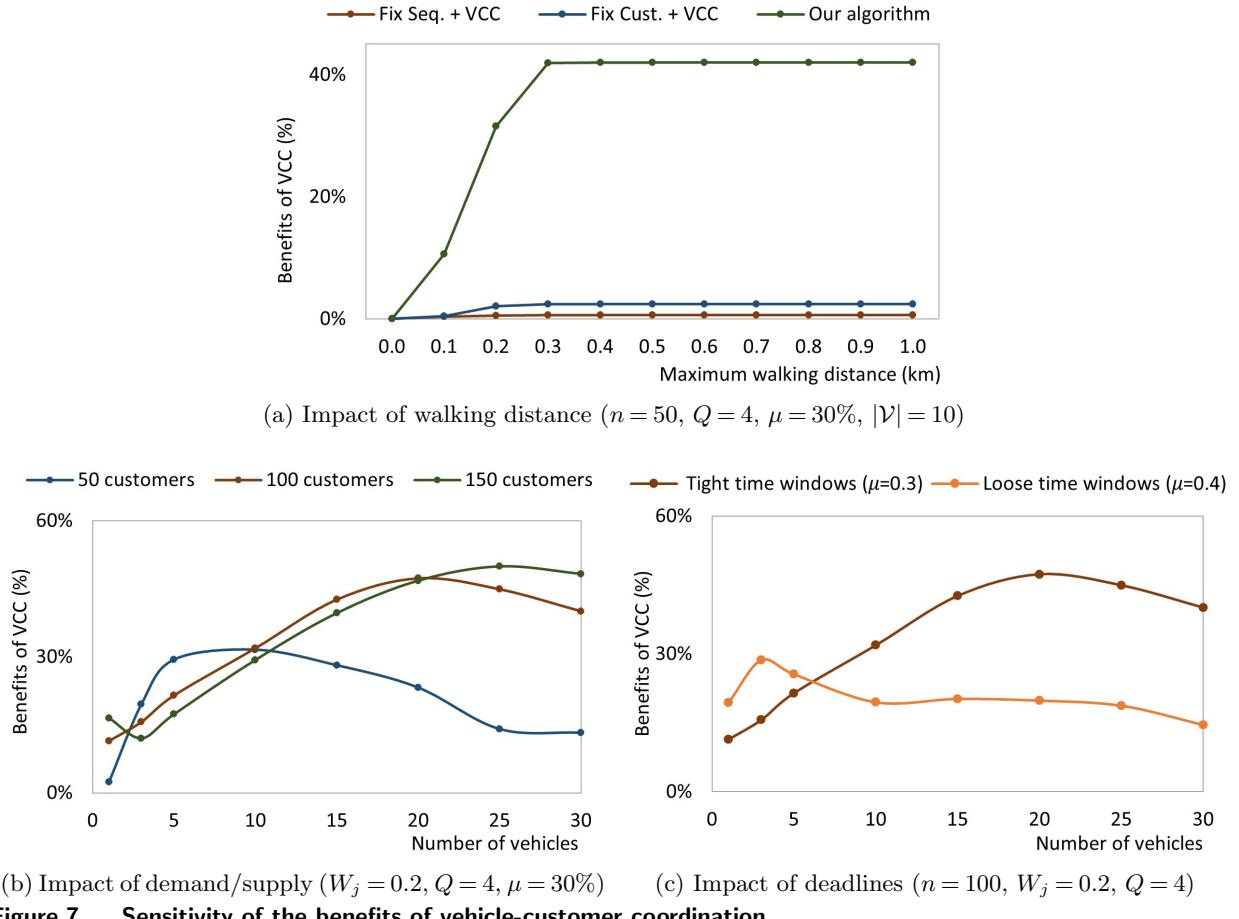


Figure 7 Sensitivity of the benefits of vehicle-customer coordination.

may be such that vehicles will ‘miss’ many customers, so the time savings from vehicle-customer coordination enable vehicles to serve more customers (or higher-margin customers).

6. Online Dial-A-Ride with Vehicle-Customer Coordination

The offline DAR–VCC took as input a static set of customer requests. In many practical settings, however, customer demand arises dynamically, thus providing operators with limited visibility into future demand. Moreover, even with perfect information, the scale of real-world operations is often so large that the DAR–VCC could not be solved within reasonable computational times.

We thus tackle the online variant, referred to as O–DAR–VCC. We proceed via batching and optimization by aggregating customer requests every 10 seconds and then matching them to vehicles using our DAR–VCC algorithm. By design, this approach ignores future demand information. Although myopic, it is easy to implement, and is consistent with practice and with the literature (Alonso-Mora et al. 2017, Vazifeh et al. 2018, Bertsimas et al. 2019, Yan et al. 2020). Yet, to alleviate the impact of myopic decision-making, we implement a history-based repositioning heuristic and an online re-optimization procedure. We apply our approach to real-world operations in New York

City to (i) demonstrate the viability of our algorithm in support of real-time decision-making in very large-scale systems, and (ii) assess the impact of vehicle-customer coordination in practice.

To capture real-world operations, we also relax three assumptions from the DAR–VCC.

- We do not fix vehicle destinations; instead, vehicle operations are driven by customer demand.
- We allow for flexible dropoffs, besides flexible pickups. To use our DAR–VCC algorithm, we make the simplifying, albeit realistic, assumption that, whenever a vehicle carries one customer (i.e., single-customer trip, end of pooled trip), it will reach the customer’s destination.
- We use OpenStreetMap (2021) to construct a discretized road network and compute travel times—capturing one-way streets, congestion, etc. We use demand data from the NYC Taxi & Limousine Commission (2021). We filter the trips starting between 5:00 PM and 6:30 PM from December 1, 2019 to December 14, 2019. After removing outliers, we end up with 15,000–20,000 requests over 90 minutes. This simulation environment follows Alonso-Mora et al. (2017), Vazifeh et al. (2018) and Bertsimas et al. (2019) (see EC.5.1 for details). In addition, to capture vehicle-customer coordination, we define a discrete set of ten candidate stopping locations for each pickup and each dropoff (e.g., road intersections).

6.1. Online optimization

In the O–DAR–VCC, each customer request j comes at a specific time $v_j^R \geq 0$. We use a batching window of $\delta v = 10$ seconds. We budget 5 seconds for optimization and 5 seconds for information sharing and implementation. The trips that start immediately are dispatched, but others will be subject to subsequent re-optimization. So, at each decision epoch, the algorithm optimizes service to “new” and “backlogged” customers. Specifically, the O–DAR–VCC proceeds as follows between epochs t and $t + 1$ (i.e., between time $t \cdot \delta v$ and time $(t + 1) \cdot \delta v$) (see Figure 8):

- *Demand*: new customers (who requested service between $t - 1$ and t), and backlogged customers (who requested service earlier but have not received a service yet).
- *Supply inputs*. Vehicles that are idle at time $(t + 1) \cdot \delta v$ are available for immediate dispatch. Others can only be assigned a trip starting after the completion of their ongoing trip.
- *Optimization*. We optimize operations using a variant of DAR–VCC in the discretized space, based on information available at time $t \cdot \delta v$. We also implement a heuristic to support the repositioning of empty vehicles, in order to alleviate the costs of spatial-temporal imbalances.
- *Commitment*. New customers are notified whether their request is accepted or rejected. Rejected customers leave the system (in practice, this can be implemented via a “no available vehicle” message, a high price, or a long wait time). Accepted customers are either served immediately or will need to be served in the future (by their deadlines). This choice prevents the platform from keeping a customer waiting for a long period of time before rejecting them.

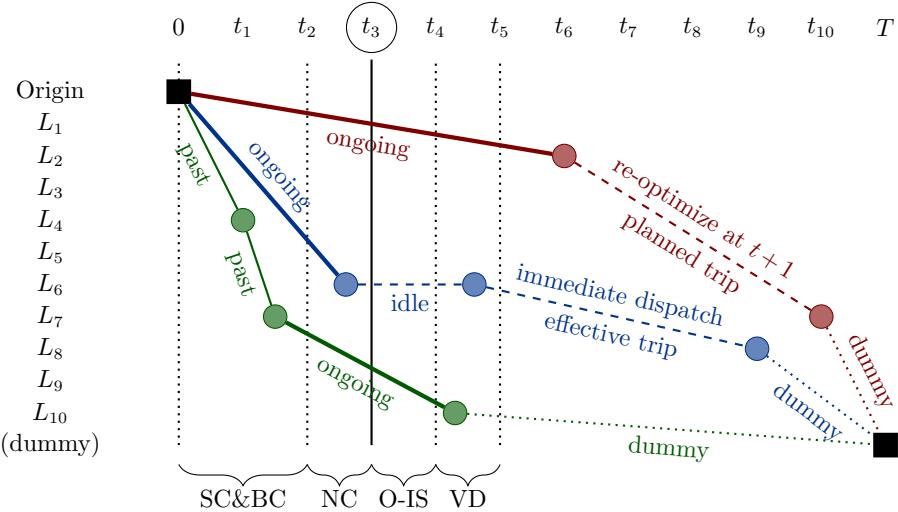


Figure 8 Illustration of the sequence of events of epoch t_3 (“SC”: served customers; “BC”: backlogged customers; “NC”: new customers; “O-IS”: optimization and information pooling; “VD”: vehicle dispatch).

- *Dispatch.* All vehicles that start a trip (either a customer-serving one or a repositioning one) between $(t+1)\cdot\delta v$ and $(t+2)\cdot\delta v$ are dispatched immediately. En-route vehicles continue their ongoing trips. Other vehicles remain in their current location. Similarly, all customers on the path of a dispatched vehicle will be instructed to walk to their pickup location. However, customers that are accepted but assigned to a trip that starts after $(t+2)\delta v$ will not be provided pickup information, to allow for subsequent re-optimization.

From an algorithmic standpoint, the O–DAR–VCC has three main implications. First, the discretized network simplifies the SSO–VCC algorithm: as opposed to relying on our geometric results (Section 3), we can solve SSO–VCC through exhaustive search. Second, our MSO–VCC algorithm (Algorithm 1) no longer benefits from the theoretical guarantees of Theorem 1 because of the lack of differentiability in the discretized space; nonetheless, the algorithm exhibits strong practical performance in the discretized space (see EC.2.3 and EC.4.5). Third, we modify our DAR–VCC algorithm to account for online dynamics and for flexible dropoffs (see EC.5.2 for details).

In addition, we develop a history-based acceleration heuristic. Indeed, the model involves 50–150 customers at each epoch, for a system of the size of Manhattan. Although highly efficient, our DAR–VCC algorithm exceeds the five-second limit in 2% of the cases. Our heuristic reduces the number of time-space arcs by restricting the search to “synergistic” customers at each node (see EC.5.2 for details). Such heuristic acceleration is consistent with the literature (see, e.g., Santi et al. 2014, Alonso-Mora et al. 2017, Vazifeh et al. 2018, Bertsimas et al. 2019). One difference, however, is that our arc-reduction heuristic does not aim to accelerate the resulting network optimization model but the label-setting dynamic programming algorithm for pre-processing. Ultimately, the heuristic enables to consistently solve large-scale instances in seconds at moderate loss in terms of performance, thus enabling online implementation in practice (see EC.5.3).

6.2. Results

Figure 9 illustrates O-DAR-VCC operations in Manhattan, showing that vehicle-customer coordination is most prevalent in areas of high demand (e.g., Midtown). Indeed, under high demand, the operator may not be able to serve all customers, so the extra flexibility is highly beneficial to free up capacity and serve additional customers. This observation echoes our result from Section 5: vehicle-customer coordination is most beneficial when spatial-temporal imbalances are most costly.

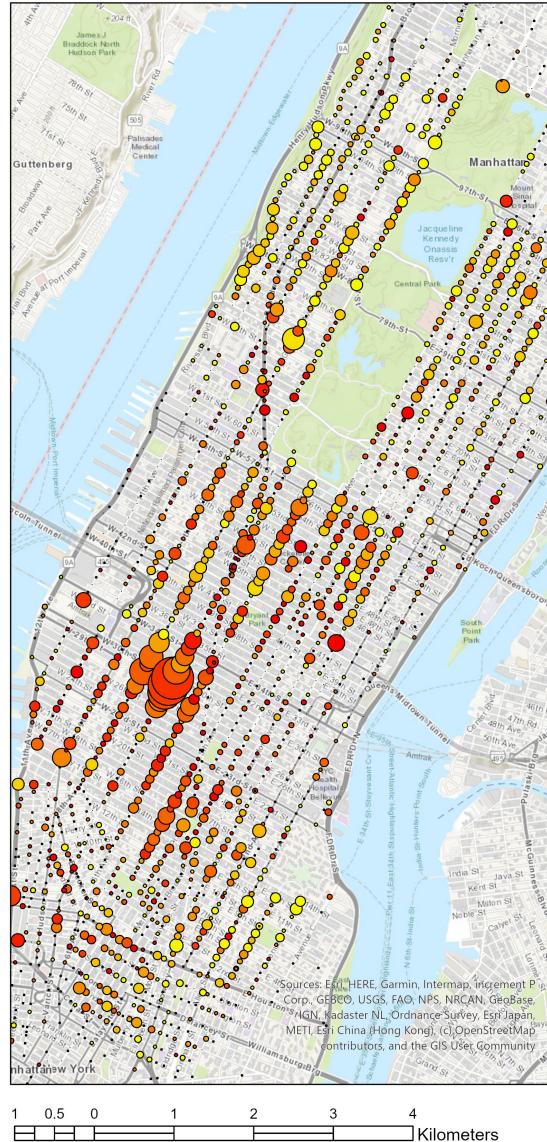


Figure 9 Customer view of O-DAR-VCC operations in Manhattan. The size of each dot indicates the number of accepted requests, and its color (yellow to red) indicates the proportion of trips that make use of vehicle-customer coordination (low to high).

Table 3 reports average performance metrics over the 14 days, characterizing profits, customer service, vehicle-customer coordination, ride-pooling, and vehicle operations. The first and main observation is that vehicle-customer coordination can yield significant benefits, with a 6.2% profit increase (with 3,000 vehicles). For a system of the size of Manhattan, this represents an estimated gain of \$100,000 daily, or \$35–40M annually. We break down these benefits in Table 4 to generate further insights on the impacts of vehicle-customer coordination.

Table 3 Average performance metrics over the 14 days, with 1,000, 2,000 and 3,000 vehicles.

V	Metric	No ride-pooling		Ride-pooling			
		No VCC	VCC:PU	No VCC	VCC:PU	VCC:DO	VCC:PU-DO
1,000	Profit	\$113,776	\$118,462	\$115,602	\$119,956	\$115,880	\$121,068
	Revenue	\$117,448	\$122,101	\$118,869	\$123,034	\$119,116	\$124,118
	Cost	\$3,672	\$3,638	\$3,267	\$3,078	\$3,236	\$3,050
	Cost per request	\$0.40	\$0.37	\$0.34	\$0.31	\$0.34	\$0.30
	Acceptance rate	57%	61%	59%	62%	59%	62%
	VCC: Pickups away from origin	—	15%	—	18%	—	18%
	VCC: Dropoffs away from destination	—	—	—	—	3%	4%
	Pooled trips	—	—	19%	24%	19%	25%
	Vehicle miles traveled (total)	30,600	30,291	27,004	25,458	26,785	25,291
	Vehicle miles traveled (per request)	3.29	3.08	2.85	2.56	2.82	2.51
2,000	Profit	\$136,752	\$141,534	\$142,126	\$146,410	\$142,653	\$147,256
	Revenue	\$142,382	\$146,872	\$146,757	\$150,577	\$147,250	\$151,378
	Cost	\$5,629	\$5,338	\$4,631	\$4,167	\$4,597	\$4,123
	Cost per request	\$0.47	\$0.43	\$0.37	\$0.32	\$0.37	\$0.32
	Acceptance rate	73%	77%	77%	80%	77%	80%
	VCC: Pickups away from origin	—	10%	—	14%	—	14%
	VCC: Dropoffs away from destination	—	—	—	—	3%	4%
	Pooled trips	—	—	18%	24%	19%	24%
	Vehicle miles traveled (total)	46,375	44,106	38,343	34,494	38,082	34,193
	Vehicle miles traveled (per request)	3.90	3.54	3.10	2.68	3.06	2.64
3,000	Profit	\$137,888	\$144,378	\$144,662	\$152,697	\$144,757	\$153,714
	Revenue	\$144,806	\$151,041	\$150,330	\$157,615	\$150,314	\$158,503
	Cost	\$6,917	\$6,663	\$5,667	\$4,919	\$5,556	\$4,789
	Cost per request	\$0.57	\$0.52	\$0.44	\$0.36	\$0.44	\$0.35
	Acceptance rate	75%	80%	79%	84%	79%	85%
	VCC: Pickups away from origin	—	9%	—	13%	—	13%
	VCC: Dropoffs away from destination	—	—	—	—	3%	4%
	Pooled trips	—	—	18%	23%	19%	26%
	Vehicle miles traveled (total)	56,803	54,471	46,705	40,619	45,720	39,612
	Vehicle miles traveled (per request)	4.67	4.22	3.66	2.98	3.58	2.88

“VCC:PU”, “VCC:DO”, “VCC:PU-DO”: vehicle-customer coordination for pickups, dropoffs and both.

Benefits of re-optimization. We decompose the profit increase into cost reduction versus change in the customer mix. Let π , P , C and q refer to the profit, the average fare per request, the average cost per request and the number of accepted requests, respectively, and let the “0” and “1” subscripts refer to outcomes without and with vehicle-customer coordination. We have:

Table 4 Drivers of the profit increase from vehicle-customer coordination, averaged over the 14 days.

V	Metric	No ride-pooling		Ride-pooling			
		No VCC	VCC:PU	No VCC	VCC:PU	VCC:DO	VCC:PU-DO
1,000	Profit	\$113,776	\$118,462	\$115,602	\$119,956	\$115,880	\$121,068
	Profit increase	(base)	4.1%	(base)	3.6%	0.2%	4.5%
	Revenue increase	(base)	3.9%	(base)	3.4%	0.1%	4.2%
	Cost decrease	(base)	1.0%	(base)	5.9%	1.0%	6.7%
	Downstream contribution	(base)	-0.7%	(base)	15.8%	-2.4%	10.4%
	Upstream contribution	(base)	100.7%	(base)	84.2%	102.4%	89.6%
	Max discount (uniform)	(base)	3.7%	(base)	3.4%	0.2%	4.2%
	Max discount (targeted)	(base)	12.5%	(base)	8.6%	1.3%	9.4%
	Max discount (prorated, per 100m)	(base)	\$1.9	(base)	\$1.2	\$0.2	\$1.2
	Profit	\$136,752	\$141,534	\$142,126	\$146,410	\$142,653	\$147,256
2,000	Profit	\$137,888	\$144,378	\$144,662	\$152,697	\$144,757	\$153,714
	Profit increase	(base)	4.8%	(base)	5.6%	0.0%	6.2%
	Revenue increase	(base)	4.4%	(base)	4.9%	0.0%	5.4%
	Cost decrease	(base)	3.6%	(base)	13.1%	2.0%	15.4%
	Downstream contribution	(base)	10.2%	(base)	13.7%	1.2%	14.2%
	Upstream contribution	(base)	89.8%	(base)	86.3%	98.8%	85.8%
	Max discount (uniform)	(base)	4.4%	(base)	5.1%	0.0%	5.7%
	Max discount (targeted)	(base)	24.1%	(base)	18.0%	-0.2%	16.0%
	Max discount (prorated, per 100m)	(base)	\$3.6	(base)	\$2.5	\$0.0	\$2.0

“Downstream contribution”: share of the profit increase from downstream cost savings, i.e., $\Delta\pi^D/\Delta\pi$.

“Upstream contribution”: share of the profit increase from upstream change in customer mix, i.e., $\Delta\pi^U/\Delta\pi$.

Max. discount: largest discounts for the operator to break even, under the uniform, targeted and prorated schemes.

$$\pi_0 = (P_0 - C_0) \cdot q_0$$

$$\pi_1 = (P_1 - C_1) \cdot q_1$$

$$\Delta\pi = \pi_1 - \pi_0 = \underbrace{(C_0 - C_1) \cdot q_0}_{\Delta\pi^D: \text{downstream cost savings}} + \underbrace{[(P_1 \cdot q_1 - P_0 \cdot q_0) - C_1 \cdot (q_1 - q_0)]}_{\Delta\pi^U: \text{upstream change in customer mix}}$$

This decomposition identifies two sources of profit increase: (i) cost savings at the “downstream” level, by meeting customers in more convenient locations, and (ii) profit improvements at the “upstream” level, by serving more customers or higher-margin customers. Table 4 shows that the second term has a much larger contribution to the profit increase than the first one (over 80%). These results confirm one of the main insights from Section 5: the main benefits of vehicle-customer

coordination do not stem from downstream adjustments in vehicle routes keeping all else equal, but in leveraging the time savings to re-optimize upstream operations using our methodology.

Flexible pickups vs. flexible dropoffs. Vehicle-customer coordination is much more beneficial for pickups than for dropoffs. By themselves, flexible pickups induce a profit increase of 3–6%, as opposed to 0–0.5% for flexible dropoffs. One explanation is that dropoffs generally occur later over the course of a vehicle trip, which limits their ripple effects for other customers (especially since most trips serve a single customer). A second explanation is that flexible dropoffs are beneficial for vehicles but induce longer travel times for customers (who need to walk to their destination). In contrast, flexible pickups cut travel times for vehicles (by inducing a smoother path) but also for customers (by mitigating wait times), which induces synergies to meet customer deadlines.

Synergies with ride-pooling. We observe two-way synergies between vehicle-customer coordination and ride-pooling. On the one hand, ride-pooling increases the use of vehicle-customer coordination by 3%–4% (see “VCC:PU” without and with ride-pooling). Indeed, single-customer trips leave limited coordination flexibility between the vehicle’s dispatch and the customer’s pickup. In contrast, pooled trips offer more time for subsequent customers to walk to a convenient pickup location, and more opportunities for the vehicle to optimize its path. At the same time, vehicle-customer coordination increases the proportion of pooled trips by about 6% (see “No VCC” and “VCC:PU–DO”). These synergies underscore the benefits of vehicle-customer coordination on longer and pooled trips: by saving time early on, vehicle-customer coordination creates opportunities for the vehicle to serve additional customers within their deadlines.

Benefits of vehicle-customer coordination: win-win-win outcomes. Vehicle-customer coordination increases profits both by increasing revenue (especially with few vehicles) and reducing costs (especially with a large fleet). In other words, the operator saves costs while serving more customers within their deadlines, leading to even stronger reductions in the unit cost per request. In addition, the cost savings are accompanied by a reduction in vehicle miles traveled and, again, even stronger reductions in vehicle miles traveled per customer request. As such, vehicle-customer coordination can contribute to mitigating the environmental footprint of urban transportation.

Underlying these results lies a trade-off between profit improvements and customer service: vehicle-customer coordination enables to serve more customers, as opposed to merely saving costs, yet some customers have to walk before pickup or after dropoff. One possibility for the operator is to share the profit increase through lower fares—as is the case in practice. We can consider three discount schemes: (a) a uniform discount to all customers, (b) a targeted discount to customers who do walk before pickup or after dropoff, and (c) a discount prorated with walking distance. Table 4 shows that the operator can implement significant discounts, while breaking even (e.g., a uniform discount of 5.7%, a targeted discount of 16.0%, or a prorated discount of \$2.0 per 100 meters). By

sharing the benefits equally between the operator and customers, we obtain a solution such that (i) the operator’s profit increases by 3.1% (2.7% more revenue and 7.7% lower costs), (ii) 6.1% extra customers receive a service, (iii) 22% of customers walk to the pickup locations and 7% walk from the dropoff locations, with an average distance of 100 meters and a targeted discount of 8.0%, and (iv) vehicle miles traveled are reduced by 15.2%, corresponding to a 21% reduction in vehicle miles traveled per request. This solution represents a win-win-win outcome: higher operating profits, better level of service, and smaller environmental footprint.

7. Conclusion

Fueled by digitization, several modern transportation and logistics systems rely on vehicle-customer coordination: rather than vehicles visiting customers in fixed locations, vehicles and customers can arrange to meet in mutually convenient locations. To take full advantage of such coordination opportunities, however, operators require dedicated routing solutions. To tackle this problem, this paper formulated the Dial-A-Ride problem with Vehicle-Customer Coordination (DAR–VCC) via mixed-integer second-order cone optimization (MISOCO) in the Euclidean ℓ_2 space or mixed-integer linear optimization (MILO) in the Manhattan ℓ_1 space or in a discretized space. Either way, off-the-shelf implementation can only solve small-scale instances.

In response, this paper proposed a novel decomposition algorithm, combining: (i) a geometric routine to optimize the time and location of a single stop in the ℓ_2 space and the ℓ_1 space; (ii) a tailored coordinate descent scheme to optimize multiple stops in a given sequence; and (iii) a new subpath-based time-space optimization for dial-a-ride optimization. Our main theoretical result is that the coordinate descent scheme guarantees feasibility and global optimality for the multi-stop optimization—a constrained and non-separable problem. Our main computational results show that our algorithm provides strong Pareto improvements over MISOCO and MILO benchmarks: far superior solutions in much shorter computational times.

Armed with this algorithm, we optimized online routing operations with vehicle-customer coordination, via batching and optimization. Using Manhattan data, we devised an experimental setup that replicates ride-sharing operations where customers may need to walk before pickup or after dropoff, in a discretized space capturing such real-world features as one-way streets, traffic congestion, stops at intersections, etc. Our algorithm consistently terminates in seconds for the full Manhattan taxi system, enabling implementation in very large-scale operations.

From a practical standpoint, vehicle-customer coordination provides operating flexibility to improve first- and last-mile transportation. However, the “low-hanging fruits” may be limited: vehicle-customer coordination may yield small efficiency improvements when applied to merely adjust stopping locations at the “downstream” level. Instead, most of its benefits stem from comprehensively re-optimizing “upstream” operations—which customers to serve, with which vehicles, and

in which sequence. When combined with dedicated routing algorithms (such as the ones developed in this paper), vehicle-customer coordination can lead to significant performance improvements, with an average profit increase estimated at 6% (a gain of \$100,000 daily, or \$35–40M annually for a system of the size of Manhattan). In fact, we uncover that vehicle-customer coordination can bring economic benefits for mobility operators, higher level of service for customers, and a smaller environmental footprint of urban transportation—a win-win-win outcome.

References

- Agatz N, Erera AL, Savelsbergh MW, Wang X (2011) Dynamic ride-sharing: A simulation study in metro Atlanta. *Procedia-Social and Behavioral Sciences* 17:532–550.
- Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* 114(3):462–467.
- Baldacci R, Mingozi A, Roberti R (2012) New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing* 24(3):356–371.
- Balseiro SR, Brown DB, Chen C (2021) Dynamic pricing of relocating resources in large networks. *Management Science* 67(7):4075–4094.
- Bertsimas D, Jaillet P, Martin S (2019) Online vehicle routing: The edge of optimization in large-scale applications. *Operations Research* 67(1):143–162.
- Besbes O, Castro F, Lobel I (2021) Surge pricing and its spatial supply response. *Management Science* 67(3):1350–1367.
- Bimpikis K, Candogan O, Saban D (2019) Spatial pricing in ride-sharing networks. *Operations Research* 67(3):744–769.
- Boland N, Hewitt M, Marshall L, Savelsbergh M (2017) The continuous-time service network design problem. *Operations Research* 65(5):1303–1321.
- Boland N, Savelsbergh M (2019) Perspectives on integer programming for time-dependent models. *TOP* 27(2):147–173.
- Braverman A, Dai JG, Liu X, Ying L (2019) Empty-car routing in ridesharing systems. *Operations Research* 67(5):1437–1452.
- Bubeck S, Lee YT, Li Y, Sellke M (2019) Competitively chasing convex bodies. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 861–868.
- Carlsson JG, Song S (2017) Coordinated logistics with a truck and a drone. *Management Science* 64(9):4052–4069.
- Chardaire P, McKeown GP, Verity-Harrison S, Richardson S (2005) Solving a time-space network formulation for the convoy movement problem. *Operations Research* 53(2):219–230.

- Cordeau JF (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54(3):573–586.
- Dabia S, Ropke S, van Woensel T, De Kok T (2013) Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science* 47(3):380–396.
- Dash S, Günlük O, Lodi A, Tramontani A (2012) A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal on Computing* 24(1):132–147.
- Desaulniers G (2010) Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research* 58(1):179–192.
- Friedman J, Linial N (1993) On convex body chasing. *Discrete & Computational Geometry* 9(3):293–321.
- 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.
- Holland C, Levis J, Nuggehalli R, Santilli B, Winters J (2017) UPS optimizes delivery routes. *Interfaces* 47(1):8–23.
- Hu B, Hu M, Zhu H (2022) Surge pricing and two-sided temporal responses in ride hailing. *Manufacturing & Service Operations Management* 24(1):91–109.
- Huang Y, Sun D, Tang J (2018) Taxi driver speeding: who, when, where and how? A comparative study between Shanghai and New York City. *Traffic Injury Prevention* 19(3):311–316.
- Lyft (2021) Introducing shared saver, our most affordable ride. Available at: <https://blog.lyft.com/posts/introducing-shared-saver>.
- NYC Taxi & Limousine Commission (2021) TLC Trip Record Data. Available at: <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.
- OpenStreetMap (2021) Planet OSM retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>.
- Ozbaygin G, Karasan OE, Savelsbergh M, Yaman H (2017) A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological* 100:115–137.
- Özkan E, Ward AR (2020) Dynamic matching for real-time ride sharing. *Stochastic Systems* 10(1):29–70.
- Poikonen S, Golden B (2020) The mothership and drone routing problem. *INFORMS Journal on Computing* 32(2):249–262.
- Psaraftis HN (1980) A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science* 14(2):130–154.
- Santi P, Resta G, Szell M, Sobolevsky S, Strogatz SH, Ratti C (2014) Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences* 111(37):13290–13294.

- Savelsbergh MW, Sol M (1995) The general pickup and delivery problem. *Transportation Science* 29(1):17–29.
- Stieber A, Fügenschuh A, Epp M, Knapp M, Rothe H (2015) The multiple traveling salesmen problem with moving targets. *Optimization Letters* 9(8):1569–1583.
- Uber (2021) Express Pool: Walk a little, save a lot with Uber’s most affordable option. Available at: <https://www.uber.com/us/en/ride/express-pool/>.
- Vazifeh MM, Santi P, Resta G, Strogatz SH, Ratti C (2018) Addressing the minimum fleet problem in on-demand urban mobility. *Nature* 557(7706):534–538.
- Xia J, Wang K, Wang S (2019) Drone scheduling to monitor vessels in emission control areas. *Transportation Research Part B: Methodological* 119:174–196.
- Yan C, Zhu H, Korolko N, Woodard D (2020) Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics* 67(8):705–724.

Appendix A: Vehicle Routing Problem with Vehicle-Customer Coordination

We consider the VRP–VCC formulation from Gambella et al. (2018). It is a special case of the DAR–VCC in the two-dimensional Euclidean space where: (i) all customers have the same destination as the vehicle ($G_{j+n} = D$); (ii) customers have no deadlines ($\bar{T}_j = \infty$); (iii) walking distance is unrestricted ($W_j = \infty$); and (iv) the operator has to visit all customers (akin to setting a large fare g_j), so the objective is to minimize travel times. The VRP–VCC can be formulated via mixed-integer second-order cone optimization (MISOCO), which Gambella et al. (2018) solve via branch-and-price (B&P). Instead, we develop a dynamic programming (DP) algorithm to optimize vehicle-customer assignments and the sequence of customers for each vehicle, which embeds our exact MSO–VCC algorithm to optimize the time and location of each visit. To solve large-scale instances, we also design an acceleration heuristic, referred to as “DP+H” (see EC.3).

Table 5 compares our algorithms to two MISOCO benchmarks from Gambella et al. (2018): direct CPLEX implementation and B&P. We use the same problem instances, referred to as “p_n_m_Q”, where n , m and Q denote the number of requests, the number of vehicles, and the capacity of each vehicle, respectively. We use a Xeon E5-2680 processor, which has similar performance as the one used by Gambella et al. (2018).

Table 5 Computational results comparing our algorithm to Gambella et al. (2018).

Instance	CPLEX (Gambella et al. 2018)				B&P (Gambella et al. 2018)				Our algorithm				
	Sol.	LB	G_C^O	CPU(s)	Sol.	LB	G_{BP}^O	CPU(s)	Sol.	CPU(s)	CPU(%)	G_C^S	G_{BP}^S
p_10.3_6	72.69	72.69	0%	230	72.69	72.69	0%	184	72.69	56	-70%	0%	0%
p_10.4_5	62.14	62.14	0%	449	62.14	62.14	0%	70	62.14	9	-87%	0%	0%
p_10.5_4	71.71	71.71	0%	543	71.71	71.71	0%	161	71.71	1	-99%	0%	0%
p_12.3_6	76.91	76.91	0%	580	76.91	76.91	0%	377	76.91	189	-50%	0%	0%
p_12.4_5	91.36	84.05	8%	>7,200	91.36	91.36	0%	1,209	91.36	35	-97%	0%	0%
p_12.5_5	80.08	80.08	0%	4,107	80.08	80.08	0%	532	80.08	48	-91%	0%	0%
p_14.3_7	84.47	84.47	0%	6,961	84.47	84.47	0%	3,842	84.47	3,328	-13%	0%	0%
p_14.4_6	103.97	90.45	13%	>7,200	97.62	97.62	0%	5,131	97.62	1817	-65%	-6%	0%
p_14.5_5	80.18	73.77	8%	>7,200	80.18	80.18	0%	377	80.18	304	-19%	0%	0%
p_16.3_8	85.84	78.97	8%	>7,200	181.58	78.08	57%	>7,200	87.36	1436	-	+2%	-52%
p_16.4_6	76.71	75.18	2%	>7,200	75.85	75.85	0%	6,635	75.85	5165	-22%	-1%	0%
p_16.5_6	96.70	64.79	33%	>7,200	91.68	91.68	0%	6,084	91.68	5713	-6%	-5%	0%
p_18.3_8	132.58	86.18	35%	>7,200	217.47	97.86	55%	>7,200	115.15	5,500	-	-13%	-47%
p_18.4_7	87.64	60.47	31%	>7,200	80.68	75.03	7%	>7,200	80.54	4,321	-	-8%	0%
p_18.5_6	78.73	59.05	25%	>7,200	76.18	76.18	0%	1,132	76.18	3,074	+171%	-3%	0%
p_20.3_9	104.74	67.03	36%	>7,200	193.88	50.41	74%	>7,200	88.83	6,492	-	-15%	-54%
p_20.4_7	127.02	83.83	34%	>7,200	216.63	106.15	51%	>7,200	110.08	1,345	-	-13%	-49%
p_20.5_6	133.24	97.27	27%	>7,200	128.35	114.23	11%	>7,200	127.80	4,041	-	-4%	0%

“>7,200” means that the instance cannot be solved to optimality within two hours.

CPU(%) = (CPU of our algorithm – CPU of B&P)/CPU of B&P.

G_C^O = (CPLEX solution – Lower bound found by CPLEX)/CPLEX solution.

G_{BP}^O = (B&P solution – Lower bound found by B&P)/B&P solution.

G_C^S = (Solution found by our algorithm – CPLEX solution)/CPLEX solution.

G_{BP}^S = (Solution found by our algorithm – B&P solution)/B&P solution.

Our algorithm: DP in the 12 instances solved optimally by B&P; DP+H in the other 6 instances.

Note that our algorithm dominates both benchmarks. Among the 12 instances solved optimally by B&P, our DP algorithm reaches the optimum much faster. For instance, we derive the optimal solution in less than a minute in 5 instances, for which the corresponding B&P runtimes span 1–20 minutes. In 6 of the 7 other instances, our algorithm improves computational times by 6% to 65%; it is slower than B&P in only one instance. For the 6 more computationally challenging instances (which are not solved optimally by B&P), our DP+H algorithm significantly outperforms CPLEX and B&P. In 5 of the 6 instances, our algorithm

provides the best overall solution, outperforming the CPLEX (resp. B&P) solution by 4–15% (resp. 54%). To put things into perspective, B&P improves the CPLEX solution in only 2 of these 6 instances, and can otherwise deteriorate the solution by up to 85%. Equally important, in terms of computational times, our algorithm terminates in 24–108 minutes, whereas CPLEX and B&P do not terminate within 2 hours.

Let us turn to the impact of vehicle-customer coordination in the VRP–VCC. We define two benchmarks:

- “No VCC”: we optimize VRP operations in the absence of vehicle-customer coordination.
- “Seq.”: we optimize vehicle paths without vehicle-customer coordination; fix the sequence of customers for each vehicle; and re-optimize VRP–VCC operations at the “downstream” level.

In comparison, our algorithm leverages vehicle-customer coordination to also re-optimize “upstream” operations: vehicle-customer assignments and the sequence of customer visits.

Figure 10 shows that the VRP–VCC can reduce costs by up to 16% versus the baseline without vehicle-customer coordination. As compared to the sequential benchmark, it also results in significant benefits, by up to 6%. These results confirm the benefit of re-optimizing operations with vehicle-customer coordination. In addition, increasing the maximum walking distance generally yields marginally decreasing gains, with again some exceptions due to the interplay of downstream cost savings and upstream re-optimization. This, again, confirms that limited flexibility from vehicle-customer coordination can provide strong benefits.

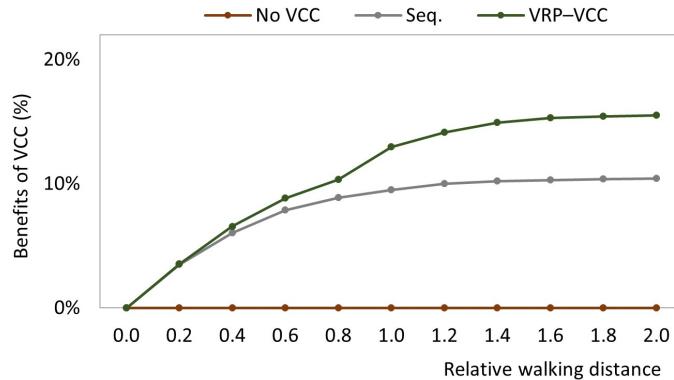


Figure 10 Cost comparison as a function of the relative walking distance (Instance p_14_5_5).

In summary, our VRP–VCC results confirm three insights from DAR–VCC: (i) the superior performance of our algorithm as compared to state-of-the-art MISOCO benchmarks; (ii) the benefits of upstream re-optimization to take full advantage of vehicle-customer coordination; and (iii) the value of even limited flexibility from vehicle-customer coordination.

Routing Optimization with Vehicle-Customer Coordination

Additional details and proof of statements

Appendix EC.1: Proofs of statements on SSO–VCC (Section 3)

EC.1.1. Proofs of statements in Section 3.1

Proof of Lemma 1

PROOF. SSO–VCC minimizes the time at which the vehicle reaches its destination, while ensuring that the vehicle and the customer reach the stopping location by time v_M and that the customer does not walk more than W . It is formulated in Equations (EC.1)–(EC.5).

$$[\text{SSO–VCC}] \quad \min \left(v_M + \frac{\|D - M\|}{\bar{S}} \right), \quad (\text{EC.1})$$

$$\text{s.t. } v_M \geq \bar{v} + \frac{\|M - O\|}{\bar{S}}, \quad (\text{EC.2})$$

$$v_M \geq \frac{\|M - H\|}{S}, \quad (\text{EC.3})$$

$$\|M - H\| \leq W, \quad (\text{EC.4})$$

$$v_M \geq 0, \quad M \in \mathbb{R}^2. \quad (\text{EC.5})$$

Suppose by contradiction that an optimal solution of SSO–VCC (M, v_M) satisfies $v_M > \bar{v} + \frac{\|M - O\|}{\bar{S}}$. First, note that, if $v_M > \frac{\|M - H\|}{S}$, we can construct another solution (M', v'_M) such that:

$$M' = M, \quad \text{and} \quad v'_M = \max \left\{ \bar{v} + \frac{\|M - O\|}{\bar{S}}, \frac{\|M - H\|}{S} \right\}. \quad (\text{EC.6})$$

This solution is clearly feasible and $v'_M < v_M$. This contradicts the optimality of (M, v_M) .

Now, if $v_M = \frac{\|M - H\|}{S}$, the vehicle reaches the stopping location M at time $\bar{v} + \frac{\|M - O\|}{\bar{S}}$, and then waits for the customer. Let \tilde{M} be the location of the customer at that time. We have

$$v_M = \bar{v} + \frac{\|M - O\|}{\bar{S}} + \frac{\|\tilde{M} - M\|}{S}. \quad (\text{EC.7})$$

Instead, if the vehicle travels from M to \tilde{M} to meet the customer in-between, the vehicle and the customer will meet after a time $\frac{\|\tilde{M} - M\|}{\bar{S} + S}$. After that, the vehicle travels back to M (with the customer), which takes an extra $\frac{\|\tilde{M} - M\|}{\bar{S} + S}$. The vehicle thus reaches location M at time

$$v'_M = \bar{v} + \frac{\|M - O\|}{\bar{S}} + \frac{2\|\tilde{M} - M\|}{\bar{S} + S} < \bar{v} + \frac{\|M - O\|}{\bar{S}} + \frac{\|\tilde{M} - M\|}{S} = v_M, \quad \text{because } \bar{S} > S. \quad (\text{EC.8})$$

This, again, contradicts the optimality of (M, v_M) . This completes the proof by contradiction. \square

Proof of Lemma 2

PROOF. From the triangle inequality, $\|M - O\| + \|D - M\| \geq \|D - O\|$ for all $M \in \mathbb{R}^2$. Moreover $\|M - O\| + \|D - M\| = \|D - O\|$ if and only if M is on the $[OD]$ segment. \square

Proof of Lemma 3

PROOF. Suppose by contradiction that the optimal solution M^* satisfies $\frac{\|M^*-H\|}{S} < \bar{v} + \frac{\|M^*-O\|}{\bar{S}}$ and $\|M-H\| < W$. From Lemma 1, the objective function becomes $v^* = \bar{v} + \frac{\|M^*-O\|}{S} + \frac{\|D-M^*\|}{\bar{S}}$.

Let us define point M_1 as a feasible stopping point on the straight path between O and M^* such that $\|M_1-O\| = \|M^*-O\| - \varepsilon$. For $\varepsilon > 0$ sufficiently small, the solution M_1 is feasible. We have:

$$\begin{aligned} v^* &= \bar{v} + \frac{\|M^*-O\|}{S} + \frac{\|D-M^*\|}{\bar{S}} \\ &= \bar{v} + \frac{\|M_1-O\|}{S} + \frac{\|M^*-M_1\|}{\bar{S}} + \frac{\|D-M^*\|}{\bar{S}} \\ &> \bar{v} + \frac{\|M_1-O\|}{S} + \frac{\|D-M_1\|}{\bar{S}}, \end{aligned} \quad (\text{EC.9})$$

where the last inequality stems from the triangle inequality and the fact that M^* is not on the straight path between O and D . This contradicts the optimality of M^* . \square

EC.1.2. Proofs of statements in Section 3.2

Proof of Lemma 4

PROOF. Consider the points of coordinates $M = (\chi, \gamma)$, $O = (0, 0)$ and $H = (d, 0)$. We have:

$$\begin{aligned} \frac{\|M-H\|_2}{S} \leq \frac{\|M-O\|_2}{S} &\iff \frac{(\chi-d)^2 + (\gamma-0)^2}{S^2} \leq \frac{(\chi-0)^2 + (\gamma-0)^2}{\bar{S}^2} \\ &\iff (\bar{S}^2 - S^2)\chi^2 + (\bar{S}^2 - S^2)\gamma^2 - 2d\bar{S}^2\chi \leq -d^2\bar{S}^2 \\ &\iff \chi^2 + \gamma^2 - 2d\frac{\bar{S}^2}{\bar{S}^2 - S^2}\chi \leq -d^2\frac{\bar{S}^2}{\bar{S}^2 - S^2} \\ &\iff \left(\chi - d\frac{\bar{S}^2}{\bar{S}^2 - S^2}\right)^2 + \gamma^2 \leq \left(d\frac{\bar{S}S}{\bar{S}^2 - S^2}\right)^2 \\ \|M-H\|_2 \leq W &\iff (\chi-d)^2 + \gamma^2 \leq W^2. \quad \square \end{aligned}$$

Preliminaries on SSO–VCC with the Euclidean distance

Whenever a straight-path solution does not exist, the optimal stopping location lies on an arc of circle \mathcal{C}^V or an arc of circle \mathcal{C}^C (Lemma 3). The SSO–VCC solution is shown in Figure EC.1. The different cases will be used in the proof of Proposition 1. Before proceeding, we first introduce some notations and use our geometric insights to restrict the search space in Lemma EC.1.

LEMMA EC.1. *Let us define M^+ and M^- (resp. N^+ and N^-) as the points on the lines starting from $O = (0, 0)$ and tangent to circle \mathcal{C}^V (resp. \mathcal{C}^C); and I^+ and I^- as the points at the intersection of \mathcal{C}^C and \mathcal{C}^V , when $\mathcal{C}^C \cap \mathcal{C}^V \neq \emptyset$ (Figure EC.1). We have:*

$$M^+(X_M, Y_M), M^-(X_M, -Y_M), \quad \text{with: } X_M = d \quad \text{and} \quad Y_M = d \frac{S}{\sqrt{\bar{S}^2 - S^2}},$$

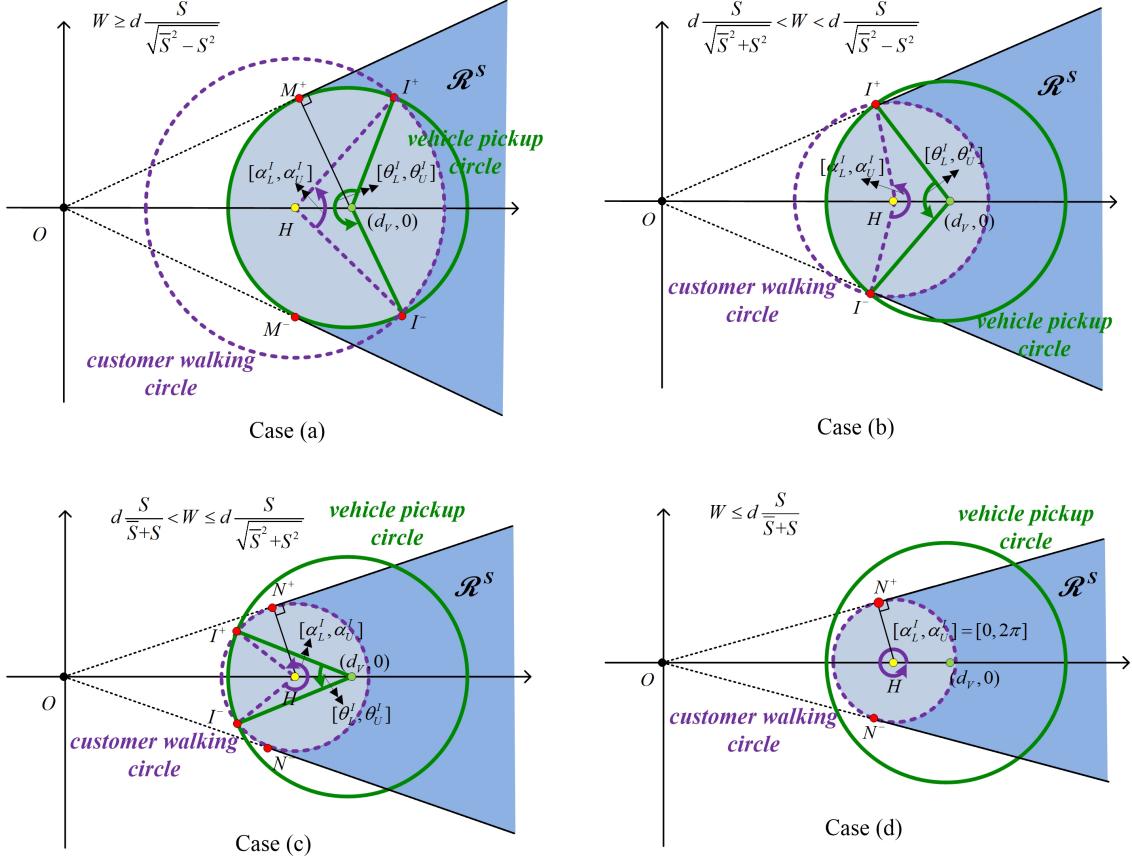


Figure EC.1 Structure of the SSO-VCC solution with the Euclidean distance.

$$\begin{aligned}
 & N^+(X_N, Y_N), N^-(X_N, -Y_N), \quad \text{with: } X_N = d - \frac{W^2}{d} \quad \text{and} \quad Y_N = W \sqrt{1 - \frac{W^2}{d^2}}, \\
 & I^+(X_I, Y_I), I^-(X_I, -Y_I), \quad \text{with: } X_I = \frac{W^2 - r_V^2 + d_V^2 - d^2}{2(d_V - d)} \quad \text{and} \quad Y_I = \sqrt{W^2 - \left(\frac{W^2 - r_V^2 + (d_V - d)^2}{2(d_V - d)} \right)^2}.
 \end{aligned}$$

Let us define the left-hand arcs of the vehicle pickup circle and the customer walking circle:

$$\begin{aligned}
 \widehat{M^- M^+} &= \{(\chi, \gamma) \in \mathbb{R}^2 : (\chi - d_V)^2 + \gamma^2 = r_V^2 \text{ and } \chi \leq d\}, \\
 \widehat{N^- N^+} &= \left\{ (\chi, \gamma) \in \mathbb{R}^2 : (\chi - d)^2 + \gamma^2 = W^2 \text{ and } \chi \leq d - \frac{W^2}{d} \right\}.
 \end{aligned}$$

Let $\mathcal{A}^V = \mathcal{C}^V \cap \mathcal{D}^C$ denote the arc on the vehicle pickup circle lying within the customer walking disk. Similarly, let $\mathcal{A}^C = \mathcal{C}^C \cap \mathcal{D}^V$ denote the arc on the customer walking circle lying within the vehicle pickup disk. If a straight-path solution is not feasible, the optimal solution M^* lies on

$$\underbrace{\left(\widehat{M^- M^+} \cap \mathcal{A}^V \right)}_{\text{arc of circle } \mathcal{C}^V} \cup \underbrace{\left(\widehat{N^- N^+} \cap \mathcal{A}^C \right)}_{\text{arc of circle } \mathcal{C}^C}.$$

Moreover, there exist $\theta_L^M \in [\frac{\pi}{2}, \pi]$, $\alpha_L^N \in [\frac{\pi}{2}, \pi]$, $\theta_L^I \in [0, \pi]$, $\alpha_L^I \in [-\pi, 0]$, $\theta_U^M \in [\pi, \frac{3\pi}{2}]$, $\alpha_U^N \in [\pi, \frac{3\pi}{2}]$, $\theta_U^I \in [\pi, 2\pi]$, $\alpha_U^I \in [0, \pi]$ such that:

$$\Delta(OHD) \cap \widehat{M^- M^+} = \{(\chi, \gamma) \mid \exists \theta \in [\theta_L^M, \theta_U^M], \chi = d_V + r_V \cos(\theta) \text{ and } \gamma = r_V \sin(\theta)\},$$

$$\Delta(OHD) \cap \mathcal{A}^V = \{(\chi, \gamma) \mid \exists \theta \in [\theta_L^I, \theta_U^I], \chi = d_V + r_V \cos(\theta) \text{ and } \gamma = r_V \sin(\theta)\},$$

$$\Delta(OHD) \cap \widehat{N^- N^+} = \{(\chi, \gamma) \mid \exists \alpha \in [\alpha_L^N, \alpha_U^N], \chi = d + W \cos(\alpha) \text{ and } \gamma = W \sin(\alpha)\},$$

$$\Delta(OHD) \cap \mathcal{A}^C = \{(\chi, \gamma) \mid \exists \alpha \in [\alpha_L^I, \alpha_U^I], \chi = d + W \cos(\alpha) \text{ and } \gamma = W \sin(\alpha)\}.$$

PROOF OF LEMMA EC.1.

Coordinates of M^+ and M^- . Let (X_M, Y_M) be the coordinates of M^+ or M^- . We have:

$$(X_M - d_V)^2 + Y_M^2 = r_V^2. \quad (\text{EC.10})$$

We know that (OM^+) (resp. (OM^-)) is orthogonal to $(O^V M^+)$ (resp $(O^V M^-)$), where $O^V(d_V, 0)$ is the center of \mathcal{C}^V . We obtain, from the Pythagorean theorem:

$$(Y_M^2 + X_M^2) + r_V^2 = d_V^2. \quad (\text{EC.11})$$

We obtain from Equations (EC.10) and (EC.11):

$$X_M = d_V - \frac{r_V^2}{d_V} \quad \text{and} \quad Y_M^2 = r_V^2 - \frac{r_V^4}{d_V^2}. \quad (\text{EC.12})$$

By substituting $d_V = d \frac{\bar{S}^2}{\bar{S}^2 - S^2}$ and $r_V = d \frac{\bar{S}S}{\bar{S}^2 - S^2}$, we obtain

$$X_M = d \quad \text{and} \quad Y_M = \pm d \frac{S}{\sqrt{\bar{S}^2 - S^2}}. \quad (\text{EC.13})$$

Coordinates of N^+ and N^- . Let (X_N, Y_N) be the coordinates of N^+ or N^- . We have:

$$(X_N - d)^2 + Y_N^2 = W^2, \quad (\text{EC.14})$$

$$(X_N^2 + Y_N^2) + W^2 = d^2. \quad (\text{EC.15})$$

By proceeding similarly:

$$X_N = d - \frac{W^2}{d}, \quad Y_N = \pm W \sqrt{1 - \frac{W^2}{d^2}}. \quad (\text{EC.16})$$

Coordinates of I^+ and I^- . Let (X_I, Y_I) be the coordinates of I^+ or I^- . We have:

$$(X_I - d)^2 + Y_I^2 = W^2, \quad (\text{EC.17})$$

$$(X_I - d_V)^2 + Y_I^2 = r_V^2. \quad (\text{EC.18})$$

From (EC.17) and (EC.18), we have:

$$-(d_V - d)^2 - 2(d_V - d)(d - X_I) = W^2 - r_V^2, \quad (\text{EC.19})$$

hence:

$$X_I = \frac{W^2 - r_V^2 + d_V^2 - d^2}{2(d_V - d)} \quad \text{and} \quad Y_I = \pm \sqrt{W^2 - \left(\frac{W^2 - r_V^2 + (d_V - d)^2}{2(d_V - d)} \right)^2}. \quad (\text{EC.20})$$

Proof that the optimal solution is on $\widehat{M^-M^+}$ or $\widehat{N^-N^+}$.

From Lemma 3, the optimal solution M^* is on \mathcal{C}^V or \mathcal{C}^C . Suppose, by contradiction, that M^* lies on arc $\{(\chi, \gamma) \in \mathbb{R}^2 : (\chi - d_V)^2 + \gamma^2 = r_V^2 \text{ and } \chi > d\}$. There exists a feasible stopping location M_1 at the intersection of arc $\widehat{M^-M^+}$ and segment $[OM^*]$. We have:

$$\begin{aligned} v_2 &= \frac{\|M^* - O\|_2}{\bar{S}} + \frac{\|D - M^*\|_2}{\bar{S}} \\ &= \frac{\|M_1 - O\|_2}{\bar{S}} + \frac{\|M^* - M_1\|_2}{\bar{S}} + \frac{\|D - M^*\|_2}{\bar{S}} \\ &> \frac{\|M_1 - O\|_2}{\bar{S}} + \frac{\|D - M_1\|_2}{\bar{S}}, \end{aligned} \quad (\text{EC.21})$$

where the last inequality stems from the triangle inequality and the fact that M^* is not on the $[OD]$ segment. This contradicts the optimality of M^* .

Similarly, the optimal solution M^* cannot lie on arc $\{(\chi, \gamma) \in \mathbb{R}^2 : (\chi - d)^2 + \gamma^2 = W^2 \text{ and } \chi > d - \frac{W^2}{d}\}$.

This shows that, if a straight-path solution is not feasible, the solution lies on

$$\underbrace{\left(\widehat{M^-M^+} \cap \mathcal{A}^V \right)}_{\text{arc of disk } \mathcal{D}^V} \cup \underbrace{\left(\widehat{N^-N^+} \cap \mathcal{A}^C \right)}_{\text{arc of disk } \mathcal{D}^C}.$$

Polar coordinates.

We set $\theta_L^M \in [\frac{\pi}{2}, \pi]$, $\alpha_L^N \in [\frac{\pi}{2}, \pi]$, $\theta_L^I \in [0, \pi]$, $\alpha_L^I \in [-\pi, 0]$, $\theta_U^M \in [\pi, \frac{3\pi}{2}]$, $\alpha_U^N \in [\pi, \frac{3\pi}{2}]$, $\theta_U^I \in [\pi, 2\pi]$, and $\alpha_U^I \in [0, \pi]$ to define the arcs in polar coordinates:

$$\begin{aligned} \widehat{M^-M^+} &= \{(\chi, \gamma) \mid \exists \theta \in [\theta_L^M, \theta_U^M], \chi = d_V + r_V \cos(\theta) \text{ and } \gamma = r_V \sin(\theta)\}, \\ \mathcal{A}^V &= \{(\chi, \gamma) \mid \exists \theta \in [\theta_L^I, \theta_U^I], \chi = d_V + r_V \cos(\theta) \text{ and } \gamma = r_V \sin(\theta)\}, \\ \widehat{N^-N^+} &= \{(\chi, \gamma) \mid \exists \alpha \in [\alpha_L^N, \alpha_U^N], \chi = d + W \cos(\alpha) \text{ and } \gamma = W \sin(\alpha)\}, \\ \mathcal{A}^C &= \{(\chi, \gamma) \mid \exists \alpha \in [\alpha_L^I, \alpha_U^I], \chi = d + W \cos(\alpha) \text{ and } \gamma = W \sin(\alpha)\}. \end{aligned}$$

Note that θ defines a point on circle \mathcal{C}^V , hence arcs $\widehat{M^-M^+} \subset \mathcal{C}^V$ and $\mathcal{A}^V \subseteq \mathcal{C}^V$. In contrast, α defines a point on circle \mathcal{C}^C , hence arcs $\widehat{N^-N^+} \subset \mathcal{C}^C$ and $\mathcal{A}^C \subseteq \mathcal{C}^C$.

The angles $\theta_L^M \in [\frac{\pi}{2}, \pi]$, $\alpha_L^N \in [\frac{\pi}{2}, \pi]$, $\theta_U^M \in [\pi, \frac{3\pi}{2}]$, and $\alpha_U^N \in [\pi, \frac{3\pi}{2}]$ are defined as follows:

$$\theta_L^M = \pi - \tan^{-1} \left(\frac{Y_M}{d_V - X_M} \right) \text{ and } \theta_U^M = \pi + \tan^{-1} \left(\frac{Y_M}{d_V - X_M} \right), \quad (\text{EC.22})$$

$$\alpha_L^N = \pi - \tan^{-1} \left(\frac{Y_N}{d - X_N} \right) \text{ and } \alpha_U^N = \pi + \tan^{-1} \left(\frac{Y_N}{d - X_N} \right). \quad (\text{EC.23})$$

In contrast, the angles $\theta_L^I \in [0, \pi]$, $\alpha_L^I \in [-\pi, 0]$, $\theta_U^I \in [\pi, 2\pi]$, and $\alpha_U^I \in [0, \pi]$, which are associated with the intersection points I^+ and I^- , depend on the parameter values. This is illustrated in Figure EC.1, where arc \mathcal{A}^V is shown in green and arc \mathcal{A}^C is shown in purple.

- If $\mathcal{C}^C \cap \mathcal{C}^V = \emptyset$ and $\mathcal{D}^C \cap \mathcal{D}^V = \mathcal{D}^V$ (Case (a)), then $[\theta_L^I, \theta_U^I] = [0, 2\pi]$ and $[\alpha_L^I, \alpha_U^I] = \emptyset$.
- If $\mathcal{C}^C \cap \mathcal{C}^V = \emptyset$ and $\mathcal{D}^C \cap \mathcal{D}^V = \mathcal{D}^C$ (Case (e)), then $[\theta_L^I, \theta_U^I] = \emptyset$ and $[\alpha_L^I, \alpha_U^I] = [0, 2\pi]$.
- If $\mathcal{C}^C \cap \mathcal{C}^V \neq \emptyset$ (Cases (b), (c), (d)), then:

$$\text{If } X_I > d_V, \text{ then } \theta_L^I = \tan^{-1} \left(\frac{Y_I}{X_I - d_V} \right) \text{ and } \theta_U^I = 2\pi - \tan^{-1} \left(\frac{Y_I}{X_I - d_V} \right).$$

$$\text{If } X_I = d_V, \text{ then } \theta_L^I = \frac{\pi}{2} \text{ and } \theta_U^I = \frac{3\pi}{2}.$$

$$\text{If } X_I < d_V, \text{ then } \theta_L^I = \pi - \tan^{-1} \left(\frac{Y_I}{d_V - X_I} \right) \text{ and } \theta_U^I = \pi + \tan^{-1} \left(\frac{Y_I}{d_V - X_I} \right).$$

$$\text{If } X_I > d, \text{ then } \alpha_L^I = -\tan^{-1} \left(\frac{Y_I}{X_I - d} \right) \text{ and } \alpha_U^I = \tan^{-1} \left(\frac{Y_I}{X_I - d} \right).$$

$$\text{If } X_I = d, \text{ then } \alpha_L^I = -\frac{\pi}{2} \text{ and } \alpha_U^I = \frac{\pi}{2}.$$

$$\text{If } X_I < d, \text{ then } \alpha_L^I = -\pi + \tan^{-1} \left(\frac{Y_I}{d - X_I} \right) \text{ and } \alpha_U^I = \pi - \tan^{-1} \left(\frac{Y_I}{d - X_I} \right). \quad \square$$

REMARK EC.1. We can show easily that M^* lies in the triangle $\Delta(OHD)$ formed by O , H and D . From an algorithmic standpoint, we can use this result to restrict the search to the subset of $\widehat{M^-M^+}, \mathcal{A}^V, \widehat{N^-N^+}, \mathcal{A}^C$ in the triangle of $\Delta(OHD)$, and re-define all angles accordingly.

Main result: Proof of Proposition 1

PROOF. We define the following four cases, indicated in Figure EC.1:

- Case (a): $W \geq d \frac{S}{\sqrt{S^2 - S^2}}$. In this case, the region \mathcal{R}^S is defined by the two tangent lines to circle \mathcal{C}^V , with tangent points M^+ and M^- . We obtain:

$$\mathcal{R}^S = (\mathcal{D}^V \cap \mathcal{D}^C) \cup \left\{ (X, Y) \in \mathbb{R}^2 : X > X_M \text{ and } \left| \frac{Y}{X} \right| \leq \frac{Y_M}{X_M} \right\}. \quad (\text{EC.24})$$

- Case (b): $d \frac{S}{\sqrt{S^2 + S^2}} < W < d \frac{S}{\sqrt{S^2 - S^2}}$. The region \mathcal{R}^S is defined by (OI^+) and (OI^-) . We obtain:

$$\mathcal{R}^S = (\mathcal{D}^V \cap \mathcal{D}^C) \cup \left\{ (X, Y) \in \mathbb{R}^2 : X > X_I \text{ and } \left| \frac{Y}{X} \right| \leq \frac{Y_I}{X_I} \right\}. \quad (\text{EC.25})$$

- Case (c): $d \frac{S}{S+S} < W \leq d \frac{S}{\sqrt{S^2 + S^2}}$. The region \mathcal{R}^S is defined by the two tangent lines to circle \mathcal{C}^C , with tangent points N^+ and N^- . We obtain:

$$\mathcal{R}^S = (\mathcal{D}^V \cap \mathcal{D}^C) \cup \left\{ (X, Y) \in \mathbb{R}^2 : X > X_N \text{ and } \left| \frac{Y}{X} \right| \leq \frac{Y_N}{X_N} \right\}. \quad (\text{EC.26})$$

- Case (d): $W \leq d \frac{S}{S+S}$. In this case, $\mathcal{D}^V \cap \mathcal{D}^C = \mathcal{D}^C$. The region \mathcal{R}^S is still defined by the two tangent lines to circle \mathcal{C}^C , with tangent points N^+ and N^- . It is given by Equation (EC.26). In all cases, if $D \in \mathcal{R}^S$, all solutions in $\mathcal{D}^V \cap \mathcal{D}^C \cap [OD]$ are then optimal. Otherwise, we represent the stopping location M by $(d_V + W \cos(\theta), W \sin(\theta)) \in \mathcal{C}^V$ and $(d + W \cos(\alpha), W \sin(\alpha)) \in \mathcal{C}^C$. From Lemma 1, the SSO–VCC objective function is equal to $\frac{\|M-O\|_2 + \|D-M\|_2}{S} = \frac{f(\theta)}{S}$ if $M \in \mathcal{C}^V$ and to $\frac{\|M-O\|_2 + \|D-M\|_2}{S} = \frac{h(\alpha)}{S}$ if $M \in \mathcal{C}^C$, with:

$$\begin{aligned} f(\theta) &= \sqrt{(d_V + r_V \cos(\theta) - 0)^2 + (r_V \sin(\theta) - 0)^2} + \sqrt{(d_V + r_V \cos(\theta) - X)^2 + (r_V \sin(\theta) - Y)^2} \\ &= \sqrt{d_V^2 + r_V^2 \cos^2 \theta + 2d_V r_V \cos(\theta) + r_V^2 \sin^2 \theta} \\ &\quad + \sqrt{r_V^2 \cos^2 \theta + (d_V - X)^2 + 2r_V(d_V - X) \cos(\theta) + r_V^2 \sin^2 \theta + Y^2 - 2r_V Y \sin(\theta)} \\ &= \sqrt{r_V^2 + d_V^2 + 2d_V r_V \cos(\theta)} + \sqrt{r_V^2 + (d_V - X)^2 + Y^2 + 2r_V(d_V - X) \cos(\theta) - 2r_V Y \sin(\theta)}, \\ h(\alpha) &= \sqrt{(d + W \cos(\alpha) - 0)^2 + (W \sin(\alpha) - 0)^2} + \sqrt{(d + W \cos(\alpha) - X)^2 + (W \sin(\alpha) - Y)^2} \\ &= \sqrt{d^2 + W^2 \cos^2 \alpha + 2dW \cos(\alpha) + W^2 \sin^2 \alpha} \\ &\quad + \sqrt{W^2 \cos^2 \theta + (d - X)^2 + 2W(d - X) \cos(\alpha) + W^2 \sin^2 \alpha + Y^2 - 2WY \sin(\alpha)} \\ &= \sqrt{W^2 + d^2 + 2dW \cos(\alpha)} + \sqrt{W^2 + (d - X)^2 + Y^2 + 2W(d - X) \cos(\alpha) - 2WY \sin(\alpha)}. \end{aligned}$$

We obtain:

$$v^* = \frac{1}{S} \min \left\{ \min_{\theta \in [\theta_L^M, \theta_U^M] \cap [\theta_L^I, \theta_U^I]} f(\theta), \min_{\alpha \in [\alpha_L^N, \alpha_U^N] \cap [\alpha_L^I, \alpha_U^I]} h(\alpha) \right\}.$$

Finally, we need to prove that f and h are unimodal. Although we have explicit functions, it is hard to prove unimodality analytically. Instead, we leverage the geometric structure of the problem.

To simplify the exposition, we adjust our coordinate system so that $O = (\bar{c}, 0)$, $D = (-\bar{c}, 0)$, with $2\bar{c} = \|D - O\|_2$. Let us focus on “iso-objective” regions, i.e., sets of points that achieve objective value of SSO–VCC. Remark EC.2 shows that these regions define ellipses.

REMARK EC.2. Let $\mathcal{E}(a) = \{M \in \mathbb{R}^2 : \|M - O\|_2 + \|D - M\|_2 = 2a\}$ be the set of points with the same objective $\frac{2a}{S}$. With $O = (\bar{c}, 0)$, $D = (-\bar{c}, 0)$ and $M = (\tilde{x}, \tilde{y})$, $\mathcal{E}(a)$ is the ellipse $\frac{\tilde{x}^2}{a^2} + \frac{\tilde{y}^2}{a^2 - \bar{c}^2} = 1$ with center $(0, 0)$, focal distance \bar{c} , semi-major axis a , and semi-minor axis $b = \sqrt{a^2 - \bar{c}^2}$.

By construction, all the points on a given ellipse (shown in Figure EC.2) correspond to stopping locations that result in the same overall travel time from O to D . Next, note that the ellipse is governed by two parameters: the focal distance \bar{c} (a parameter, determined by the distance from O to D), and the semi-major axis a (to be determined). In other words, for any $a_1 < a_2$, all the points on the ellipse defined by $\frac{\tilde{x}^2}{a_1^2} + \frac{\tilde{y}^2}{a_1^2 - \bar{c}^2} = 1$ will result in a travel time that is shorter, by $\frac{2(a_2 - a_1)}{S}$, than all the points on the ellipse defined by $\frac{\tilde{x}^2}{a_2^2} + \frac{\tilde{y}^2}{a_2^2 - \bar{c}^2} = 1$. At the same time, any travel time is achievable only if the corresponding ellipse intersects with the vehicle pickup disk \mathcal{D}^V .

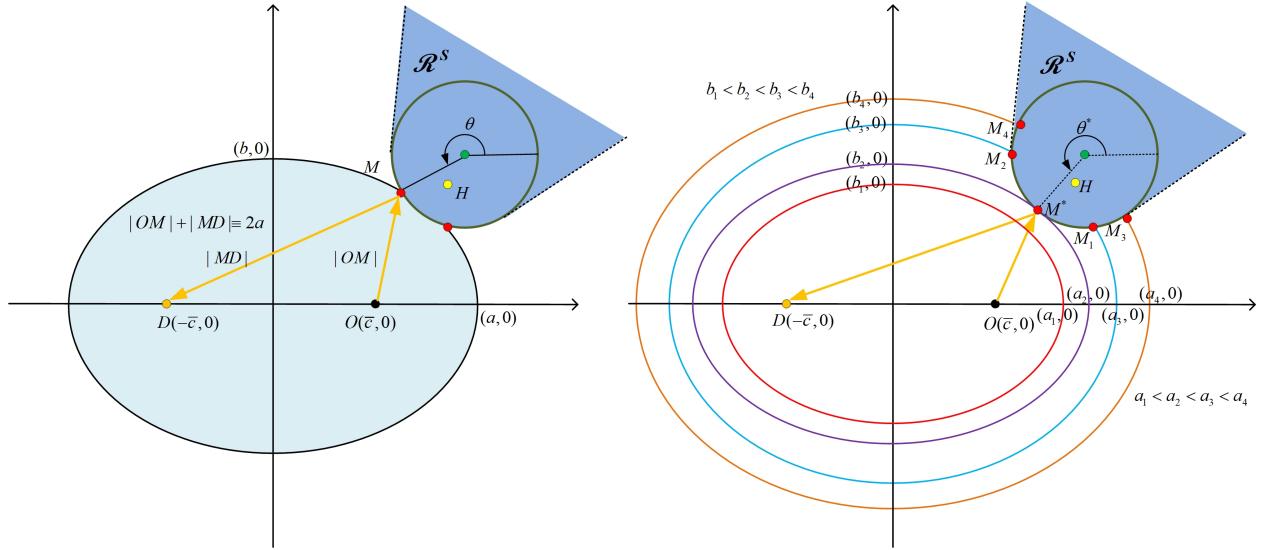


Figure EC.2 Elliptical representation of iso-objective regions.

SSO-VCC can thus be re-stated as follows: find the ellipse with the smallest possible semi-major axis that intersects with \mathcal{C}^V , that is, find the ellipse that is externally tangent to \mathcal{C}^V (shown in purple in Figure EC.2). Let M^* denote the corresponding stopping location, a^* denote the corresponding semi-major axis, and θ^* denote the corresponding angle on the disk \mathcal{D}^V . Then, M^* is of coordinates $(d_V + r_V \cos(\theta^*), r_V \sin(\theta^*))$ (in the original coordinate system) and $\|M^* - O\|_2 + \|D - M^*\|_2 = 2a^*$.

As we further increase $a > a^*$, the ellipse will intersect the vehicle pickup circle \mathcal{C}^V in two points—one corresponding to an angle $\theta^- < \theta^*$ and one corresponding to an angle $\theta^+ > \theta^*$. Moreover, θ^- decreases and θ^+ increases as a increases. We repeat the process until we reach θ_L^M and θ_U^M . Therefore, as θ decreases from θ^* to θ_L^M , the objective function $\frac{\|M - O\|_2 + \|D - M\|_2}{S}$ increases; similarly, as θ increases from θ^* to θ_U^M , the objective function $\frac{\|M - O\|_2 + \|D - M\|_2}{S}$ increases.

This proves that f is unimodal over $[\theta_L^M, \theta_U^M]$. We can proceed similarly to prove that $h(\alpha)$ is unimodal in $[\alpha_L^N, \alpha_U^N]$. This completes the proof of Proposition 1. \square

Proof of Proposition 2

PROOF. Recall that \tilde{H}_φ refers to the customer's location at time \bar{v} and that \tilde{M}_φ refers to the intersection of the ray $H\tilde{H}_\varphi$ and the boundary of $\mathcal{D}^\varphi \cap \mathcal{D}^C$. Let us define $\tilde{v}(\varphi)$ as the vehicle's arrival time in D when the customer walks in direction φ from time 0 to time \bar{v} :

$$\tilde{v}(\varphi) = \bar{v} + \min_{M \in \mathbb{R}^2} \left\{ \frac{\|M - O\|_2 + \|D - M\|_2}{S} \text{ s.t. } \frac{\|M - \tilde{H}_\varphi\|_2}{S} \leq \frac{\|M - O\|_2}{S}, \|M - H\|_2 \leq W \right\} \quad (\text{EC.27})$$

$$= \bar{v} + \min_{M \in \mathcal{D}^C \cap \mathcal{D}^\varphi} \left\{ \frac{\|M - O\|_2 + \|D - M\|_2}{S} \right\}. \quad (\text{EC.28})$$

Let us denote by $\hat{v}(\varphi)$ the vehicle's arrival time given stopping location \tilde{M}_φ . We have:

$$\hat{v}(\varphi) = \bar{v} + \frac{\|\tilde{M}_\varphi - O\|_2 + \|D - \tilde{M}_\varphi\|_2}{\bar{S}}.$$

By construction, \tilde{M}_φ is a feasible solution to (EC.27), so $\tilde{v}(\varphi) \leq \hat{v}(\varphi), \forall \varphi \in [\varphi_L, \varphi_U]$. It comes:

$$\min_{\varphi \in [\varphi_L, \varphi_U]} \{\tilde{v}(\varphi)\} \leq \min_{\varphi \in [\varphi_L, \varphi_U]} \hat{v}(\varphi). \quad (\text{EC.29})$$

Next, let us denote by M^* the optimal stopping location of SSO-VCC, so that:

$$\begin{aligned} \varphi^* &= \arg \min_{\varphi \in [\varphi_L, \varphi_U]} \{\tilde{v}(\varphi)\}, \\ M^* &= \arg \min_{M \in \mathbb{R}^2} \left\{ \frac{\|M - O\|_2 + \|D - M\|_2}{\bar{S}} \text{ s.t. } \frac{\|M - \tilde{H}_{\varphi^*}\|_2}{S} \leq \frac{\|M - O\|_2}{\bar{S}}, \|M - H\|_2 \leq W \right\}. \end{aligned}$$

Let us assume that \tilde{H}_{φ^*} is not on segment $[HM^*]$. Then, let \tilde{H}'_{φ^*} be the projection of \tilde{H}_{φ^*} on the segment $[HM^*]$. By definition of the projection, we have:

$$\frac{\|M^* - \tilde{H}'_{\varphi^*}\|_2}{S} \leq \frac{\|M^* - \tilde{H}_{\varphi^*}\|_2}{S} \leq \frac{\|M^* - O\|_2}{\bar{S}}. \quad (\text{EC.30})$$

Moreover, from the triangle inequality, we have:

$$\|M^* - H\|_2 = \|M^* - \tilde{H}'_{\varphi^*}\|_2 + \|\tilde{H}'_{\varphi^*} - H\|_2 \leq \|M^* - \tilde{H}_{\varphi^*}\|_2 + \|\tilde{H}_{\varphi^*} - H\|_2 \leq W.$$

We can thus construct a feasible solution that achieves the same objective value, while satisfying the additional equation that the customer does change directions at time \bar{v} . This proves that:

$$\min_{\varphi \in [\varphi_L, \varphi_U]} \hat{v}(\varphi) \leq \hat{v}(\varphi^*) \leq \tilde{v}(\varphi^*) = \min_{\varphi \in [\varphi_L, \varphi_U]} \{\tilde{v}(\varphi)\}. \quad (\text{EC.31})$$

Ultimately, referring to (EC.29), this proves that:

$$\min_{\varphi \in [\varphi_L, \varphi_U]} \{\tilde{v}(\varphi)\} = \min_{\varphi \in [\varphi_L, \varphi_U]} \hat{v}(\varphi). \quad (\text{EC.32})$$

This completes the proof. \square

EC.1.3. Proofs of statements in Section 3.3

Let us first illustrate the feasible region in the two-dimensional space with the Manhattan distance. Consider a coordinate system with $O = (a, b)$, $H = (c, d)$ and $D = (X, Y)$, and let $M = (\chi, \gamma)$ denote the stopping location. The constraint $\frac{\|M - H\|_1}{S} \leq \bar{v} + \frac{\|M - O\|_1}{\bar{S}}$ is equivalent to $M \in \mathcal{G}^V$ and the constraint $\|M - H\|_1 \leq W$ is equivalent to $M \in \mathcal{G}^C$, where:

$$\begin{aligned} \mathcal{G}^V &= \left\{ (\chi, \gamma) \in \mathbb{R}^2 : \frac{|\chi - c| + |\gamma - d|}{S} \leq \bar{v} + \frac{|\chi - a| + |\gamma - b|}{\bar{S}} \right\}, \\ \mathcal{G}^C &= \left\{ (\chi, \gamma) \in \mathbb{R}^2 : \frac{|\chi - c| + |\gamma - d|}{S} \leq W \right\}. \end{aligned}$$

Proof of Proposition 3

PROOF. For any stopping location M , we show that the distance from O to M and from M to D is equal to the shortest distance from O to D plus twice the distance L from M to $O \square D$:

$$\|O - M\|_1 + \|M - D\|_1 = \|O - D\|_1 + 2L. \quad (\text{EC.33})$$

Let us define the coordinates $M = (\chi, \gamma)$, $O = (a, b)$ and $D = (X, Y)$. We distinguish the following cases (shown in Figure EC.3). In each case, at most one of the inequalities is strict:

- (i) $\chi \leq \min\{a, X\}$, $\gamma \leq \min\{b, Y\}$;
- (ii) $\chi \geq \max\{a, X\}$, $\gamma > \max\{b, Y\}$;
- (iii) $\chi \leq \min\{a, X\}$, $\gamma \geq \max\{b, Y\}$;
- (iv) $\chi \geq \max\{a, X\}$, $\gamma \leq \min\{b, Y\}$;
- (v) $\min\{a, X\} \leq \chi \leq \max\{a, X\}$, $\gamma \leq \min\{b, Y\}$;
- (vi) $\min\{a, X\} \leq \chi \leq \max\{a, X\}$, $\gamma \geq \max\{b, Y\}$;
- (vii) $\chi \leq \min\{a, X\}$, $\min\{b, Y\} \leq \gamma \leq \max\{b, Y\}$;
- (viii) $\chi \geq \max\{a, X\}$, $\min\{b, Y\} \leq \gamma \leq \max\{b, Y\}$;
- (ix) $\min\{a, X\} \leq \chi \leq \max\{a, X\}$, $\min\{b, Y\} \leq \gamma \leq \max\{b, Y\}$.

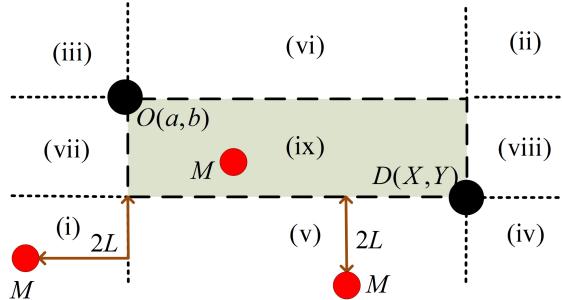


Figure EC.3 Illustration of the nine cases, depending on the location of M versus the rectangle $O \square D$.

Let us consider Case (i), assuming that $a = \min\{a, X\}$ and $Y = \min\{b, Y\}$. It holds that:

$$\begin{aligned} \|O - D\|_1 + 2L &= |a - X| + |b - Y| + 2L \\ &= |a - X| + |b - Y| + 2(|\chi - \min\{a, X\}| + |\gamma - \min\{b, Y\}|) \quad (\text{from Case (i)}) \\ &= (X - a) + (b - Y) + 2(a - \chi) + 2(Y - \gamma) \\ &= (a - \chi) + (b - \gamma) + (X - \chi) + (Y - \gamma) \\ &= \|O - M\|_1 + \|M - D\|_1. \end{aligned}$$

All instances in Cases (ii)–(iv) are treated similarly.

In Case (v), it holds that:

$$\begin{aligned}
\|O - D\|_1 + 2L &= |a - X| + |b - Y| + 2L \\
&= |a - X| + |b - Y| + 2(0 + |\gamma - \min\{b, Y\}|) \quad (\text{from Case (v)}) \\
&= (X - a) + (b - Y) + 2(Y - \gamma) \\
&= (X - \chi) + (\chi - a) + (b - Y) + 2(Y - \gamma) \\
&= (X - \chi) + (\chi - a) + (b - \gamma) + (Y - \gamma) \\
&= \|O - M\|_1 + \|M - D\|_1.
\end{aligned}$$

Cases (vi)–(viii) are treated similarly.

In Case (ix), assuming that $a = \min\{a, X\}$ and $Y = \min\{b, Y\}$, it holds that:

$$\begin{aligned}
\|O - D\|_1 + 2L &= |a - X| + |b - Y| \\
&= (X - \chi) + (\chi - a) + (b - \gamma) + (\gamma - Y) \\
&= \|O - M\|_1 + \|M - D\|_1.
\end{aligned}$$

This completes the proof. \square

Appendix EC.2: Details on MSO–VCC (Section 4)

EC.2.1. Proof of Theorem 1

PROOF. We proceed by induction over K . If $K = 1$, MSO–VCC is exactly SSO–VCC. We now assume that the statement holds for $K - 1$ visits, and show that it holds for K visits.

Recall that $\Xi^*(M_1, v_1) = (M_2^*(M_1, v_1), \dots, M_K^*(M_1, v_1))$ denotes the optimal locations of visits $2, \dots, K$, and that $\Phi(M_1, M_2, \dots, M_K) = \sum_{k=1}^K \frac{\|M_{k+1} - M_k\|}{S}$ denotes the total cost of traveling from M_1 to the destination $M_{K+1} = D$ through locations M_2, \dots, M_K . The MSO–VCC becomes:

$$\min_{M_1 \in \mathbb{F}_1(O, \bar{v})} \left\{ \bar{v} + \frac{\|M_1 - O\|}{S} + \Phi(M_1, \Xi^*(M_1, v_1)) \right\}, \quad \text{where } v_1 = \bar{v} + \frac{\|M_1 - O\|}{S}.$$

The challenge is that, even though we can derive $\Xi^*(M_1, v_1)$ for any $M_1 \in \mathbb{R}^2$ and $v_1 \geq 0$ (per the induction hypothesis), the full functional form $\Xi^*(M_1, v_1)$ is unknown.

We therefore propose the following iterative algorithm. Let $M_1^{(l)}$ and $v_1^{(l)}$ denote the location and time of visit 1 in iteration l . We initialize the algorithm with a feasible solution $M_1^{(0)}, \dots, M_K^{(0)}$, $v_1^{(0)} = \bar{v} + \frac{\|M_1^{(0)} - O\|}{S}$ and $v_{k+1}^{(0)} = v_k^{(0)} + \frac{\|M_{k+1}^{(0)} - M_k^{(0)}\|}{S}$ for all $k = 1, \dots, K$. At each iteration $l = 1, 2, \dots$, we update the locations and times of visits $2, \dots, K$ as a function of $M_1^{(l-1)}$ and $v_1^{(l-1)}$. We then re-optimize the location and time of visit 1, given all subsequent visits. This is written as follows:

$$M_1^{(l)} \in \arg \min_{M_1 \in \mathbb{F}_1(O, \bar{v})} \left\{ \bar{v} + \frac{\|M_1 - O\|}{S} + \Phi(M_1, \Xi^*(M_1^{(l-1)}, v_1^{(l-1)})) \right\}, \quad v_1^{(l)} = \bar{v} + \frac{\|M_1^{(l)} - O\|}{S}. \quad (\text{EC.34})$$

The algorithm terminates when $M_1^{(l+1)} = M_1^{(l)}$ (hence, $M_k^{(l+1)} = M_k^{(l)}$ for all $k = 1, \dots, K$).

Claim 1: The algorithm in Equation (EC.34) only involves SSO–VCC solutions.

By induction, optimizing the locations and times of visits $2, \dots, K$ (for any location M_1) is obtained by solving SSO–VCC problems iteratively. But then, once the locations $M_2^*(M_1, v_1), \dots, M_K^*(M_1, v_1)$ are fixed, minimizing the overall travel time is equivalent to minimizing the travel time from O to $M_2^*(M_1, v_1)$. Equation (EC.34) can thus be rewritten as:

$$M_1^{(l)} \in \arg \min_{M_1 \in \mathbb{F}_1(O, \bar{v})} \left\{ \bar{v} + \frac{\|M_1 - O\|}{\bar{S}} + \frac{\|M_2^*(M_1^{(l-1)}, v_1^{(l-1)}) - M_1\|}{\bar{S}} \right\}. \quad (\text{EC.35})$$

Therefore, Equation (EC.34) is equivalent to SSO–VCC, for a vehicle leaving origin O at time \bar{v} and heading to “destination” $M_2^*(M_1^{(l-1)}, v_1^{(l-1)})$. This completes the proof of Claim 1.

Claim 2: The algorithm in Equation (EC.34) does not increase costs at each iteration.

Let $\Psi(M_1, M_2, \dots, M_K)$ denote the arrival time at the vehicle’s destination D given the departure time \bar{v} from the origin O , and stopping locations M_1, M_2, \dots, M_K :

$$\Psi(M_1, M_2, \dots, M_K) = \bar{v} + \frac{\|M_1 - O\|}{\bar{S}} + \Phi(M_1, M_2, \dots, M_K).$$

Let us also denote by $\bar{\Psi}(\widehat{M}_1, \widehat{v}_1)$ the optimal objective value of MSO–VCC (Equations (4)–(9)), subject to the additional constraint that $M_1 = \widehat{M}_1$ and $v_1 = \widehat{v}_1$ (i.e., by fixing the location and time of the first visit). Per an analogous version of Lemma 5, the optimal solution satisfies $v_k = v_{k-1} + \frac{\|M_k - M_{k-1}\|}{\bar{S}}, \forall k = 1, \dots, K+1$. From the optimality of $\Xi^*(M_1^{(l+1)}, v_1^{(l+1)})$, we obtain:

$$\Psi(M_1^{(l+1)}, \Xi^*(M_1^{(l+1)}, v_1^{(l+1)})) = \bar{\Psi}(M_1^{(l+1)}, v_1^{(l+1)}). \quad (\text{EC.36})$$

Next, let us construct a solution by fixing the stopping locations to $M_1^{(l+1)}$ and $\Xi^*(M_1^{(l)}, v_1^{(l)})$, re-optimizing the time of the first visit, and leaving the times of all subsequent visits unchanged.

$$M_1 = M_1^{(l+1)}, \quad (\text{EC.37})$$

$$M_k = M_k^*(M_1^{(l)}, v_1^{(l)}) \quad \forall k = 2, \dots, K, \quad (\text{EC.38})$$

$$v_1 = \bar{v} + \frac{\|M_1^{(l+1)} - O\|}{\bar{S}}, \quad (\text{EC.39})$$

$$v_2 = \bar{v} + \frac{\|M_1^{(l)} - O\|}{\bar{S}} + \frac{\|M_2^*(M_1^{(l)}, v_1^{(l)}) - M_1^{(l)}\|}{\bar{S}}, \quad (\text{EC.40})$$

$$v_{k+1} = v_k + \frac{\|M_{k+1}^*(M_1^{(l)}, v_1^{(l)}) - M_k^*(M_1^{(l)}, v_1^{(l)})\|}{\bar{S}} \quad \forall k = 2, \dots, K. \quad (\text{EC.41})$$

By construction, the solution given in Equations (EC.37)–(EC.41) achieves a cost of $v_{K+1} = \Psi(M_1^{(l)}, \Xi^*(M_1^{(l)}, v_1^{(l)}))$. Moreover, we have, by definition of $M_1^{(l+1)}$ (Equation (EC.35)):

$$\frac{\|M_1^{(l+1)} - O\|}{\bar{S}} + \frac{\|M_2^*(M_1^{(l)}, v_1^{(l)}) - M_1^{(l+1)}\|}{\bar{S}} \leq \frac{\|M_1^{(l)} - O\|}{\bar{S}} + \frac{\|M_2^*(M_1^{(l)}, v_1^{(l)}) - M_1^{(l)}\|}{\bar{S}}.$$

Therefore, we obtain from Equations (EC.39)–(EC.40):

$$v_2 \geq v_1 + \frac{\|M_2^*(M_1^{(l)}, v_1^{(l)}) - M_1^{(l+1)}\|}{\bar{S}}.$$

In other words, after stopping in $M_1^{(l+1)}$, the vehicle can reach $M_2^*(M_1^{(l)}, v_1^{(l)})$ by time v_2 . All other constraints are satisfied by construction, so Equations (EC.37)–(EC.41) define a feasible MSO–VCC solution. Therefore, $\bar{\Psi}(M_1^{(l+1)}, v_1^{(l+1)}) \leq \Psi(M_1^{(l)}, \Xi^*(M_1^{(l)}, v_1^{(l)}))$. Combined with Equation (EC.36), we obtain the following inequality, which completes the proof of Claim 2:

$$\Psi(M_1^{(l+1)}, \Xi^*(M_1^{(l+1)}, v_1^{(l+1)})) \leq \Psi(M_1^{(l)}, \Xi^*(M_1^{(l)}, v_1^{(l)})).$$

Claim 3: The algorithm in Equation (EC.34) terminates at the optimum if $\|\cdot\|$ is differentiable.

As a sum of norms, $\Psi(M_1, M_2, \dots, M_K)$ is convex. By assumption, it is also differentiable. Let us denote by $\frac{\partial \Psi}{\partial x(M_k)}$ and $\frac{\partial \Psi}{\partial y(M_k)}$ the partial derivative of Ψ with respect to the x - and y -coordinates of M_k , respectively. Upon termination, $M_1^{(l^*+1)} = M_1^{(l^*)}$, i.e., $M_1^{(l^*)}$ is the optimal location of visit 1, given locations $M_2^*(M_1^{(l^*)}, v_1^{(l^*)}), \dots, M_K^*(M_1^{(l^*)}, v_1^{(l^*)})$ (Equation (EC.35)). Therefore:

$$\left. \frac{\partial \Psi}{\partial x(M_1)} \right|_{M_1=M_1^{(l^*)}} = \left. \frac{\partial \Psi}{\partial y(M_1)} \right|_{M_1=M_1^{(l^*)}} = 0. \quad (\text{EC.42})$$

Moreover, from the optimality of $M_2^*(M_1^{(l^*)}, v_1^{(l^*)}), \dots, M_K^*(M_1^{(l^*)}, v_1^{(l^*)})$, we have:

$$\left. \frac{\partial \Psi}{\partial x(M_k)} \right|_{M_k=M_k^*(M_1^{(l^*)}, v_1^{(l^*)})} = \left. \frac{\partial \Psi}{\partial y(M_k)} \right|_{M_k=M_k^*(M_1^{(l^*)}, v_1^{(l^*)})} = 0. \quad \forall k = 2, \dots, K, \quad (\text{EC.43})$$

By convexity of Ψ , we also have, for all M_1, \dots, M_K :

$$\begin{aligned} \Psi(M_1, \dots, M_K) &\geq \Psi(M_1^{(l^*)}, \Xi^*(M_1^{(l^*)}, v_1^{(l^*)})) \\ &\quad + \underbrace{\nabla \Psi(M_1^{(l^*)}, \Xi^*(M_1^{(l^*)}, v_1^{(l^*)}))^T}_{=0 \text{ per Equations (EC.42)–(EC.43)}} \left((M_1, \dots, M_K) - (M_1^{(l^*)}, \Xi^*(M_1^{(l^*)}, v_1^{(l^*)})) \right). \end{aligned}$$

Therefore, $(M_1^{(l^*)}, \Xi^*(M_1^{(l^*)}, v_1^{(l^*)}))$ is optimal. This completes the proof of Claim 3.

Claim 4: The algorithm in Equation (EC.34) terminates at the optimum if $\|\cdot\|$ is the ℓ_1 -norm.

Before proceeding, recall from Remark 1 that the solution of SSO–VCC is never on the “border” of the rectangle defined by the previous and subsequent stops, whenever possible.

A. Preliminaries.

For notational ease, let us denote $(M_1^*, \dots, M_K^*) = \left(M_1^{(l^*)}, \Xi^* \left(M_1^{(l^*)}, v_1^{(l^*)} \right) \right)$, and by v_k^* the stopping time in location M_k^* , for $k = 1, \dots, K$ (obtained from Lemma 5). Let $x(M_k)$ and $y(M_k)$ denote the x - and y -coordinates of M_k . We expand the function $\Psi(M_1, M_2, \dots, M_K)$ using the ℓ_1 -norm:

$$\begin{aligned}\Psi(M_1, M_2, \dots, M_K) &= \bar{v} + \sum_{k=0}^K \frac{\|M_{k+1} - M_k\|_1}{\bar{S}} \quad (\text{with } M_0 = O \text{ and } M_{K+1} = D) \\ &= \bar{v} + \sum_{k=0}^K \frac{|x(M_{k+1}) - x(M_k)|}{\bar{S}} + \sum_{k=0}^K \frac{|y(M_{k+1}) - y(M_k)|}{\bar{S}}.\end{aligned}$$

Each term $x(M_k)$ appears in two terms: $\frac{|x(M_{k+1}) - x(M_k)|}{\bar{S}}$ and $\frac{|x(M_k) - x(M_{k-1})|}{\bar{S}}$. Therefore, $\Psi(M_1, M_2, \dots, M_K)$ is differentiable in $x(M_k)$ (resp. $y(M_k)$) if and only if $x(M_k) \neq x(M_{k+1})$ and $x(M_k) \neq x(M_{k-1})$ (resp. $y(M_k) \neq y(M_{k+1})$ and $y(M_k) \neq y(M_{k-1})$).

B. Perturbation.

Let $\varepsilon > 0$ be a small number. We introduce a perturbed solution $(\tilde{M}_1, \dots, \tilde{M}_K)$ and corresponding times $\tilde{v}_1, \dots, \tilde{v}_K$, such that the x - and y -coordinates of consecutive points will be different:

- If $x(M_k^*) \neq x(M_{k+1}^*)$, $x(M_k^*) \neq x(M_{k-1}^*)$, $y(M_k^*) \neq y(M_{k+1}^*)$, and $y(M_k^*) \neq y(M_{k-1}^*)$, we have $\tilde{M}_k = M_k^*$, i.e.: $x(\tilde{M}_k) = x(M_k^*)$ and $y(\tilde{M}_k) = y(M_k^*)$.
- Otherwise, we build a perturbation $x(\tilde{M}_k)$ and $y(\tilde{M}_k)$ such that

$$x(\tilde{M}_k) \neq x(\tilde{M}_{k+1}) \text{ and } x(\tilde{M}_k) \neq x(\tilde{M}_{k-1}), \tag{EC.44}$$

$$y(\tilde{M}_k) \neq y(\tilde{M}_{k+1}) \text{ and } y(\tilde{M}_k) \neq y(\tilde{M}_{k-1}), \tag{EC.45}$$

$$\tilde{M}_k \in \mathbb{F}_k \left(\tilde{M}_{k-1}, \tilde{v}_{k-1} \right), \tag{EC.46}$$

$$x(\tilde{M}_k) - x(M_k^*) \in \left\{ -\frac{k\varepsilon}{K}, 0, \frac{k\varepsilon}{K} \right\}, \tag{EC.47}$$

$$y(\tilde{M}_k) - y(M_k^*) \in \left\{ -\frac{k\varepsilon}{K}, 0, \frac{k\varepsilon}{K} \right\}, \tag{EC.48}$$

$$\tilde{v}_k = \tilde{v}_{k-1} + \frac{\|\tilde{M}_k - \tilde{M}_{k-1}\|_1}{\bar{S}}, \forall k = 1, \dots, K+1. \tag{EC.49}$$

This perturbation is designed to ensure that Ψ is differentiable in $(\tilde{M}_1, \dots, \tilde{M}_K)$, while complying with Lemma 5 (i.e., the vehicle never waits for a customer at the stopping location). The terms $\frac{k\varepsilon}{K}$ ensure that the deviations are increasing in k . Note that, given basic features of the ℓ_1 -norm, each feasible region is continuous, so there exists a sequence $(\tilde{M}_1, \dots, \tilde{M}_K)$ that satisfies the above properties. We will show this by construction, in the proof of Lemma EC.2.

C. Differentiation of Ψ .

By construction, Ψ is differentiable in $(\widetilde{M}_1, \dots, \widetilde{M}_K)$. By convexity, we have, for all M_1, \dots, M_K :

$$\begin{aligned}\Psi(M_1, \dots, M_K) &\geq \Psi(\widetilde{M}_1, \dots, \widetilde{M}_K) + \nabla \Psi(\widetilde{M}_1, \dots, \widetilde{M}_K)^T ((M_1, \dots, M_K) - (\widetilde{M}_1, \dots, \widetilde{M}_K)) \\ &= \Psi(\widetilde{M}_1, \dots, \widetilde{M}_K) \\ &\quad + \sum_{k \in \{1, \dots, K\}} \frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k = \widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) + \sum_{k \in \{1, \dots, K\}} \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k = \widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)).\end{aligned}$$

To analyze the above inequality, we resort to the following lemma.

LEMMA EC.2. *The perturbed solutions $(\widetilde{M}_1, \dots, \widetilde{M}_K)$ (Equations (EC.44)–(EC.48)) satisfy:*

$$\Psi(M_1, \dots, M_K) \geq \Psi(\widetilde{M}_1, \dots, \widetilde{M}_K) - \frac{16\varepsilon}{S} \frac{K(K+1)}{2} \quad \forall M_1, \dots, M_K. \quad (\text{EC.50})$$

Therefore, any perturbation from $(\widetilde{M}_1, \dots, \widetilde{M}_K)$ will either worsen the objective, or improve it by a factor $\mathcal{O}(\varepsilon)$. By taking the limit when $\varepsilon \rightarrow 0$, we conclude, from the continuity of Ψ :

$$\Psi(M_1, \dots, M_K) \geq \Psi(M_1^*, \dots, M_K^*) \quad \forall M_1, \dots, M_K.$$

This proves the optimality of solution (M_1^*, \dots, M_K^*) with the ℓ_1 -norm. \square

Proof of Lemma EC.2

PROOF. We first characterize the stopping times in $\widetilde{M}_1, \dots, \widetilde{M}_K$ in Lemma EC.3. Specifically, the difference between \tilde{v}_k and v_k^* increases with k (detours accumulate) but remains within $\mathcal{O}(\varepsilon)$.

LEMMA EC.3. *The following holds: $\tilde{v}_k \leq v_k^* + \frac{2(2k-1)\varepsilon}{S}$, $\forall k = 1, \dots, K$.*

PROOF OF LEMMA EC.3. Let us fix $k = 2, \dots, K$. Note that:

$$\begin{aligned}\tilde{v}_k &= \tilde{v}_{k-1} + \frac{\|\widetilde{M}_k - \widetilde{M}_{k-1}\|_1}{S} \quad (\text{Lemma 5}) \\ &\leq \tilde{v}_{k-1} + \frac{\|\widetilde{M}_k - M_k^*\|_1 + \|M_k^* - M_{k-1}^*\|_1 + \|M_{k-1}^* - \widetilde{M}_{k-1}\|_1}{S} \quad (\text{triangle inequality}) \\ &\leq \tilde{v}_{k-1} + \frac{4k\varepsilon}{KS} + \frac{\|M_k^* - M_{k-1}^*\|_1}{S} \quad \text{because } \|\widetilde{M}_k - M_k^*\|_1 \leq \frac{2k}{K}\varepsilon \text{ for all } k = 1, \dots, K \\ &\leq \tilde{v}_{k-1} + \frac{4\varepsilon}{S} + \frac{\|M_k^* - M_{k-1}^*\|_1}{S}.\end{aligned}$$

For $k = 1$, we modify the statement as follows (since the origin O is fixed):

$$\tilde{v}_1 \leq \bar{v} + \frac{2\varepsilon}{S} + \frac{\|M_1^* - O\|_1}{S} = v_1^* + \frac{2\varepsilon}{S}.$$

This proves Lemma EC.3 for $k = 1$. We conclude by induction:

$$\begin{aligned}\tilde{v}_k &\leq \tilde{v}_{k-1} + \frac{4\varepsilon}{S} + \frac{\|M_k^* - M_{k-1}^*\|_1}{S} \\ &\leq v_{k-1}^* + \frac{2(2k-3)\varepsilon}{S} + \frac{4\varepsilon}{S} + \frac{\|M_k^* - M_{k-1}^*\|_1}{S} \quad (\text{induction hypothesis}) \\ &= v_k^* + \frac{2(2k-1)\varepsilon}{S} \quad (\text{Lemma 5}). \quad \square\end{aligned}$$

Let \mathcal{P}^x (resp. \mathcal{P}^y) be the sets of indices k such that Ψ is differentiable in $x(M_k^*)$ (resp. $y(M_k^*)$):

$$\mathcal{P}^x = \{1, \dots, K | x(M_k^*) \neq x(M_{k+1}^*), x(M_k^*) \neq x(M_{k-1}^*)\},$$

$$\mathcal{P}^y = \{1, \dots, K | y(M_k^*) \neq y(M_{k+1}^*), y(M_k^*) \neq y(M_{k-1}^*)\}.$$

We then show the following statement, for each $k = 1, \dots, K$ (which implies Equation (EC.50)):

$$\sum_{k \in \{1, \dots, K\}} \frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k = \tilde{M}_k} (x(M_k) - x(\tilde{M}_k)) + \sum_{k \in \{1, \dots, K\}} \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k = \tilde{M}_k} (y(M_k) - y(\tilde{M}_k)) \geq -\frac{16k}{S} \varepsilon. \quad (\text{EC.51})$$

Let us first note that the derivatives can take at most three values:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k = \tilde{M}_k} = \begin{cases} 0 & \text{if } x(\tilde{M}_{k-1}) < x(\tilde{M}_k) < x(\tilde{M}_{k+1}) \text{ or } x(\tilde{M}_{k-1}) > x(\tilde{M}_k) > x(\tilde{M}_{k+1}) \\ -\frac{2}{S} & \text{if } x(\tilde{M}_k) \leq x(\tilde{M}_{k-1}) \text{ and } x(\tilde{M}_k) \leq x(\tilde{M}_{k+1}) \\ +\frac{2}{S} & \text{if } x(\tilde{M}_k) \geq x(\tilde{M}_{k-1}) \text{ and } x(\tilde{M}_k) \geq x(\tilde{M}_{k+1}). \end{cases} \quad (\text{EC.52})$$

$$\frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k = \tilde{M}_k} = \begin{cases} 0 & \text{if } y(\tilde{M}_{k-1}) < y(\tilde{M}_k) < y(\tilde{M}_{k+1}) \text{ or } y(\tilde{M}_{k-1}) > y(\tilde{M}_k) > y(\tilde{M}_{k+1}) \\ -\frac{2}{S} & \text{if } y(\tilde{M}_k) \leq y(\tilde{M}_{k-1}) \text{ and } y(\tilde{M}_k) \leq y(\tilde{M}_{k+1}) \\ +\frac{2}{S} & \text{if } y(\tilde{M}_k) \geq y(\tilde{M}_{k-1}) \text{ and } y(\tilde{M}_k) \geq y(\tilde{M}_{k+1}). \end{cases} \quad (\text{EC.53})$$

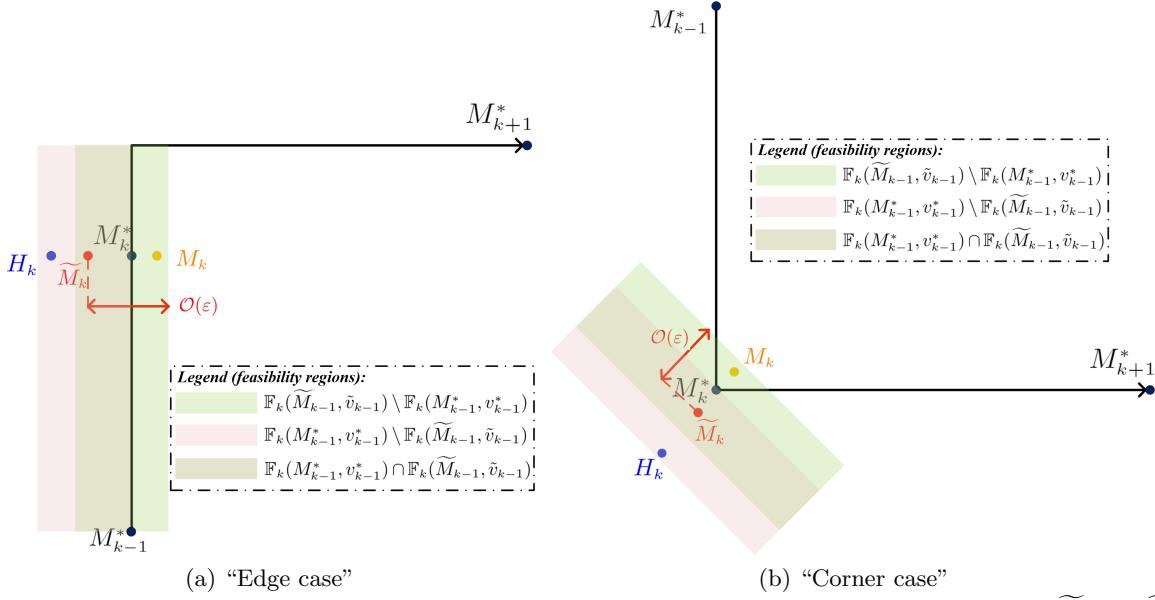


Figure EC.4 Illustration of the feasible regions with the sequence (M_1^*, \dots, M_K^*) vs. the sequence $(\tilde{M}_1, \dots, \tilde{M}_K)$.

Case 1: $k \in \mathcal{P}^x$ and $k \in \mathcal{P}^y$. By construction, $\tilde{M}_k = M_k^*$. We consider the four sub-cases: (i) $x(M_{k-1}^*) < x(M_k^*) < x(M_{k+1}^*)$; (ii) $x(M_{k-1}^*) > x(M_k^*) > x(M_{k+1}^*)$; (iii) $x(M_k^*) < x(M_{k-1}^*)$ and $x(M_k^*) < x(M_{k+1}^*)$; and (iv) $x(M_k^*) > x(M_{k-1}^*)$ and $x(M_k^*) > x(M_{k+1}^*)$.

Subcases (i) and (ii). From Equation (EC.52), we have $\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k = M_k^*} = 0$.

Subcase (iii). We must have $x(H_k) < x(M_k^*)$. Indeed, if $x(H_k) > x(M_k^*)$, there exists another stopping location M'_k with $x(M'_k) > x(M_k^*)$ that is closer to M_{k-1}^* and M_{k+1}^* , which contradicts Remark 1. Moreover, if $x(H_k) = x(M_k^*)$, we must have $y(M_k^*) = y(M_{k-1}^*)$ or $y(M_k^*) = y(M_{k+1}^*)$,

which contradicts $k \in \mathcal{P}^y$. From Equation (EC.52), we obtain $\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=M_k^*} = -\frac{2}{S}$. Therefore, for each M_k such that $x(M_k) \leq x(M_k^*)$:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=M_k^*} (x(M_k) - x(M_k^*)) = +\frac{2}{S} |x(M_k) - x(M_k^*)| \geq 0.$$

Let us turn to the case where $x(M_k) > x(M_k^*)$. In the original solution, $\mathbb{F}_k(M_{k-1}^*, v_{k-1}^*)$ is included in the hyperplane $\{M_k : x(M_k) \leq x(M_k^*)\}$ (otherwise, there would exist another stopping location M'_k with $x(M'_k) > x(M_k^*)$ that is closer to M_{k-1}^* and M_{k+1}^* , which contradicts the optimality of the SSO-VCC solution). However, $\mathbb{F}_k(\tilde{M}_{k-1}, \tilde{v}_{k-1})$ is not necessarily included in that hyperplane: since the vehicle arrives later in M_{k-1}^* (due to earlier perturbations), the customer can walk further. Yet, these perturbations are $\mathcal{O}(\varepsilon)$ (Lemma EC.3). This situation is illustrated in Figure EC.4a.

Let M_k be such that $x(M_k) > x(M_k^*)$. We know that $\frac{\|H_k - M_k^*\|_1}{S} \leq v_k^*$ (Lemma 5) and that $\|H_k - M_k\|_1 = \|H_k - M_k^*\|_1 + \|M_k^* - M_k\|_1$ (M_k^* is on the “optimal ℓ_1 path” from H_k to M_k). Thus:

$$\begin{aligned} \frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=M_k^*} (x(M_k) - x(M_k^*)) &= -\frac{2}{S} (x(M_k) - x(M_k^*)) \\ &\geq -\frac{2}{S} \|M_k - M_k^*\|_1 \\ &= -\frac{2}{S} [\|M_k - H_k\|_1 - \|H_k - M_k^*\|_1] \\ &\geq -\frac{2}{S} [\tilde{v}_k \times S_k - v_k^* \times S_k] \quad (\text{Equation (7) and Lemma 5}) \\ &\geq -\frac{2}{S} \frac{2(2k-1)S_k\varepsilon}{S} \quad (\text{Lemma EC.3}) \\ &\geq -\frac{8k}{S}\varepsilon. \end{aligned}$$

Subcase (iv) is treated similarly.

Case 2: $k \notin \mathcal{P}^x$ and $k \in \mathcal{P}^y$. In this case, M_k^* is on an “edge” of the rectangle between M_{k-1}^* and M_{k+1}^* but not on a “corner”. Since $k \in \mathcal{P}^y$, we have $y(M_k^*) \neq y(M_{k-1}^*)$ and $y(M_k^*) \neq y(M_{k+1}^*)$. Due to Remark 1, the feasible region $\mathbb{F}_k(M_{k-1}^*, v_{k-1}^*)$ is contained in the hyperplane $\{M_k : x(M_k) \leq x(M_k^*)\}$ or in the hyperplane $\{M_k : x(M_k) \leq x(M_k^*)\}$. We consider three subcases: (i) $x(M_{k-1}^*) = x(M_k^*) \neq x(M_{k+1}^*)$; (ii) $x(M_{k-1}^*) \neq x(M_k^*) = x(M_{k+1}^*)$; (iii) $x(M_{k-1}^*) = x(M_k^*) = x(M_{k+1}^*)$. We analyze these cases separately:

Subcase (i). Without loss of generality, $x(M_{k-1}^*) = x(M_k^*) < x(M_{k+1}^*)$ and that $y(M_{k-1}^*) < y(M_k^*)$. We distinguish two instances, depending on whether $y(M_{k+1}^*) > y(M_k^*)$ or $y(M_{k+1}^*) < y(M_k^*)$.

- a. If $y(M_{k+1}^*) > y(M_k^*)$, we know that $x(H_k) < x(M_k^*)$ (due to the optimality of SSO-VCC and Remark 1). We then build $x(\tilde{M}_k) = x(M_k^*) - \frac{k\varepsilon}{K}$. Moreover, we also have $x(\tilde{M}_{k-1}) = x(M_{k-1}^*) \pm \frac{(k-1)\varepsilon}{K} > x(\tilde{M}_k)$, so $x(\tilde{M}_{k+1}) > x(\tilde{M}_k)$. From Equation (EC.52), we obtain $\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\tilde{M}_k} = -\frac{2}{S}$.

The remaining discussion is similar to Subcase (iii) in Case 1, except that $\widetilde{M}_k \neq M_k^*$. For each M_k such that $x(M_k) \leq x(\widetilde{M}_k)$, we have $\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) = +\frac{2}{S} |x(M_k) - x(\widetilde{M}_k)| \geq 0$. Vice versa, if $x(M_k) > x(\widetilde{M}_k)$, the following holds (Figure EC.4a):

$$\begin{aligned} \frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) &\geq -\frac{2}{S} \|M_k - \widetilde{M}_k\|_1 \\ &\geq -\frac{2}{S} [\|M_k - M_k^*\|_1 + \|M_k^* - \widetilde{M}_k\|] \quad (\text{triangle inequality}) \\ &= -\frac{2}{S} [(\|M_k - H_k\|_1 - \|H_k - M_k^*\|_1) + \|M_k^* - \widetilde{M}_k\|] \\ &\geq -\frac{2}{S} \left[\tilde{v}_k \times S_k - v_k^* \times S_k + \frac{k\varepsilon}{K} \right] \quad (\text{Equation (7), (EC.47) and Lemma 5}) \\ &\geq -\frac{2}{S} \left[\frac{2(2k-1)S_k\varepsilon}{S} + \frac{k\varepsilon}{K} \right] \quad (\text{Lemma EC.3}) \\ &\geq -\frac{8k}{S}\varepsilon. \end{aligned}$$

- b. If $y(M_{k+1}^*) < y(M_k^*)$, we can have $x(H_k) < x(M_k^*)$ (as before) or $x(H_k) = x(M_k^*)$ and $y(H_k) > y(M_k^*)$. We cannot simply perturb $x(M_k^*)$ by $\frac{k\varepsilon}{K}$ because this would violate Equation (EC.49). Therefore, we also perturb $y(M_k^*)$. We build $x(\widetilde{M}_k) = x(M_k^*) + \frac{k\varepsilon}{K}$ and $y(\widetilde{M}_k) = y(M_k^*) + \frac{k\varepsilon}{K}$. By construction, Equations (EC.44)–(EC.48) are satisfied. From Equation (EC.52), we have $\frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} = 0$. Similarly, from Equation (EC.53), $\frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} = \frac{2}{S}$.

It comes, for each M_k satisfying $y(M_k) \geq y(\widetilde{M}_k)$:

$$\frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) = +\frac{2}{S} |y(M_k) - y(\widetilde{M}_k)|.$$

If $y(M_k) < y(\widetilde{M}_k)$, we proceed as before and obtain:

$$\begin{aligned} \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) &= -\frac{2}{S} (y(\widetilde{M}_k) - y(M_k)) \\ &\geq -\frac{2}{S} \|M_k - \widetilde{M}_k\|_1 \\ &\geq -\frac{2}{S} [\|M_k - M_k^*\|_1 + \|M_k^* - \widetilde{M}_k\|_1] \quad (\text{triangle inequality}) \\ &= -\frac{2}{S} [(\|M_k - H_k\|_1 - \|H_k - M_k^*\|_1) + \|M_k^* - \widetilde{M}_k\|_1] \\ &\geq -\frac{2}{S} \left[\tilde{v}_k \times S_k - v_k^* \times S_k + \frac{k\varepsilon}{K} \right] \quad (\text{Equation (7), (EC.48) and Lemma 5}) \\ &\geq -\frac{2}{S} \left[\frac{2(2k-1)S_k\varepsilon}{S} + \frac{k\varepsilon}{K} \right] \quad (\text{Lemma EC.3}) \\ &\geq -\frac{8k}{S}\varepsilon. \end{aligned}$$

Subcase (ii). Without loss of generality, $x(M_{k-1}^*) < x(M_k^*) = x(M_{k+1}^*)$ and that $y(M_k^*) < y(M_{k+1}^*)$. We distinguish between $y(M_{k-1}^*) > y(M_k^*)$ and $y(M_{k-1}^*) < y(M_k^*)$.

- a. If $y(M_{k-1}^*) < y(M_k^*)$, we know that $x(H_k) > x(M_k^*)$. We build $x(\widetilde{M}_k) = x(M_k^*) + \frac{k\varepsilon}{K}$. We have $x(\widetilde{M}_{k+1}) = x(M_{k+1}^*) \pm \frac{(k+1)\varepsilon}{K}$. If $x(\widetilde{M}_{k+1}) = x(M_{k+1}^*) + \frac{(k+1)\varepsilon}{K}$, we have $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} = 0$ (Equation (EC.52)). If $x(\widetilde{M}_{k+1}) = x(M_{k+1}^*) - \frac{(k+1)\varepsilon}{K}$, we have $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} = \frac{2}{S}$ (Equation (EC.52)). We conclude as before.
- b. If $y(M_{k-1}^*) > y(M_k^*)$, we can have $x(H_k) < x(M_k^*)$ (as in the previous case) or $x(H_k) = x(M_k^*)$. In the latter case, we perturb both $x(\widetilde{M}_k)$ and $y(\widetilde{M}_k)$ and conclude as in Subcase (i).

Subcase (iii). This case is very similar to Subcases (i) and (ii). For simplicity of the exposition, we assume that $x(H_{k-1}) \neq x(M_{k-1}^*)$, $x(H_{k+1}) \neq x(M_{k+1}^*)$ (the other cases are treated identically). Without loss of generality, we assume that $x(H_{k-1}) < x(M_{k-1}^*)$, so that $x(\widetilde{M}_{k-1}) = x(M_{k-1}^*) - \frac{(k-1)\varepsilon}{K}$.

- a. If $x(H_k) < x(M_k^*)$ and $x(H_{k+1}) < x(M_{k+1}^*)$, we have $x(\widetilde{M}_{k-1}) = x(M_{k-1}^*) - \frac{(k-1)\varepsilon}{K}$, $x(\widetilde{M}_k) = x(M_k^*) - \frac{k\varepsilon}{K}$ and $x(\widetilde{M}_{k+1}) = x(M_{k+1}^*) - \frac{(k+1)\varepsilon}{K}$. Thus, $x(\widetilde{M}_{k-1}) > x(\widetilde{M}_k) > x(\widetilde{M}_{k+1})$. From Equation (EC.52), $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} = 0$.
- b. If $x(H_k) < x(M_k^*)$ and $x(H_{k+1}) > x(M_{k+1}^*)$, we have $x(\widetilde{M}_{k-1}) = x(M_{k-1}^*) - \frac{(k-1)\varepsilon}{K}$, $x(\widetilde{M}_k) = x(M_k^*) - \frac{k\varepsilon}{K}$ and $x(\widetilde{M}_{k+1}) = x(M_{k+1}^*) + \frac{(k+1)\varepsilon}{K}$. Thus, $x(\widetilde{M}_{k+1}) > x(\widetilde{M}_{k-1}) > x(\widetilde{M}_k)$. From Equation (EC.52), $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} = -\frac{2}{S}$. We conclude as before: if $x(M_k) \leq x(\widetilde{M}_k)$, then $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) = +\frac{2}{S} |x(M_k) - x(\widetilde{M}_k)| \geq 0$; and if $x(M_k) > x(\widetilde{M}_k)$, then $|x(M_k) - x(\widetilde{M}_k)| = \mathcal{O}(\varepsilon)$ and $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) \geq -\frac{8k}{S}\varepsilon$.
- c. If $x(H_k) > x(M_k^*)$ and $x(H_{k+1}) < x(M_{k+1}^*)$, we proceed as in the b. by symmetry, and obtain $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) \geq -\frac{8k}{S}\varepsilon$.
- d. If $x(H_k) > x(M_k^*)$ and $x(H_{k+1}) > x(M_{k+1}^*)$, we obtain, as in a., $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} = 0$.
- e. If $x(H_k) = x(M_k^*)$, we need to perturb both $x(M_k^*)$ and $y(M_k^*)$ to comply with Equation (EC.49). First, we can build $x(\widetilde{M}_k) = x(M_k^*) - \frac{k\varepsilon}{K}$ or $x(\widetilde{M}_k) = x(M_k^*) + \frac{k\varepsilon}{K}$. By proceeding as earlier, we show that $\frac{\partial\Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) \geq -\frac{8k}{S}\varepsilon$.

Without loss of generality, we assume $y(M_{k-1}^*) < y(M_k^*)$ (so that $y(M_k^*) < y(H_k)$). Thus, we build $y(\widetilde{M}_k) = y(M_k^*) + \frac{k\varepsilon}{K}$. If $y(M_{k+1}^*) > y(M_k^*)$, we have $\frac{\partial\Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} = 0$ (Equation (EC.53)). Otherwise, $\frac{\partial\Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} = \frac{2}{S}$ (Equation (EC.53)). The remaining discussion is the same as that of Subcase (i).b., so that $\frac{\partial\Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) \geq -\frac{8k}{S}\varepsilon$.

Case 3: if $k \in \mathcal{P}^x$ and $k \notin \mathcal{P}^y$. This is symmetric to Case 2, and thus omitted.

Case 4: if $k \notin \mathcal{P}^x$ and $k \notin \mathcal{P}^y$. By proceeding as earlier, we need to distinguish nine subcases:

- (i) $x(M_{k-1}^*) = x(M_k^*) \neq x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) \neq y(M_{k+1}^*)$
- (ii) $x(M_{k-1}^*) = x(M_k^*) \neq x(M_{k+1}^*)$ and $y(M_{k-1}^*) \neq y(M_k^*) = y(M_{k+1}^*)$
- (iii) $x(M_{k-1}^*) = x(M_k^*) \neq x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) = y(M_{k+1}^*)$
- (iv) $x(M_{k-1}^*) \neq x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) \neq y(M_{k+1}^*)$
- (v) $x(M_{k-1}^*) \neq x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) \neq y(M_k^*) = y(M_{k+1}^*)$

- (vi) $x(M_{k-1}^*) \neq x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) = y(M_{k+1}^*)$
- (vii) $x(M_{k-1}^*) = x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) \neq y(M_{k+1}^*)$
- (viii) $x(M_{k-1}^*) = x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) \neq y(M_k^*) = y(M_{k+1}^*)$
- (ix) $x(M_{k-1}^*) = x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) = y(M_{k+1}^*)$.

In both Subcases (ii) and (iv), M_k^* is on a “corner” of the rectangle defined by M_{k-1}^* and M_{k+1}^* .

In Subcases (i), (iii) and (vii), we have $M_{k-1}^* = M_k^* \neq M_{k+1}^*$. In Subcases (v), (vi) and (viii), $M_{k-1}^* \neq M_k^* = M_{k+1}^*$. And in Subcase (ix), $M_{k-1}^* = M_k^* = M_{k+1}^*$. Without loss of generality, we only treat Subcases (ii), (i), (v) and (ix).

Subcase (ii). Without loss of generality, we assume that $x(M_{k-1}^*) = x(M_k^*) < x(M_{k+1}^*)$ and $y(M_{k-1}^*) > y(M_k^*) = y(M_{k+1}^*)$. In this case, $x(H_k) \leq x(M_k^*)$ and $y(H_k) \leq y(M_k^*)$ (otherwise, the solution would contradict the optimality of SSO–VCC and Remark 1).

a. If $x(H_k) < x(M_k^*)$ and $y(H_k) < y(M_k^*)$, we have:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} = \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} = -\frac{2}{S}.$$

First, if $x(M_k) \leq x(\widetilde{M}_k)$ and $y(M_k) \leq y(\widetilde{M}_k)$, then we directly obtain:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) + \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) = +\frac{2}{S} \|M_k - \widetilde{M}_k\|_1 \geq 0.$$

Let us now treat the case where $x(M_k) > x(\widetilde{M}_k)$ and/or $y(M_k) > y(\widetilde{M}_k)$. Unlike earlier, the deviation along one dimension can be large. However, this is necessarily compensated along the other dimension, so any deviation has an ultimate impact of the order $\mathcal{O}(\varepsilon)$ (Figure EC.4b).

Since M_k^* lies on the boundary of the optimal region, the customer walks a maximal amount under solution (M_1^*, \dots, M_K^*) —that is, either $\|M_k^* - H_k\|_1 = \min(v_k^* \times S_k, W_k)$. Under solution $(\widetilde{M}_1, \dots, \widetilde{M}_K)$, the customer can walk up to $\min(\tilde{v}_k \times S_k, W_k)$. Therefore, the maximal added distance to the customer under solution $(\widetilde{M}_1, \dots, \widetilde{M}_K)$, as compared to (M_1^*, \dots, M_K^*) , is equal to $(\tilde{v}_k - v_k^*) \times S_k$. It comes:

$$\begin{aligned} & \frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) + \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) \\ &= -\frac{2}{S} (x(M_k) - x(\widetilde{M}_k) + y(M_k) - y(\widetilde{M}_k)) \\ &= -\frac{2}{S} \left(x(M_k) - \left(x(M_k^*) - \frac{k\varepsilon}{K} \right) + y(M_k) - \left(y(M_k^*) - \frac{k\varepsilon}{K} \right) \right) \\ &= -\frac{2}{S} \left[(x(M_k) - x(H_k)) + (y(M_k) - y(H_k)) - (x(M_k^*) - x(H_k)) - (y(M_k^*) - y(H_k)) + 2\frac{k\varepsilon}{K} \right] \\ &\geq -\frac{2}{S} \left[\|M_k - H_k\|_1 - \|M_k^* - H_k\|_1 + 2\frac{k\varepsilon}{K} \right] \quad \text{because } x(M_k^*) \geq x(H_k) \text{ and } y(M_k^*) \geq y(H_k) \end{aligned}$$

$$\begin{aligned}
&\geq -\frac{2}{\bar{S}} \left[(\tilde{v}_k - v_k^*) \times S_k + 2 \frac{k\varepsilon}{K} \right] \\
&\geq -\frac{2}{\bar{S}} \left[\frac{2(2k-1)S_k\varepsilon}{\bar{S}} + 2 \frac{k\varepsilon}{K} \right] \quad (\text{Lemma EC.3}) \\
&\geq -\frac{8k}{\bar{S}} \varepsilon.
\end{aligned}$$

- b. If $x(H_k) = x(M_k^*)$ and $y(H_k) < y(M_k^*)$, we perturb both $x(M_k^*)$ and $y(M_k^*)$, such that $x(\widetilde{M}_k) = x(M_k^*) + \frac{k\varepsilon}{K}$ and $y(\widetilde{M}_k) = y(M_k^*) - \frac{k\varepsilon}{K}$. The analysis is identical to Case 2, Subcase (i).b.
- c. If $x(H_k) < x(M_k^*)$ and $y(H_k) = y(M_k^*)$, we proceed similarly by symmetry.

Subcase (i). Without loss of generality, we assume $x(M_{k-1}^*) = x(M_k^*) < x(M_{k+1}^*)$ and $y(M_{k-1}^*) = y(M_k^*) < y(M_{k+1}^*)$. We have $x(H_k) \leq x(M_k^*)$, $y(H_k) \leq y(M_k^*)$, $x(H_{k-1}) \leq x(M_{k-1}^*)$ and $y(H_{k-1}) \leq y(M_{k-1}^*)$ (otherwise, the solution would contradict the optimality of SSO-VCC and Remark 1).

The analysis reduces to earlier instances, and thus omitted for conciseness.

Subcase (v). Without loss of generality, we assume $x(M_{k-1}^*) > x(M_k^*) = x(M_{k+1}^*)$ and $y(M_{k-1}^*) > y(M_k^*) = y(M_{k+1}^*)$. We have $x(H_k) \leq x(M_k^*)$, $y(H_k) \leq y(M_k^*)$, $x(H_{k+1}) \leq x(M_{k+1}^*)$ and $y(H_{k+1}) \leq y(M_{k+1}^*)$ (otherwise, the solution would contradict the optimality of SSO-VCC and Remark 1).

The analysis reduces to earlier instances, and thus omitted for conciseness.

Subcase (ix). We have $M_{k-1}^* = M_k^* = M_{k+1}^*$. Consider the four quadrants $\{(x, y) | x \leq x(M_k^*), y \leq y(M_k^*)\}$, $\{(x, y) | x \leq x(M_k^*), y \geq y(M_k^*)\}$, $\{(x, y) | x \geq x(M_k^*), y \leq y(M_k^*)\}$ and $\{(x, y) | x \geq x(M_k^*), y \geq y(M_k^*)\}$. Then, H_{k-1} , H_k and H_{k+1} must lie within the same one. We need to consider many instances, based on whether $x(H_k) < x(M_k^*)$ or $x(H_k) = x(M_k^*)$, on whether $y(H_k) < y(M_k^*)$ or $y(H_k) = y(M_k^*)$, on whether $x(H_{k-1}) < x(M_{k-1}^*)$ or $x(H_{k-1}) = x(M_{k-1}^*)$, on whether $y(H_{k-1}) < y(M_{k-1}^*)$ or $y(H_{k-1}) = y(M_{k-1}^*)$, on whether $x(H_{k+1}) < x(M_{k+1}^*)$ or $x(H_{k+1}) = x(M_{k+1}^*)$, and on whether $y(H_{k+1}) < y(M_{k+1}^*)$ or $y(H_{k+1}) = y(M_{k+1}^*)$. However, all instances reduce to earlier instances, and thus omitted for conciseness.

In summary, we conclude that for all $k = 1, \dots, K$, we either have:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) \geq -\frac{8k}{\bar{S}} \varepsilon \quad \text{and} \quad \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) \geq -\frac{8k}{\bar{S}} \varepsilon,$$

or:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) + \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) \geq -\frac{8k}{\bar{S}} \varepsilon.$$

Either way:

$$\frac{\partial \Psi}{\partial x(M_k)} \Big|_{M_k=\widetilde{M}_k} (x(M_k) - x(\widetilde{M}_k)) + \frac{\partial \Psi}{\partial y(M_k)} \Big|_{M_k=\widetilde{M}_k} (y(M_k) - y(\widetilde{M}_k)) \geq -\frac{16k}{\bar{S}} \varepsilon.$$

This completes the proof of Lemma EC.2. \square

Summary. The algorithm given in Equation (EC.34) optimizes the location and time of the first visit while always re-optimizing all subsequent ones. We have shown that this algorithm only involves SSO–VCC solutions, improves the solution at every iteration, and terminates at MSO–VCC optimum. This completes the induction. \square

EC.2.2. Computational results in the continuous space

We apply our MSO–VCC algorithm and compare it to direct CPLEX implementation (using the formulation in Equations (4)–(9)). We vary the number of customers from $K = 10$ to $K = 200$. We fix the vehicle’s origin in $(-10 \text{ km}, 0)$ and its destination in $(10 \text{ km}, 0)$. Customers’ home locations are evenly distributed on the horizontal axis, but randomly deviate on the vertical axis: $H_k = \left(-10 + \frac{20k}{K+1}, y_k\right)$, where $y_k \sim \mathcal{N}(0, \sigma^2)$. We set customers’ speed to 4 km/h, the vehicle’s speed to 20 km/h, and the customers’ maximum walking distance to 300 meters.

Table EC.1 reports the average computational times over 10 random instances. All methods consistently yield the same optimal solution, but our algorithm, even without warm start, terminates much faster than CPLEX (by a factor of 10–50, and by a factor of 2–17 even if we ignore the CPLEX setup time). The number of iterations increases supra-linearly with the number of customers, due to the back-and-forth iterations in order to guarantee optimality of the full sequence. Nonetheless, the number of iterations remains manageable even with 200 customers, leading to fast MSO–VCC computational times. Moreover, our warm-start procedure accelerates the algorithm significantly, by a factor of 3–15 for the Euclidean distance and a factor of 100–1,000 for the Manhattan distance. As we shall see, these benefits will be instrumental to solve the DAR–VCC at scale.

Table EC.1 Average performance of MSO–VCC algorithm, with and without warm start (WS), against second-order conic optimization (SOCO) in the ℓ_2 space and linear optimization (LO) in the ℓ_1 space ($\sigma = 3 \text{ km}$).

Customers	MSO–VCC (Euclidean distance)						MSO–VCC (Manhattan distance)					
	Iterations		CPU (ms)			SOCO	Iterations		CPU (ms)			LO
	Backward	Forward	Alg.	Alg. (WS)	Backward		Forward	Alg.	Alg. (WS)	LO		
10	54	44	5	<1	76		42	32	3	<1	57	
25	311	286	15	1	179		227	202	9	<1	118	
50	1,123	1,073	40	8	526		857	807	12	<1	284	
75	2,368	2,293	57	13	737		1,954	1,879	14	<1	326	
100	3,973	3,873	82	21	1,234		3,403	3,303	19	<1	496	
150	7,816	7,666	146	45	2,325		7,540	7,390	23	1	1,024	
200	13,880	13,680	311	91	5,679		13,793	13,593	52	2	1,569	

Figure EC.5 illustrates how vehicle-customer coordination can reduce the distance traveled by the vehicle: rather than going back and forth between customers’ home locations, the vehicle can visit all customers in a smoother path. Moreover, some sub-sequences enable a straight path, whereas others still require detours—albeit, smaller ones than without vehicle-customer coordination.

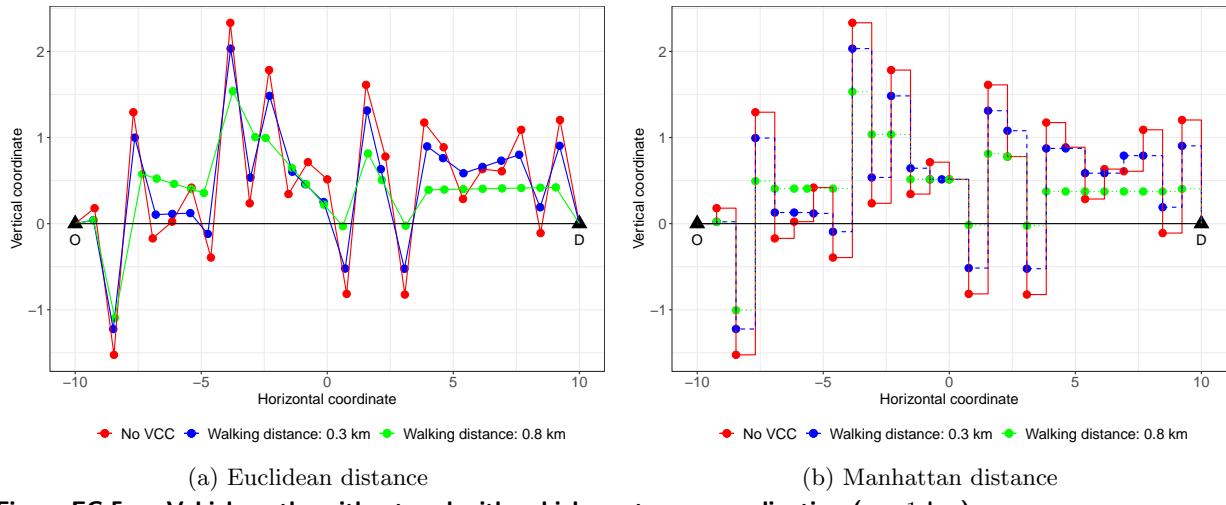


Figure EC.5 Vehicle paths without and with vehicle-customer coordination ($\sigma = 1 \text{ km}$).

EC.2.3. MSO–VCC algorithm in the discretized space

In our paper, we apply the MSO–VCC algorithm in the continuous space for the offline VRP–VCC and DAR–VCC problems (Appendix A, Section 5), and in the discretized space for the O–DAR–VCC problem (Section 6). However, Theorem 1 only guarantees global optimality in the continuous space where the norm is differentiable (e.g., in the ℓ_2 space) and in the continuous ℓ_1 space, but not in the discretized space. Before proceeding, we therefore establish the finiteness and performance of the MSO–VCC algorithm in the discretized space.

First, the MSO–VCC algorithm still terminates finitely. Indeed, at each iteration, the MSO–VCC objective is non-increasing. If the time v_k and the location M_k remain unchanged, then the MSO–VCC objective remains unchanged too. By design, the algorithm involves at most K consecutive iterations over which the MSO–VCC objective can remain constant (one for each stop). If, however, the time v_k and the location M_k are updated, then the vehicle reaches its destination D earlier, and the MSO–VCC objective strictly decreases. Since the solution space is finite in the discretized space, we conclude that the algorithm converges in a finite number of iterations.

Second, the MSO–VCC algorithm may return a local optimum in the discretized space. Figure EC.6 provides a counterexample featuring two customers each with two candidate locations. Let us assume that the algorithm starts from the home locations (either by design, or following a greedy approach picking the closest locations from the origin and the destination). At the first iteration, the algorithm first leaves the second stop unchanged (because H_1-H_2-D has length 11 but H_1-M_2-D has length 13). At the second iteration, the algorithm then leaves the first stop unchanged (because $O-H_1-H_2$ has length 10 but $O-M_1-H_2$ has length 12). Therefore, it returns the path $O-H_1-H_2-D$, of length 14 (shown in the solid green line). However, the alternative path $O-M_1-M_2-D$ is optimal, with a total length equal to 12 (shown in the dotted green line).

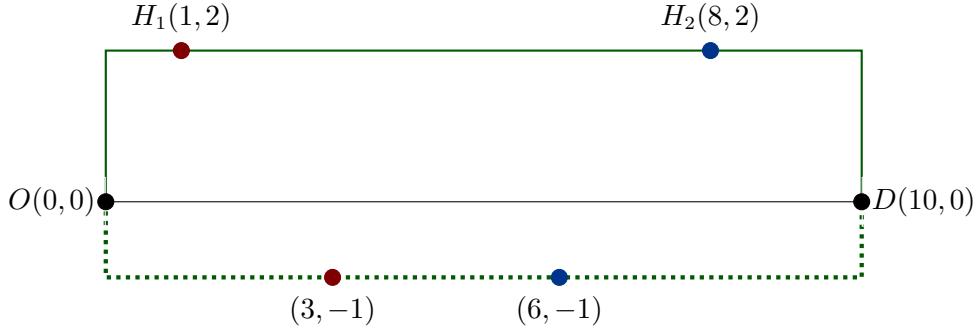


Figure EC.6 Example where the algorithm returns a local optimum in the discretized space.

Nonetheless, the algorithm consistently returns strong solutions in practice. Table EC.2 compares the solution from our algorithm (with and without warm start) to an exact MSO–VCC solution obtained via mixed-integer optimization. We use here the same experimental setup as before, and sample nine candidate stops within the 300-meter walking distance limit. Again, our MSO–VCC algorithm is much faster than direct mixed-integer optimization, by one to two orders of magnitude. Yet, it can now induce an optimality loss of 1–2%. When applied in conjunction with our warm start heuristic, however, our coordinate descent scheme actually leads to consistently optimal solutions, across all these experiments, suggesting that the warm start initialization seems instrumental to avoid local optima. Together, these results motivate our use of the MSO–VCC algorithm in Section 6 of the paper.

Table EC.2 Average MSO–VCC costs and computational times in the discretized space ($\sigma = 3$ km).

Customers	Mixed-integer optimization		Algorithm (no warm start)			Algorithm (with warm start)		
	Cost	CPU (ms)	Cost	Gap	CPU (ms)	Cost	Gap	CPU (ms)
10	2.8	76	2.8	0.0%		1	2.8	0.0% <1
25	4.8	122	4.8	0.0%		3	4.8	0.0% <1
50	9.2	326	9.4	2.2%		3	9.2	0.0% <1
75	12.9	506	13.2	2.3%		5	12.9	0.0% 1
100	17.2	855	17.6	2.3%		5	17.2	0.0% 2
150	26.2	1,450	26.7	1.9%		12	26.2	0.0% 3
200	34.7	2,366	35.3	1.7%		13	34.7	0.0% 3

Appendix EC.3: Details on VRP–VCC (Appendix A)

EC.3.1. Problem Statement

The VRP–VCC minimizes total travel times, while visiting all customers. It jointly optimizes (i) the assignment of customers to vehicles; (ii) the (ordered) sequence of customers visited by each vehicle; (iii) the location where each customer shall be visited; and (iv) the time of each customer visit.

We use the same notations as in the DAR–VCC (Section 5). In VRP–VCC, each vehicle has a capacity of \bar{Q} and each request $j \in \mathcal{P} = \{1, \dots, n\}$ has one customer. Therefore, each vehicle makes

at most \bar{Q} stops and reaches the destination by stop $\bar{Q} + 1$. We define a set $\mathcal{N} = \{1, \dots, \bar{Q}\}$ of candidate stops for each vehicle, indexed by $i \in \mathcal{N}$. By convention, $j = 0$ and $j = n + 1$ represent the vehicle's origin O and destination D , respectively. We define the following decision variables:

$$x_{ij}^r = \begin{cases} 1 & \text{if pickup } j \in \mathcal{P} \text{ is performed in stop } i \in \mathcal{N} \text{ of vehicle } r \in \mathcal{V}, \\ 0 & \text{otherwise.} \end{cases}$$

M_j : coordinates in \mathbb{R}^2 of pickup $j \in \mathcal{P}$,

M_{ri}^V : coordinates in \mathbb{R}^2 of the i^{th} stop by vehicle r , for $i \in \mathcal{N} \cup \{0, \bar{Q} + 1\}$,

v_{ri} : arrival time at the i^{th} stop of vehicle r , for $i \in \mathcal{N} \cup \{\bar{Q} + 1\}$.

We formulate VRP–VCC via mixed-integer second-order cone optimization (MISOCO) as follows, where N^W and N_j^T denote “big- M ” coefficients. This mirrors the formulation from Gambella et al. (2018), except that we impose a limit W_j on the walking distance of customer j .

$$[\text{VRP–VCC}] \quad \min \sum_{r \in \mathcal{V}} v_{r, \bar{Q}+1}, \tag{EC.54}$$

$$\text{s.t. } M_{r0}^V = O, \quad \forall r \in \mathcal{V}, \tag{EC.55}$$

$$M_{r, \bar{Q}+1}^V = \sum_{j \in \mathcal{P}} x_{1j}^r D + \left(1 - \sum_{j \in \mathcal{P}} x_{1j}^r\right) O, \quad \forall r \in \mathcal{V}, \tag{EC.56}$$

$$\|M_{ri}^V - M_j\| \leq N^W (1 - x_{ij}^r) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \tag{EC.57}$$

$$v_{ri} \geq v_{r, i-1} + \frac{\|M_{ri}^V - M_{r, i-1}^V\|}{S} \quad \forall i \in \mathcal{N} \cup \{\bar{Q} + 1\}, \forall r \in \mathcal{V}, \tag{EC.58}$$

$$v_{ri} \geq \frac{\|M_j - H_j\|}{S_j} - N_j^T (1 - x_{ij}^r) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \tag{EC.59}$$

$$\|M_j - H_j\| \leq W_j, \quad \forall j \in \mathcal{P}, \tag{EC.60}$$

$$\sum_{j \in \mathcal{P}} x_{ij}^r \geq \sum_{j \in \mathcal{P}} x_{i+1,j}^r \quad \forall i \in \mathcal{N} \setminus \{\bar{Q}\}, \forall r \in \mathcal{V}, \tag{EC.61}$$

$$\sum_{j \in \mathcal{P}} x_{ij}^r \leq 1 \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \tag{EC.62}$$

$$\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{ij}^r = 1 \quad \forall j \in \mathcal{P}, \tag{EC.63}$$

$$x_{ij}^r \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \tag{EC.64}$$

$$v_{ri} \geq 0 \quad \forall i \in \mathcal{N} \cup \{\bar{Q} + 1\}, \forall r \in \mathcal{V}, \tag{EC.65}$$

$$M_{ri}^V \in \mathbb{R}^2 \quad \forall i \in \mathcal{N} \cup \{\bar{Q} + 1\}, \forall r \in \mathcal{V}, \tag{EC.66}$$

$$M_j \in \mathbb{R}^2 \quad \forall j \in \mathcal{P}. \tag{EC.67}$$

PROPOSITION EC.1. *Let us denote by X^{\min} and X^{\max} (resp. Y^{\min} and Y^{\max}) the minimum and maximum horizontal (resp. vertical) coordinates, across the vehicle's origin, all customers' homes and the destination. The big- M coefficients in [VRP–VCC] can be set to: (i) $N^W = \|(\mathbf{X}^{\max}, \mathbf{Y}^{\max}) - (\mathbf{X}^{\min}, \mathbf{Y}^{\min})\|$, and (ii) $N_j^T = \frac{\|(\mathbf{X}^{\max}, \mathbf{Y}^{\max}) - (\mathbf{X}^{\min}, \mathbf{Y}^{\min})\|}{S_j}$ for all $j \in \mathcal{P}$.*

EC.3.2. Dynamic Programming (DP) Algorithm

Our DP involves two steps. In the first step, we explore all feasible paths that start at the origin O , visit a subset of customers, and end at the destination D . For each path, we invoke Algorithm 1 to optimize the time and location of each customer visit. For each subset of customers, we retrieve the path that minimizes the travel time. In the second step, we assign customers to vehicles to minimize total travel times, while ensuring that all customers are visited. For large-scale instances, we also apply an acceleration heuristic to circumvent the curse of dimensionality.

First step. Consider a vehicle that departs from O (stop 0), visits stops $i = 1, \dots, \bar{Q}$, and ends in D (stop $\bar{Q} + 1$). Let (j^l, Q^l, \mathbb{S}^l) denote the state variable, where j^l tracks the customer (in \mathcal{P}) visited in the “current” stop (or the origin 0, or the destination $n + 1$); Q^l tracks the number of customers in the vehicle; and $\mathbb{S}^l \subseteq \mathcal{P}$ tracks the subset of customers that have received a visit. The initial state is defined as $(j^0 = 0, Q^0 = 0, \mathbb{S}^0 = \emptyset)$.

Each state transition $(j^{l'}, Q^{l'}, \mathbb{S}^{l'}) \leftarrow (j^l, Q^l, \mathbb{S}^l)$ adds a stop to the sub-path. Such a transition is admissible if: $j^l \neq n + 1$ (the vehicle has not reached its destination); $Q^l < \bar{Q}$ or $j^{l'} = n + 1$ (the vehicle has sufficient capacity or heads to the destination); and $j^{l'} \notin \mathbb{S}^l$ (the next stop has not been visited). Each admissible transition involves the following state updates: (i) $j^{l'}$ is updated to the “next” stop; (ii) $Q^{l'} = Q^l + 1$ if $j^{l'} \in \mathcal{P}$ and $Q^{l'} = Q^l$ if $j^{l'} = n + 1$; and (iii) $\mathbb{S}^{l'} = \mathbb{S}^l \cup \{j^{l'}\}$ if $j^{l'} \in \mathcal{P}$ and $\mathbb{S}^{l'} = \mathbb{S}^l$ if $j^{l'} = n + 1$. We use a label setting algorithm to solve this DP.

Ultimately, we extract the set \mathcal{F} of all feasible paths, which collects all states such that $j^l = n + 1$ (indicating that the vehicle has reached its destination D). For each feasible path $f \in \mathcal{F}$, we record the subset of customers visited by the vehicle, denoted by $S_f = \mathbb{S}^l$. We derive the shortest travel time v_f^* by applying Algorithm 1. Then, for each customer subset \mathbb{C} , we retrieve the optimal travel time from the best path, denoted by $V^*(\mathbb{C})$, as follows:

$$V^*(\mathbb{C}) = \min\{v_f^* : f \in \mathcal{F}, S_f = \mathbb{C}\}. \quad (\text{EC.68})$$

Second step. We choose the best paths that minimize total travel times, while ensuring that each customer is in exactly one path. To this end, we design another DP algorithm. In the state variable (H^l, \mathbb{Y}^l, T^l) , H^l tracks the number of paths, \mathbb{Y}^l tracks the subset of visited customers, and T^l tracks the total travel time of all paths. The initial state is $(H^0 = 0, \mathbb{Y}^0 = \emptyset, T^0 = 0)$.

Each transition $(H'^l, \mathbb{Y}'^l, T'^l) \leftarrow (H^l, \mathbb{Y}^l, T^l)$ adds a path (i.e., a customer subset \mathbb{C}). Such a transition is admissible if: $H^l + 1 \leq |\mathcal{V}|$ (one vehicle remains available); and $\mathbb{C} \cap \mathbb{Y}^l = \emptyset$ (no customer in the new path was previously assigned to another path). Each admissible transition involves the following state updates: (i) $H'^l = H^l + 1$; (ii) $\mathbb{Y}'^l = \mathbb{Y}^l \cup \mathbb{C}$; and (iii) $T'^l = T^l + V^*(\mathbb{C})$.

We say that state l^1 dominates state l^2 if $H^{l^1} \leq H^{l^2}$, $\mathbb{Y}^{l^1} \supseteq \mathbb{Y}^{l^2}$, and $T^{l^1} \leq T^{l^2}$. Whenever a new state $(H'^l, \mathbb{Y}'^l, T'^l)$ is generated, it is immediately deleted if it is dominated.

Ultimately, we extract all the complete routings (H^l, \mathbb{Y}^l, T^l) such that $\mathbb{Y}^l = \mathcal{P}$ (indicating that all customers are visited), and return the one with the smallest value of T^l .

Acceleration heuristic. In large-scale instances, our DP algorithm may be slow to converge. We then implement a simple approximation by reducing the first-step graph. Specifically, we apply an upper bound Δ on the distance between the homes of consecutive customers, i.e., no path in \mathcal{F} will visit customers $j_1, j_2 \in \mathcal{P}$ consecutively if $\|H_{j_1} - H_{j_2}\| \geq \Delta$. This removes potentially inferior paths to speed up the overall algorithm—albeit, of course, at the cost of possible sub-optimality. We refer to our exact DP as “DP” and the heuristic approximation as “DP+H”.

EC.3.3. Additional Results

We showed in Appendix A that our algorithm outperforms the MISOCO benchmark from Gambella et al. (2018), and quantified the benefits of re-optimization to take full advantage of vehicle-customer coordination. We now establish the robustness of these findings to the customers’ maximum walking distance. We calibrate it as a factor ω times the average distance between each customer and its three nearest neighbors: $W_j = \omega \cdot \frac{1}{3n} \sum_{j \in \mathcal{P}} \sum_{l \in \mathcal{B}_j^3} \|H_j - H_l\|$, where \mathcal{B}_j^3 denotes the set of three closest customers from customer j , and n is the number of customers.

Table EC.3 reports computational results with a short and long maximum walking distance ($\omega = 1$ and $\omega = 2$). Out of the 36 instances, our exact dynamic programming algorithm solves 24 of them to optimality; the remaining 12 instances are solved with our DP+H acceleration heuristic. In comparison, CPLEX only solves 12 instances optimally; in these cases, our algorithm returns the optimal solution in faster computational times, by 12%–100%. Among the most computationally challenging instances, our DP+H algorithm significantly outperforms CPLEX, by returning a higher-quality solution in 11 out of 12 cases, improving solution quality by up to 23%, and terminating much faster (in 20–100 minutes, whereas CPLEX does not terminate within 2 hours).

Next, Table EC.4 evaluates the benefits of vehicle-customer coordination and the benefits of re-optimization, in each 18 instances and with the two walking limits. As compared to the baseline with no vehicle-customer coordination, the VRP–VCC results in 13–25% cost reductions with a short walking distance, and in 15–31% savings with a long walking distance. Moreover, as compared to the sequential benchmark, our algorithm also results in significant benefits. With the shorter walking limit, it lowers costs in 13 out of 18 instances, by up to 3.8%; and with the higher walking limit, it lowers costs in 15 out of 18 instances, by up to 8.8%. This suggests that higher operating flexibility leads to more complex operations, hence in a stronger edge of our integrated optimization approach. Note, though, that the benefits of such re-optimization are less significant for VRP–VCC than for DAR–VCC. This is because the operator is constrained to visit all customers, which

Table EC.3 Computational results for the VRP–VCC.

ω	Instance	CPLEX				Our algorithm		
		Sol.	LB	G_C^O	CPU(s)	Sol.	CPU(s)	G_C^S
1.0	p_10_3_6*	75.6	75.6	0%	208	75.6	55	0%
	p_10_4_5*	72.4	72.4	0%	438	72.4	4	0%
	p_10_5_4*	75.2	75.2	0%	634	75.2	1	0%
	p_12_3_6*	83.3	83.3	0%	1,571	83.3	176	0%
	p_12_4_5*	97.7	50.2	49%	>7,200	97.7	18	0%
	p_12_5_5*	83.5	83.5	0%	3,120	83.5	26	0%
	p_14_3_7*	93.1	93.1	0%	6,953	93.1	3,313	0%
	p_14_4_6*	99.0	61.4	38%	>7,200	98.5	-81%	0%
	p_14_5_5*	82.6	70.5	15%	>7,200	82.6	175	0%
	p_16_3_8	99.2	70.0	29%	>7,200	96.7	1,106	-3%
	p_16_4_6*	93.9	55.6	41%	>7,200	84.3	5,061	-10%
	p_16_5_6*	105.8	65.9	38%	>7,200	99.4	6,364	-6%
	p_18_3_8	153.5	64.9	58%	>7,200	131.2	5,067	-15%
	p_18_4_7	99.8	50.7	49%	>7,200	83.7	4,206	-16%
	p_18_5_6*	88.8	54.3	39%	>7,200	87.2	3,580	-2%
	p_20_3_9	134.4	44.2	67%	>7,200	102.8	5,760	-23%
	p_20_4_7	142.6	56.7	60%	>7,200	125.3	1,810	-12%
	p_20_5_6	147.5	81.7	45%	>7,200	134.6	5,102	-9%
2.0	p_10_3_6*	72.7	72.7	0%	132	72.7	71	0%
	p_10_4_5*	65.6	65.6	0%	154	65.6	9	0%
	p_10_5_4*	71.7	71.7	0%	413	71.7	6	0%
	p_12_3_6*	77.2	77.2	0%	355	77.2	287	0%
	p_12_4_5*	91.4	49.4	46%	>7,200	91.4	31	0%
	p_12_5_5*	80.1	80.1	0%	6,752	80.1	43	0%
	p_14_3_7*	85.1	85.1	0%	5,642	85.1	4,132	0%
	p_14_4_6*	98.2	77.0	22%	>7,200	98.2	1,709	0%
	p_14_5_5*	80.2	70.1	13%	>7,200	80.2	187	0%
	p_16_3_8	88.9	57.3	36%	>7,200	89.9	1,171	+1%
	p_16_4_6*	81.6	50.4	38%	>7,200	77.0	4,408	-6%
	p_16_5_6*	93.4	77.5	17%	>7,200	92.5	5,846	-1%
	p_18_3_8	123.5	79.8	35%	>7,200	119.4	4,583	-3%
	p_18_4_7	81.1	61.4	24%	>7,200	80.7	3,377	-1%
	p_18_5_6*	88.2	52.5	40%	>7,200	79.0	3,635	-10%
	p_20_3_9	111.8	47.2	58%	>7,200	92.6	5,559	-17%
	p_20_4_7	119.9	64.4	46%	>7,200	112.6	1,143	-6%
	p_20_5_6	139.4	75.0	46%	>7,200	128.2	3,850	-8%

“>7,200” means that the instance cannot be solved to optimality within two hours.

$G_C^O = (\text{CPU of our algorithm} - \text{CPU of CPLEX})/\text{CPU of CPLEX}$.

$G_C^S = (\text{Solution found by our algorithm} - \text{Lower bound found by CPLEX})/\text{CPLEX solution}$.

$G_C^S = (\text{Solution found by our algorithm} - \text{CPLEX solution})/\text{CPLEX solution}$.

“*” means that DP solves the 12 instances to optimality in our algorithm; DP+H in the other 6 instances (with $\Delta = 35$).

eliminates a degree of freedom. Viewed through this lens, the VRP–VCC is closer to the “Fix Cust. + VCC” benchmark in DAR–VCC (discussed in Section 5), with similar percent-wise benefits.

Figure EC.7 illustrates sequential and integrated VRP–VCC solutions. The solid lines in blue, grey and green show the trajectories of three vehicles; the dashed lines connect customers’ home locations. The grey trajectory visits the same sequence of customers, but achieves a smoother path

Table EC.4 Cost comparison and benefits of vehicle-customer coordination with the VRP–VCC.

Instances	No VCC	Short walking distance				Long walking distance			
		Seq.	VRP–VCC	vs. No VCC	vs. Seq.	Seq.	VRP–VCC	vs. No VCC	vs. Seq.
Cost	Cost	Cost	vs. No VCC	vs. Seq.	Cost	Cost	vs. No VCC	vs. Seq.	
p_10_3_6	88.6	75.6	75.6	-14.7%	0.0%	72.7	72.7	-18.0%	0.0%
p_10_4_5	87.6	72.4	72.4	-17.4%	0.0%	67.4	65.6	-25.1%	-2.6%
p_10_5_4	87.4	75.2	75.2	-14.0%	0.0%	74.4	71.7	-17.9%	-3.6%
p_12_3_6	104.0	83.5	83.3	-19.9%	-0.2%	78.3	77.2	-25.7%	-1.3%
p_12_4_5	123.6	97.7	97.7	-20.9%	0.0%	91.4	91.4	-26.1%	0.0%
p_12_5_5	100.5	83.5	83.5	-17.0%	0.0%	80.1	80.1	-20.3%	0.0%
p_14_3_7	123.3	96.2	93.1	-24.5%	-3.2%	89.0	85.1	-30.9%	-4.3%
p_14_4_6	115.4	99.1	98.5	-14.6%	-0.6%	98.7	98.2	-14.9%	-0.6%
p_14_5_5	94.9	85.9	82.6	-13.0%	-3.8%	85.1	80.2	-15.5%	-5.7%
p_16_3_8	123.3	96.8	96.7	-21.6%	-0.1%	92.4	89.9	-27.1%	-2.7%
p_16_4_6	107.2	84.5	84.3	-21.3%	-0.2%	78.2	77.0	-28.1%	-1.5%
p_16_5_6	119.8	100.3	99.4	-17.0%	-0.9%	93.0	92.5	-22.8%	-0.5%
p_18_3_8	162.2	133.6	131.2	-19.1%	-1.8%	124.3	119.4	-26.4%	-3.9%
p_18_4_7	96.4	85.1	83.7	-13.2%	-1.7%	81.9	80.7	-16.3%	-1.5%
p_18_5_6	100.0	87.6	87.2	-14.7%	-0.5%	80.0	79.0	-21.1%	-1.3%
p_20_3_9	124.0	104.5	102.8	-17.1%	-1.6%	98.2	92.6	-25.4%	-5.8%
p_20_4_7	152.6	127.9	125.3	-17.9%	-2.1%	123.5	112.6	-26.2%	-8.8%
p_20_5_6	160.1	135.0	134.6	-15.9%	-0.3%	130.0	128.2	-19.9%	-1.4%

Instance “p_n–m–Q” lists the number of customers n , the number of vehicles m , and the capacity of each vehicle Q .

thanks to vehicle-customer coordination. In the blue trajectory, the third and fourth customer switch between the two solutions. Moreover, one customer switches between the red and blue vehicles, which enables smoother paths in the integrated solution. These observations illustrate the three drivers of performance improvements: better stopping locations, better sequence of customers, and better vehicle-customer assignments.

Appendix EC.4: Details on DAR–VCC (Section 5)

EC.4.1. Details on mixed-integer optimization formulation

EC.4.1.1. Big-M coefficients

PROPOSITION EC.2. Let us denote by X^{\min} and X^{\max} (resp. Y^{\min} and Y^{\max}) the minimum and maximum horizontal (resp. vertical) coordinates, across all origins and destinations. Safe big-M coefficients are: (i) $N^W = \|(X^{\max}, Y^{\max}) - (X^{\min}, Y^{\min})\|$, (ii) $N_j^T = \min \left\{ T_j - \frac{\|G_{j+n} - H_j\| - W_j}{S}, \frac{W_j}{S_j} \right\}$ for all $j \in \mathcal{P}$, and (iii) $N_j^V = \max_{j' \in \mathcal{P}} \bar{T}_{j'} - \bar{T}_j$ for all $j \in \mathcal{P}$.

PROOF OF PROPOSITION EC.2. (i) The coefficient N^W in Equations (15)–(16) can be set as:

$$N^W = \|(X^{\max}, Y^{\max}) - (X^{\min}, Y^{\min})\|,$$

because for all the input locations, the horizontal coordinate is in the range of $[X^{\min}, X^{\max}]$ and the vertical coordinate is in the range of $[Y^{\min}, Y^{\max}]$.

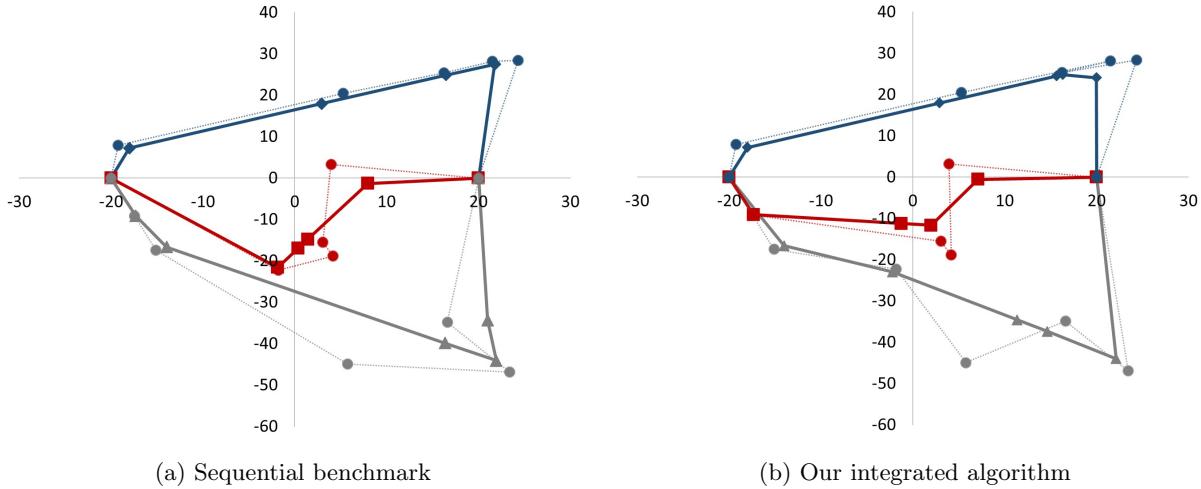


Figure EC.7 Sample solutions from the sequential and integrated algorithms.

(ii) The coefficient N_j^T for each pickup $j \in \mathcal{P}$ in Equation (18) can be set to an upper bound of the walking time of the customer(s) from H_j to M_j , i.e., an upper bound of $\frac{\|M_j - H_j\|}{S_j}$. We write:

$$N_j^T = \min \left\{ T_j - \frac{\|G_{j+n} - H_j\| - W_j}{S}, \frac{W_j}{S_j} \right\}.$$

The first part stems from the fact that the arrival time at the destination G_{j+n} is less than the deadline T_j . Therefore, the walking time is less than T_j minus the shortest possible time spent in the vehicle, minimized if customers walk W_j toward G_{j+n} . We obtain $\frac{\|M_j - H_j\|}{S_j} \leq T_j - \frac{\|G_{j+n} - H_j\| - W_j}{S}$.

The second part stems directly from the fact that $\|M_j - H_j\| \leq W_j$.

(iii) The coefficient N_j^V for each customer $j \in \mathcal{P}$ in Equation (27) can be set to the largest gap between v_i (for $i \in \mathcal{N}$) and \bar{T}_j (for $j \in \mathcal{P}$): $N_j^V = \max_{j' \in \mathcal{P}} \bar{T}_{j'} - \bar{T}_j$. \square

EC.4.1.2. Valid inequalities

Proposition EC.3 adds valid inequalities to strengthen the linear relaxation of the DAR–VCC. Equation (EC.69) shows that vehicle r must carry more passengers than q_j after conducting pickup j . Equation (EC.70) indicates that vehicle r must have q_j empty seats if it drops off q_j passengers in stop i (to satisfy the capacity constraints in stop $i - 1$), and that it must have q_j empty seats if it picks up q_j passengers in stop $i + 1$ (to satisfy the capacity constraints in stop $i + 1$). Equation (EC.71) states that if there is no pickup/dropoff in stop i of vehicle r , it returns to its destination and $Q_{ri}^P = 0$. Equation (EC.72) states that pickup j must be conducted before the corresponding dropoff $j + n$. And Equation (EC.73) lifts Equation (22) by leveraging the precedence of pickups over dropoffs. We can find examples where these constraints are not satisfied if we relax the integrality requirements, so each constraint tightens the continuous relaxation of the model.

PROPOSITION EC.3. *The following equations are valid constraints for the DAR-VCC:*

$$Q_{ri}^P \geq \sum_{j \in \mathcal{P}} q_j x_{rij} \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \quad (\text{EC.69})$$

$$Q_{ri}^P \leq \min \left\{ \bar{Q} + \sum_{j \in \mathcal{D}} q_j x_{rij}, \bar{Q} - \sum_{j \in \mathcal{P}} q_j x_{r,i+1,j} \right\} \quad \forall i \in \mathcal{N} \setminus \{2n\}, \forall r \in \mathcal{V}, \quad (\text{EC.70})$$

$$Q_{ri}^P \leq \bar{Q} \sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}, \quad (\text{EC.71})$$

$$\sum_{i'=1}^i x_{ri'j} \geq \sum_{i'=1}^i x_{r,i',j+n} \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}, \quad (\text{EC.72})$$

$$x_{rij} + \sum_{i'=1}^{i-1} x_{r,i',j+n} \leq 1 \quad \forall i \in \mathcal{N} \setminus \{1\}, \forall j \in \mathcal{P}, \forall r \in \mathcal{V}. \quad (\text{EC.73})$$

PROOF. (i) Equation (EC.69). From Equation (25), we have:

$$Q_{ri}^P = Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} + \sum_{j \in \mathcal{P}} q_j x_{rij}, \quad i \in \mathcal{N}, \forall r \in \mathcal{V}. \quad (\text{EC.74})$$

Let us now prove that $Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} \geq 0$. From Equation (21), we distinguish two cases:

- If $\sum_{j \in \mathcal{D}} x_{rij} = 0$, $Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} = Q_{r,i-1}^P$, so $Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} \geq 0$ (Equation (30)).
- If $\sum_{j \in \mathcal{D}} x_{rij} = 1$, $\sum_{j \in \mathcal{P}} x_{rij} = 0$ and $Q_{ri}^P = Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij}$, so $Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} \geq 0$ (Equation (30)).

We thus obtain $\sum_{j \in \mathcal{P}} q_j x_{rij} \leq Q_{ri}^P$.

(ii) Equation (EC.70). First, we prove

$$Q_{ri}^P \leq \bar{Q} + \sum_{j \in \mathcal{D}} q_j x_{rij} \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}. \quad (\text{EC.75})$$

From Equation (21), we distinguish two cases:

- If $\sum_{j \in \mathcal{D}} x_{rij} = 0$, then (EC.75) becomes $Q_{ri}^P \leq \bar{Q}$, that holds because of Equation (26).
- If $\sum_{j \in \mathcal{D}} x_{rij} = 1$, then:

$$\begin{aligned} Q_{ri}^P &= Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} + \sum_{j \in \mathcal{P}} q_j x_{rij} \quad \text{from Equation (25)} \\ &= Q_{r,i-1}^P + \sum_{j \in \mathcal{D}} q_j x_{rij} \quad \text{from Equation (21)} \\ &\leq \bar{Q} + \sum_{j \in \mathcal{D}} q_j x_{rij} \quad \text{from Equation (26)}. \end{aligned}$$

We proceed similarly to prove that

$$Q_{ri}^P \leq \bar{Q} - \sum_{j \in \mathcal{P}} q_j x_{r,i+1,j} \quad \forall i \in \mathcal{N} \setminus \{2n\}, \forall r \in \mathcal{V}. \quad (\text{EC.76})$$

(iii) Equation (EC.71). From Equation (21), we distinguish two cases:

- If $\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} = 1$ (i.e., a pickup or dropoff is conducted at stop i). Equation (EC.71) becomes $Q_{ri}^P \leq \bar{Q}$, which is always satisfied due to Equation (26).
- If $\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} = 0$ (no pickup or dropoff is conducted at stop i), let us prove that $Q_{ri}^P = 0$. From Equation (20), no pickup or dropoff will be conducted after stop i , i.e.:

$$\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{ri'j} = 0 \quad \forall i' \in \{i, \dots, 2n\}, \forall r \in \mathcal{V}. \quad (\text{EC.77})$$

It comes:

$$\begin{aligned} Q_{ri}^P &= \sum_{i'=1}^i \sum_{j \in \mathcal{P} \cup \mathcal{D}} q_j x_{ri'j} \quad \text{from Equation (25)} \\ &= \sum_{i'=1}^{i-1} \sum_{j \in \mathcal{P} \cup \mathcal{D}} q_j x_{ri'j} \quad \text{because } \sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{ij} = 0 \\ &= \sum_{i'=1}^{i-1} \sum_{j \in \mathcal{P}} (q_j x_{ri'j} - q_j x_{ri',j+n}) \quad \text{because } q_j = -q_{j-n} \text{ for } j \in \mathcal{D} \\ &= \sum_{j \in \mathcal{P}} q_j \left(\sum_{i'=1}^{i-1} x_{ri'j} - \sum_{i'=1}^{i-1} x_{ri',j+n} \right) \\ &= 0 \quad \text{from Equation (23) and (EC.77).} \end{aligned}$$

(iv) Equation (EC.72). From Equation (23), we distinguish two cases.

- If $\sum_{i \in \mathcal{N}} x_{rij} = \sum_{i \in \mathcal{N}} x_{r,i,j+n} = 1$ (i.e., customer request j is accepted): Since the variables x_{rij} are binary, there exists a unique i_1 such that $x_{r,i_1,j} = 1$ and there exists a unique i_2 such that $x_{r,i_2,j+n} = 1$. From Equations (22) and (24), we know that $i_1 < i_2$. We obtain:

$$\begin{aligned} \sum_{i'=1}^i x_{ri'j} &= \sum_{i'=1}^i x_{r,i',j+n} = 0 \quad \forall i \in \{1, \dots, i_1 - 1\}, \forall r \in \mathcal{V}, \\ 1 &= \sum_{i'=1}^i x_{ri'j} > \sum_{i'=1}^{i_1} x_{r,i',j+n} = 0 \quad \forall i \in \{i_1, \dots, i_2 - 1\}, \forall r \in \mathcal{V}, \\ \sum_{i'=1}^i x_{ri'j} &= \sum_{i'=1}^i x_{r,i',j+n} = 1 \quad \forall i \in \{i_2, \dots, 2n\}, \forall r \in \mathcal{V}. \end{aligned}$$

- If $\sum_{i \in \mathcal{N}} x_{rij} = \sum_{i \in \mathcal{N}} x_{r,i,j+n} = 0$ (i.e., customer request j is rejected):

$$\sum_{i'=1}^i x_{ri'j} = \sum_{i'=1}^i x_{r,i',j+n} = 0 \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}.$$

In all cases, we obtain:

$$\sum_{i'=1}^i x_{ri'j} \geq \sum_{i'=1}^i x_{r,i',j+n} \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{V}.$$

(v) Equation (EC.73). From Equation (22), we have, for all $i \in \mathcal{N} \setminus \{1\}$, $j \in \mathcal{P}$, and $r \in \mathcal{V}$:

$$\sum_{i'=1}^i x_{ri'j} \leq \sum_{i' \in \mathcal{N}} x_{ri'j} \leq 1. \quad (\text{EC.78})$$

From Equation (EC.72), we obtain:

$$x_{rij} + \sum_{i'=1}^{i-1} x_{r,i',j+n} \leq x_{rij} + \sum_{i'=1}^{i-1} x_{ri'j} = \sum_{i'=1}^i x_{ri'j} \leq 1. \quad (\text{EC.79})$$

This completes the proof. \square

We conclude by reporting the performance of CPLEX without and with the valid inequalities in Table EC.5 (using the same experimental setup as in Section 5.3). The valid inequalities reduce the computational times in 7 instances, and enable the MISOCO to reach the optimum in two extra instances. Nonetheless, they are not sufficient to scale the MISOCO to large-scale instances.

Table EC.5 Impact of valid inequalities (one vehicle, Euclidean distance).

Instances	CPLEX (without valid inequalities)					CPLEX (with valid inequalities)					Our algorithm	
	Sol.	UB	G^O	G	CPU (s)	Sol.	UB	G^O	G	CPU (s)	Sol.	CPU (s)
p_5_3_0_5	10.2	10.2	0%	0%	14	10.2	10.2	0%	0%	13	10.2	<1
p_6_3_0_3	3.8	3.8	0%	0%	56	3.8	3.8	0%	0%	14	3.8	<1
p_7_4_0_3	6.5	6.5	0%	0%	6263	6.5	6.5	0%	0%	3,382	6.5	<1
p_8_3_0_3	10.4	28.4	172%	9%	>9,000	11.4	11.4	0%	0%	601	11.4	<1
p_8_4_0_4	16.1	16.1	0%	0%	790	16.1	16.1	0%	0%	1,692	16.1	<1
p_9_5_0_5	22.7	22.7	0%	0%	5,393	22.7	22.7	0%	0%	3,643	22.7	<1
p_9_3_0_2	10.0	10.0	0%	0%	390	10.0	10.0	0%	0%	248	10.0	<1
p_10_5_0_4	7.6	27.2	256%	0%	>9,000	7.6	27.2	256%	0%	>9,000	7.6	<1
p_10_6_0_5	4.0	24.3	511%	21%	>9,000	4.8	24.3	406%	0%	>9,000	4.8	<1
p_12_4_0_3	12.4	45.4	267%	26%	>9,000	15.6	48.2	209%	0%	>9,000	15.6	<1
p_12_3_0_2	4.2	49.5	1,089%	104%	>9,000	8.5	8.5	0%	0%	1,854	8.5	<1
p_15_6_0_5	14.5	53.4	267%	73%	>9,000	5.1	53.4	954%	397%	>9,000	25.2	<1
p_15_5_0_5	4.4	53.4	1,113%	404%	>9,000	7.3	53.4	627%	202%	>9,000	22.2	<1
p_20_4_0_4	4.7	69.2	1,371%	208%	>9,000	4.7	69.2	1,371%	208%	>9,000	14.5	<1
p_20_3_0_5	4.3	62.9	1,347%	241%	>9,000	6.2	62.9	922%	140%	>9,000	14.8	<1
p_25_3_0_2	3.6	94.2	2,490%	134%	>9,000	3.8	94.4	2,379%	123%	>9,000	8.5	<1
p_25_5_0_4	9.3	94.4	918%	82%	>9,000	8.1	94.4	1,067%	108%	>9,000	16.9	<1

“>9,000” means that the instance cannot be solved to optimality within 2.5 hours.

G^O = (Upper bound – Solution)/Solution.

G = (Solution with our algorithm – Solution)/Solution.

EC.4.2. Details on the CPLEX+H heuristic for mixed-integer optimization

Let \mathcal{Q} store the served customer requests. Initially, we set $\mathcal{Q} = \emptyset$. We iterate over the n requests. At each iteration $j = 1, \dots, n$, we solve the DAR–VCC (Equations (12)–(33)) with the subset $\mathcal{Q} \cup \{j\}$. We update \mathcal{Q} with customers j such that $\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} = 1$. Once we reach the last customer request, we record the profit and repeat this process. Pseudo-code is given in Algorithm 2.

Algorithm 2 Decomposition heuristic algorithm (CPLEX+H).

```

1: Initialize the set of served customer requests  $\mathcal{Q} = \emptyset$ , profits  $\Pi^{\text{new}} = -\infty$ ,  $\Pi = 0$ 
2: while  $\Pi^{\text{new}} < \Pi$  do ▷ Stop when  $\Pi^{\text{new}}$  can no longer be improved
3:    $\Pi^{\text{new}} \leftarrow \Pi$  ▷ Record incumbent solution
4:   for  $j \leftarrow 1$  to  $n$  do
5:     if  $j \notin \mathcal{Q}$  then
6:        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{j\}$  ▷ Add customer requests one at a time
7:       Solve DAR-VCC (12)–(33), with added constraint  $\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} = 0, \forall j \notin \mathcal{Q}$ 
8:        $\mathcal{Q} \leftarrow \{j \in \mathcal{P} \mid \sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij}^* = 1\}; \Pi \leftarrow \sum_{r \in \mathcal{V}} \sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij}^* - \sum_{r \in \mathcal{V}} c \cdot v_{r,2n+1}^*$ 
9:     end if
10:    end for
11: end while

```

Output: \mathcal{Q} and Π

EC.4.3. Dynamic programming algorithm to generate subpaths in DAR–VCC

Recall that each time-space node $s \in \mathcal{S}$ is associated with a location index $L(s)$ and a time $t(s) \geq 0$. We restrict the locations to $\mathcal{D} \cup \{0, 2n + 1\}$, because vehicles can only be empty at the origin, the destination and the customers’ dropoff locations. We divide the planning horizon into discrete time intervals of length κ . The set of \mathcal{S} comprises:

- s^O : node associated with the vehicles’ origin, i.e., $L(s^O) = 0$ and $t(s^O) = 0$.
- s^D : node associated with the vehicles’ destination, i.e., $L(s^D) = 2n + 1$ and $t(s^D) = \lceil \bar{T}/\kappa \rceil \kappa$.
- For each dropoff location $j \in \mathcal{D}$, we define a node s such that $L(s) = j$ for each $t(s)$ in $\{\lfloor \underline{T}_{j-n}/\kappa \rfloor \kappa, \lfloor \underline{T}_{j-n}/\kappa \rfloor \kappa + \kappa, \dots, \lceil \bar{T}_{j-n}/\kappa \rceil \kappa\}$, where \underline{T}_{j-n} and \bar{T}_{j-n} denote the customer’s earliest and latest arrival times, respectively ($\underline{T}_{j-n} = \max\{\|G_j - O\|/\bar{S}, \|G_j - H_{j-n}\|/\bar{S}\}$).

Recall that each time-space arc connects two time-space nodes $s \in \mathcal{N}$ and $s' \in \mathcal{N}$ to indicate a trip (or subpath) where the vehicle starts empty in location $L(s)$ at time $t(s)$, picks up and drops off customers (without ever being empty), and performs its last dropoff in location $L(s')$ at time $t(s')$. We present next our label-setting dynamic programming algorithm to generate time-space arcs starting from time-space node s .

State definition. Let $(i^l, j^l, Q^l, \mathbb{S}_1^l, \mathbb{S}_2^l)$ denote a state, where i^l tracks the “current” stop, j^l tracks the “current” pickup/dropoff, Q^l tracks the number of customers in the vehicle, $\mathbb{S}_1^l \subseteq \mathcal{P}$ tracks the requests that have been picked up, and $\mathbb{S}_2^l \subseteq \mathbb{S}_1^l$ tracks “ongoing” requests that have not been dropped off. For every state, we keep track of the time as $v(i^l)$ and of the profit as $\Pi(i^l, j^l, Q^l, \mathbb{S}_1^l, \mathbb{S}_2^l)$.

Initial state: $(i^0 = 0, j^0 = L(s), Q^0 = 0, \mathbb{S}_1^0 = \emptyset, \mathbb{S}_2^0 = \emptyset)$, with $v(i^0) = t(s)$, $\Pi(i^0, j^0, Q^0, \mathbb{S}_1^0, \mathbb{S}_2^0) = 0$.

State transitions. For each pickup or dropoff $j' \in \mathcal{P} \cup \mathcal{D}$, the state is updated to $(i^{l'}, j^{l'}, Q^{l'}, \mathbb{S}_1^{l'}, \mathbb{S}_2^{l'})$, such that: (i) $i^{l'} = i^l + 1$, (ii) $j^{l'} = j'$, (iii) $Q^{l'} = Q^l + q_{j'}$, (iv) $\mathbb{S}_1^{l'} = \mathbb{S}_1^l \cup \{j'\}$ if $j' \in \mathcal{P}$, and (v) $\mathbb{S}_2^{l'} = \mathbb{S}_2^l \cup \{j'\}$ if $j' \in \mathcal{D}$ and $\mathbb{S}_2^{l'} = \mathbb{S}_2^l \setminus \{j' - n\}$ if $j' \in \mathcal{D}$. This transition is admissible if:

- $Q^l + q_{j'} \leq Q$: the vehicle has sufficient remaining capacity;
- $j' - n \in \mathbb{S}_2^l$ if $j' \in \mathcal{D}$: the vehicle can only drop off request j' if request $j' - n$ is ongoing;
- $j' \notin \mathbb{S}_1^l \cup \{L(s) - n\}$ if $j' \in \mathcal{P}$ and $s \neq s^O$ ($j' \notin \mathbb{S}_1^l$ if $j' \in \mathcal{P}$ and $s = s^O$): the vehicle can only pick up request j' if it has not been picked up yet;
- $Q^l \neq 0$: the vehicle is not empty.

Reward function. Let $\Pi(i^l, j^l, Q^l, \mathbb{S}_1^l, \mathbb{S}_2^l) = \sum_{j \in \mathbb{S}_1^l} g_j - c \cdot (v(i^l) - v(i^0))$ be the net profit of a subpath up to state l . Note that this profit cannot be evaluated in every state because pickup locations depends on the overall subpath. Therefore, each time we reach a “dropoff state” l (such that $j^l \in \mathcal{D}$), we trace back to the preceding “dropoff state” l_0 (so that $j^{l_0} \in \mathcal{D} \cup \{0\}$); and we invoke Algorithm 1 to optimize all intermediate pickup locations, starting from $G_{j^{l_0}}$ (or O , if $j^{l_0} = 0$) at time $v_{i^{l_0}}$ and ending in G_{j^l} . We then update the profit by computing $v(i^l)$.

Enforcing deadlines. If $j^l \in \mathcal{P}$, we perform a one-step feasibility check. For all ongoing requests $j \in \mathbb{S}_2^l$, we “temporarily” assume that customers in request j will be dropped off in the next stop $i^l + 1$. We derive the arrival time in G_{j+n} (i.e., $v(i^l + 1)$) with Algorithm 1. If $v(i^l + 1) > \min\{\bar{T}_j, \bar{T}\}$, state l is infeasible and can be deleted.

If $j^l \in \mathcal{D}$, we perform a multi-step feasibility check to assess whether all ongoing requests in \mathbb{S}_2^l can be dropped off by their deadlines, given that the vehicle is in location G_{j^l} at time $v(i^l)$. We solve a traveling salesman problem with time windows, by treating G_{j^l} as a “depot” and visiting all dropoff locations $\{G_{j+n}, j \in \mathbb{S}_2^l\} \cup \{D\}$ by the deadlines $\{\bar{T}_j, j \in \mathbb{S}_2^l\} \cup \{\bar{T}\}$. Again, if this problem admits no feasible solution, state l is infeasible and can be deleted.

Dominance rule. We say that state l^1 dominates state l^2 if (i) $j^{l^1} = j^{l^2} \in \mathcal{D}$, (ii) $\mathbb{S}_1^{l^1} \subseteq \mathbb{S}_1^{l^2}$, (iii) $\mathbb{S}_2^{l^1} \subseteq \mathbb{S}_2^{l^2}$, (iv) $v(i^{l^1}) \leq v(i^{l^2})$, and (v) $\Pi(i^{l^1}, j^{l^1}, Q^{l^1}, \mathbb{S}_1^{l^1}, \mathbb{S}_2^{l^1}) \geq \Pi(i^{l^2}, j^{l^2}, Q^{l^2}, \mathbb{S}_1^{l^2}, \mathbb{S}_2^{l^2})$. Conditions (ii) and (iii) indicate that $Q^{l^1} \leq Q^{l^2}$ and $i^{l^1} \leq i^{l^2}$. Any dominated state is immediately deleted.

Upon termination, we extract the end time-space node $s' \in \mathcal{S}$ such that $L(s') = j^l$ and $t(s') = \lceil v(i^l)/\kappa \rceil \kappa$, and we define a corresponding time-space arc $u = (s, s')$ in \mathcal{U}_s^+ and $\mathcal{U}_{s'}^-$. We have $g^u = \Pi(i^l, j^l, Q^l, \mathbb{S}_1^l, \mathbb{S}_2^l) = \sum_{j \in \mathbb{S}_1^l} g_j - c \cdot (v(i^l) - v(i^0))$, and $a^{u,j} = 1(j \in \mathbb{S}_1^l)$. We add a time-space arc to the destination: $u' = (s', s^D)$ with $a^{u',j} = 0$ for all $j \in \mathcal{P}$ and $g^{u'} = -c \frac{\|G_{L(s')} - D\|}{S}$.

EC.4.4. Impact of time discretization in subpath-based time-space formulation

Recall that our subpath-based formulation combines vehicle subpaths (from one “empty point” to the next) into vehicle routes (from origin to destination) via a time-space network optimization model (Equations (34)–(38)). We discuss in this appendix the impact of time discretization in this formulation. We refer here to the direct formulation as DAR–VCC–MIO (“mixed-integer optimization”), to our approach as DAR–VCC–TS–S (“time-space networks” and “subpaths”), and to baseline time-space approaches as DAR–VCC–TS–A (“time-space networks” and “arcs”).

In the absence of time windows (or deadlines), our DAR–VCC–TS–S formulation induces a loss that grows linearly with the number of customers and with the granularity of the time discretization. Therefore, the DAR–VCC–TS–S solution gets optimal in view of the DAR–VCC–MIO formulation as time discretization becomes infinitesimally granular.

THEOREM EC.1. *In the absence of deadlines (i.e., $\bar{T}_j = \infty, \forall j \in \mathcal{P}$), the gap between DAR–VCC–TS–S and DAR–VCC–MIO is bounded by $n\kappa c$, where κ is the time discretization.*

With time windows, the DAR–VCC–TS–S solution is no longer close-to-optimal due to time discretization (even without vehicle-customer coordination). For instance, consider a problem with one stop and one vehicle, a travel time of 1 unit from the vehicle’s origin to the customer’s home and of 1.1 time units from the customer’s origin to their destination. Assume also that the customer needs to reach their destination by time 2.5. The earliest dropoff time is 2.1; however, that time would be associated with a time stamp of $\lceil 2.1 \rceil = 3$, in violation of the deadline. So, the time-space formulation yields a profit of zero, inducing a loss that is no longer bounded by $\mathcal{O}(\kappa)$.

In theory, the error of time-space formulations could be alleviated via a granular discretization. For a given instance, let \mathcal{T} be the set of all possible pickup times and dropoff times. Then a discretization κ equal to the greatest common divisor of the elements in \mathcal{T} would eliminate the need to round up travel times. However, such discretization would need to be highly granular, at a price of a larger model. These challenges are further exacerbated by vehicle-customer coordination. In the discretized space (Section 6 of the paper), vehicle-customer coordination leads to a higher number of time-space nodes, requiring a more granular time discretization. In the continuous space (Section 5), the number of candidate stopping locations is uncountable, so it is not possible to define a non-zero discretization step (in time and space) that guarantees an exact time-space formulation.

That said, time discretization induces a smaller optimality loss under our DAR–VCC–TS–S approach than under baseline DAR–VCC–TS–A time-space approaches. Indeed, in DAR–VCC–TS–S, we only discretize times at dropoff locations when vehicles become empty, as opposed to discretizing times at all pickup locations and drop-off locations. For instance, for the vehicle path shown in Figure 5, the discretization in DAR–VCC–TS–S only arises at four points, whereas DAR–VCC–TS–A would involve time discretization at all 15 pick-up and drop-off locations. As a consequence, any feasible solution to DAR–VCC–TS–A can be converted to a feasible solution in DAR–VCC–TS–S with the same objective value. The converse, however, is not true. Consider the example above, assuming now a travel time of 0.5 unit from the vehicle’s origin to the customer’s home and of 1.5 time units from the customer’s origin to their destination. Then, the minimal duration of the subpath is equal to 2 units, so the time discretization does not induce any error. However, an arc-based discretization would round up travel time on both segments, resulting in a

trip time estimate of $\lceil 0.5 \rceil + \lceil 1.5 \rceil = 3$ time units. Since the customer's deadline is 2.5, the problem is solved to optimality with our subpath-based model but would not be in an arc-based model.

Proof of Theorem EC.1. Let us provide below a set partitioning formulation, referred to as DAR–VCC–SP. Although we do not use this formulation to solve the problem computationally, we use it in this proof to compare the DAR–VCC–MIO and DAR–VCC–TS formulations.

Let \mathcal{R} be the set of all feasible routes. Each route $r \in \mathcal{R}$ serves a subset of customers in \mathcal{P} along with corresponding dropoffs. Each route $r \in \mathcal{R}$ is defined via Equations (EC.80)–(EC.98), which are replicates of constraints (13)–(33) in DAR–VCC–MIO for a single vehicle.

$$M_{r0}^V = O \quad (\text{EC.80})$$

$$M_{r,2n+1}^V = \sum_{j \in \mathcal{P}} x_{r,1,j} D + \left(1 - \sum_{j \in \mathcal{P}} x_{r,1,j} \right) O, \quad (\text{EC.81})$$

$$\|M_{ri}^V - M_j\| \leq N^W (1 - x_{rij}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \quad (\text{EC.82})$$

$$\|M_{ri}^V - G_j\| \leq N^W (1 - x_{rij}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{D}, \quad (\text{EC.83})$$

$$v_{ri} \geq v_{r,i-1} + \frac{\|M_{ri}^V - M_{r,i-1}^V\|}{S} \quad \forall i \in \mathcal{N} \cup \{2n+1\}, \quad (\text{EC.84})$$

$$v_{ri} \geq \frac{\|M_j - H_j\|}{S_j} - N_j^T (1 - x_{rij}) \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P}, \quad (\text{EC.85})$$

$$\|M_j - H_j\| \leq W_j \quad \forall j \in \mathcal{P}, \quad (\text{EC.86})$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} \geq \sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{r,i+1,j} \quad \forall i \in \mathcal{N} \setminus \{2n\}, \quad (\text{EC.87})$$

$$\sum_{j \in \mathcal{P} \cup \mathcal{D}} x_{rij} \leq 1 \quad \forall i \in \mathcal{N}, \quad (\text{EC.88})$$

$$\sum_{i \in \mathcal{N}} x_{rij} \leq 1 \quad \forall j \in \mathcal{P} \cup \mathcal{D}, \quad (\text{EC.89})$$

$$\sum_{i \in \mathcal{N}} x_{rij} = \sum_{i \in \mathcal{N}} x_{r,i,j+n} \quad \forall j \in \mathcal{P}, \quad (\text{EC.90})$$

$$\sum_{i \in \mathcal{N}} ix_{rij} \leq \sum_{i \in \mathcal{N}} ix_{r,i,j+n} \quad \forall j \in \mathcal{P}, \quad (\text{EC.91})$$

$$Q_{ri}^P = Q_{r,i-1}^P + \sum_{j \in \mathcal{P} \cup \mathcal{D}} q_j x_{rij} \quad \forall i \in \mathcal{N}, \quad (\text{EC.92})$$

$$Q_{ri}^P \leq Q \quad \forall i \in \mathcal{N}, \quad (\text{EC.93})$$

$$x_{rij} \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P} \cup \mathcal{D}, \quad (\text{EC.94})$$

$$Q_{ri}^P \geq 0 \quad \forall i \in \mathcal{N}, \quad (\text{EC.95})$$

$$v_{ri} \geq 0 \quad \forall i \in \mathcal{N} \cup \{2n+1\}, \quad (\text{EC.96})$$

$$M_{ri}^V \in \mathbb{R}^2 \quad \forall i \in \mathcal{N} \cup \{2n+1\}, \quad (\text{EC.97})$$

$$M_j \in \mathbb{R}^2 \quad \forall j \in \mathcal{P}. \quad (\text{EC.98})$$

We can define the profit of each route r as $\hat{g}_r = \sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - c \cdot v_{r,2n+1}$, and define $\hat{a}_{rj} = \sum_{i \in \mathcal{N}} x_{rij}$ to indicate whether customer $j \in \mathcal{P}$ is served (note that $\hat{a}_{rj} \in \{0,1\}$ per constraint (EC.89)). The set partitioning formulation involves decision variables $\hat{z}_r \in \{0,1\}$ indicating whether route r is selected. It maximizes the operating profit subject to packing and budget constraints:

$$[\text{DAR-VCC-SP}] \quad \max \sum_{r \in \mathcal{R}} \hat{g}_r \hat{z}_r, \quad (\text{EC.99})$$

$$\text{s.t. } \sum_{r \in \mathcal{R}} \hat{a}_{rj} \hat{z}_r \leq 1 \quad \forall j \in \mathcal{P}, \quad (\text{EC.100})$$

$$\sum_{r \in \mathcal{R}} \hat{z}_r \leq |\mathcal{V}|, \quad (\text{EC.101})$$

$$\hat{z}_r \in \{0,1\}, \quad \forall r \in \mathcal{R}. \quad (\text{EC.102})$$

Proof that DAR-VCC-SP=DAR-VCC-MIO.

Let us consider a feasible solution \boldsymbol{x} to DAR-VCC-MIO, and build a feasible solution of DAR-VCC-SP with the same objective value. Recall that constraints (EC.80)–(EC.98) and constraints (13)–(33) are equivalent for each route $r \in \mathcal{R}$. We can therefore extract a set of routes in the solution, denoted by \mathcal{R}^V . Accordingly, we construct a solution for DAR-VCC-SP as:

$$\hat{z}_r = \begin{cases} 1 & \forall r \in \mathcal{R}^V \\ 0 & \forall r \in \mathcal{R} \setminus \mathcal{R}^V \end{cases}$$

- The solution $\hat{\boldsymbol{z}}$ satisfies constraint (EC.101) because $\sum_{r \in \mathcal{R}} \hat{z}_r = |\mathcal{V}|$ (by construction).
- The solution $\hat{\boldsymbol{z}}$ satisfies constraint (EC.100). Indeed, per constraints (22), each customer appears in at most one route:

$$\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} = \sum_{r \in \mathcal{R}^V} \sum_{i \in \mathcal{N}} x_{rij} = \sum_{r \in \mathcal{R}} \hat{z}_r \sum_{i \in \mathcal{N}} x_{rij} = \sum_{r \in \mathcal{R}} \hat{z}_r \hat{a}_{rj} \leq 1.$$

- The solution $\hat{\boldsymbol{z}}$ achieves the same objective value as the solution \boldsymbol{x} for DAR-VCC-MIO:

$$\sum_{r \in \mathcal{V}} \sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - \sum_{r \in \mathcal{V}} c \cdot v_{r,2n+1} = \sum_{r \in \mathcal{R}^V} \left(\sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - c \cdot v_{r,2n+1} \right) \quad (\text{EC.103})$$

$$= \sum_{r \in \mathcal{R}} \hat{z}_r \left(\sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - c \cdot v_{r,2n+1} \right) \quad (\text{EC.104})$$

$$= \sum_{r \in \mathcal{R}} \hat{z}_r \hat{g}_r \quad (\text{EC.105})$$

Let us consider a feasible solution $\hat{\boldsymbol{z}}$ to DAR-VCC-SP, and build a feasible solution of DAR-VCC-MIO with the same objective value. Let \mathcal{R}^1 be the set of routes with $\hat{z}_r = 1$. Per constraint (EC.101), we have $|\mathcal{R}^1| \leq |\mathcal{V}|$. We can assume that $|\mathcal{R}^1| = |\mathcal{V}|$, by adding empty routes to \mathcal{R}^1 if necessary. Then, for each route $r \in \mathcal{R}^1$, we assign it to only one vehicle $r \in \mathcal{V}$, and define variables x_{rij} accordingly.

- The solution \mathbf{x} satisfies constraints (23)–(33) because, by design, the solution $\hat{\mathbf{z}}$ satisfies constraints (13)–(21).
- The solution \mathbf{x} satisfies constraints (22). Indeed, for all $j \in \mathcal{P}$:

$$\sum_{r \in \mathcal{R}} \hat{a}_{rj} \hat{z}_r = \sum_{r \in \mathcal{R}^1} \hat{a}_{rj} = \sum_{r \in \mathcal{R}^1} \sum_{i \in \mathcal{N}} x_{rij} = \sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} \leq 1.$$

Then, we know from constraints (23) that $\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} = \sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{r,i,j+n}$, which implies that $\sum_{r \in \mathcal{V}} \sum_{i \in \mathcal{N}} x_{rij} \leq 1$ for all $j \in \mathcal{D}$.

- The solution \mathbf{x} achieves the same objective value as the solution $\hat{\mathbf{z}}$ for [DAR–VCC]–SP:

$$\sum_{r \in \mathcal{R}} \hat{z}_r \hat{g}_r = \sum_{r \in \mathcal{R}^1} \hat{g}_r \quad (\text{EC.106})$$

$$= \sum_{r \in \mathcal{R}^1} \left(\sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - c \cdot v_{r,2n+1} \right) \quad (\text{EC.107})$$

$$= \sum_{r \in \mathcal{V}} \sum_{j \in \mathcal{P}} \sum_{i \in \mathcal{N}} g_j x_{rij} - \sum_{r \in \mathcal{V}} c \cdot v_{r,2n+1}. \quad (\text{EC.108})$$

Proof that DAR–VCC–SP – $c \cdot n \cdot \kappa \leq \text{DAR–VCC–TS} \leq \text{DAR–VCC–SP}$

Let us consider a feasible solution $\hat{\mathbf{z}}$ to DAR–VCC–SP, and build a feasible solution of DAR–VCC–TS with an objective value within $c \cdot n \cdot \kappa$ of $\hat{\mathbf{z}}$. Let \mathcal{R}^1 be the set of routes with $\hat{z}_r = 1$. We can decompose each one into a sequence of subpaths from a point where the vehicle is empty to the next one (each route has at least one subpath since it starts and ends with a zero load). Specifically, let $\mathcal{N}_r^0 = \{i(1), \dots, i(L_r)\}$ be the set of stops with load zero in route r , i.e., $i \in \mathcal{N}_r^0 \iff Q_{ri} = 0$ and $\hat{a}_{ri} = 1$. A subpath between $i(l)$ and $i(l+1)$ in \mathcal{N}_r^0 consists of customers stored in the set $\mathcal{P}_{l,l+1}^r = \{j \in \mathcal{P} : \sum_{i' \in \{i(l), \dots, i(l+1)-1\}} x_{ri'i} = 1\}$.

For any stop $i(l) \in \mathcal{N}_r^0$, we search for the closest time-space node s_{rl} . Since the set of locations is inherently discrete, we can directly set $L(s_{rl}) = M_{r,i(l)}^V$. When it comes to time, we obtain the corresponding node by “rounding up” all times in the sequence of subpaths, that is:

$$t(s_{r,1}) = 0, \text{ and } t(s_{r,l+1}) = \left\lceil \frac{t(s_{r,l}) + v_{r,i(l+1)} - v_{r,i(l)}}{\kappa} \right\rceil \cdot \kappa, \forall l = 1, \dots, L_r - 1$$

In other words, we introduce some idleness for each vehicle whenever it reaches an “empty point”, in order to maintain feasibility of customers’ pickup locations. By construction, time discretization introduces an error in the travel time estimates of at most κ :

$$0 \leq (t(s_{r,l+1}) - t(s_{r,l})) - (v_{r,i(l+1)} - v_{r,i(l)}) \leq \kappa. \quad (\text{EC.109})$$

For each time-space node s_{rl} , we can define the subpath between stops $i(l-1)$ and $i(l)$ as the incoming arc $u(r, l-1, l) \in \mathcal{U}_{s_{rl}}^D$; similarly, we can define the subpath between stops $i(l)$ and

$i(l+1)$ as the outgoing arc $u(r, l, l+1) \in \mathcal{U}_{s_{rl}}^O$. We keep track of customer accommodations by setting $a^{u(r,l,l+1),j} = 1$ if $j \in \mathcal{P}_{l,l+1}^r$, and 0 otherwise. Finally, we define the profit as $g^{u(r,l,l+1)} = \sum_{j \in \mathcal{P}_{l,l+1}^r} g_j - c \cdot (t(s_{r,l+1}) - t(s_{r,l}))$ (for continuity, we set $g^{u_r, L_r, L_r+1} = 0$).

We construct a solution for DAR-VCC-TS as follows:

$$z^{u(r,l,l+1)} = \begin{cases} 1 & \text{for all } r \in \mathcal{R} \text{ with } \hat{z}_r = 1 \text{ and } l \in \{1, \dots, L_r\} \\ 0 & \text{otherwise} \end{cases}$$

- The solution \mathbf{z} satisfies the flow balance constraints (35). First, for each $r \in \mathcal{R}$ with $\hat{z}_r = 1$, and for each $l \in \{2, \dots, L_r - 1\}$, its corresponding time-space node s_{rl} is such that $\sum_{u \in \mathcal{U}_{s_{rl}}^O} z^u = \sum_{u \in \mathcal{U}_{s_{rl}}^D} z^u = 1$. For all other time-space nodes s , we have $\sum_{u \in \mathcal{U}_s^O} z^u = \sum_{u \in \mathcal{U}_s^D} z^u = 0$. Second, each route r starts with time-space node s^O and ends with time-space node s^D , so $\sum_{u \in \mathcal{U}_{s^O}^O} z^u - \sum_{u \in \mathcal{U}_{s^D}^D} z^u = -(\sum_{u \in \mathcal{U}_{s^D}^O} z^u - \sum_{u \in \mathcal{U}_{s^D}^D} z^u) = \sum_{r \in \mathcal{R}} \hat{z}_r = \Lambda \leq |\mathcal{V}|$.
- The solution \mathbf{z} satisfies the packing constraints (36). Indeed, for all $j \in \mathcal{P}$, we have:

$$\sum_{r \in \mathcal{R}} \hat{a}_{rj} \hat{z}_r = \sum_{r \in \mathcal{R}: \hat{z}_r=1} \hat{a}_{rj} \hat{z}_r \quad (\text{EC.110})$$

$$= \sum_{r \in \mathcal{R}: \hat{z}_r=1} \hat{z}_r \sum_{l \in \{1, \dots, L_r\}} a^{u(r,l,l+1),j} \quad (\text{EC.111})$$

$$= \sum_{r \in \mathcal{R}: \hat{z}_r=1} \sum_{l \in \{1, \dots, L_r\}} z^{u(r,l,l+1)} a^{u(r,l,l+1),j} \quad (\text{EC.112})$$

$$= \sum_{s \in \{s_{rl}, \hat{z}_r=1, l \in \{1, \dots, L_r\}\}} \sum_{u \in \mathcal{U}_s^D} a^{uj} z^u \quad (\text{EC.113})$$

$$= \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s^D} a^{uj} z^u \leq 1, \quad (\text{EC.114})$$

where the last equality also comes from the fact that $\sum_{u \in \mathcal{U}_s^D} z^u = 0$ for all other nodes s .

- The solution \mathbf{z} satisfies the budget constraints (37), due to the fact that $\sum_{r \in \mathcal{R}} \hat{z}_r \leq |\mathcal{V}|$.
- The solution \mathbf{z} achieves the following objective value:

$$\sum_{r \in \mathcal{R}} \hat{g}_r \hat{z}_r = \sum_{r \in \mathcal{R}: \hat{z}_r=1} \hat{g}_r \hat{z}_r \quad (\text{EC.115})$$

$$= \sum_{r \in \mathcal{R}: \hat{z}_r=1} \hat{z}_r \sum_{l \in \{1, \dots, L_r\}} \left(\sum_{j \in \mathcal{P}_{l,l+1}^r} g_j - c \cdot (v_{r,i(l+1)} - v_{r,i(l)}) \right) \quad (\text{EC.116})$$

$$\leq \sum_{r \in \mathcal{R}: \hat{z}_r=1} \hat{z}_r \sum_{l \in \{1, \dots, L_r\}} \left(\sum_{j \in \mathcal{P}_{l,l+1}^r} g_j - c \cdot (t(s_{r,l+1}) - t(s_{r,l})) + c \cdot \kappa \right) \quad (\text{EC.117})$$

$$\leq \sum_{r \in \mathcal{R}: \hat{z}_r=1} \sum_{l \in \{1, \dots, L_r\}} z^{u(r,l,l+1)} g^{u(r,l,l+1)} + c \cdot n \cdot \kappa \quad (\text{EC.118})$$

$$= \sum_{s \in \{s_{rl}, \hat{z}_r=1, l \in \{1, \dots, L_r\}\}} \sum_{u \in \mathcal{U}_s^D} g^u z^u + c \cdot n \cdot \kappa \quad (\text{EC.119})$$

$$= \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s^D} g^u z^u + c \cdot n \cdot \kappa. \quad (\text{EC.120})$$

where the first inequality comes from Equation (EC.109) and the second one follows from the fact that there are at most n “empty points” (one for each customer)

Let us consider a feasible solution \mathbf{z} to DAR–VCC–TS, and build a feasible solution of DAR–VCC–SP with the same objective value. Starting from the time-space node s^O , we have $\sum_{u \in \mathcal{U}_{s^O}^O} z^u = \Lambda$ outgoing arcs, which we use to define Λ activated routes. Specifically, let $\bar{\mathcal{U}}_{s^O}^O = \{u \in \mathcal{U}_{s^O}^O : z^u = 1\}$. For each $u \in \bar{\mathcal{U}}_{s^O}^O$, let $s^E(u)$ its ending time-space node. Due to flow balance constraints, there must exist $u' \in \mathcal{U}_{s^E(u)}^O$ such $z^{u'} = 1$. We follow this procedure until we reach to the time-space node s^D . Let \mathcal{R}^{TS} be the set of resulting routes. By construction, $|\mathcal{R}^{TS}| = \Lambda$. For each route $r \in \mathcal{R}^{TS}$, let \mathcal{U}_r be the set of corresponding arcs, so that $z^u = 1$ for all $u \in \mathcal{U}_r$ and $z^u = 0$ otherwise.

We construct a solution for DAR–VCC–SP by setting:

$$\hat{\mathbf{z}}_r = \begin{cases} 1 & \forall r \in \mathcal{R}^{TS} \\ 0 & \forall r \in \mathcal{R} \setminus \mathcal{R}^{TS} \end{cases}$$

- The solution $\hat{\mathbf{z}}$ satisfies the budget constraint (EC.101), since $|\mathcal{R}^{TS}| = \Lambda \leq |\mathcal{V}|$.
- The solution $\hat{\mathbf{z}}$ satisfies the packing constraints (EC.100). Indeed, for all $j \in \mathcal{P}$, we have:

$$\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s^D} a^{uj} z^u = \sum_{r \in \mathcal{R}^{TS}} \sum_{u \in \mathcal{U}_r} a^{uj} z^u = \sum_{r \in \mathcal{R}^{TS}} \hat{z}_r \sum_{u \in \mathcal{U}_r} a^{uj} = \sum_{r \in \mathcal{R}^{TS}} \hat{z}_r \hat{a}_{rj} = \sum_{r \in \mathcal{R}} \hat{z}_r \hat{a}_{rj} \leq 1.$$

- The solution $\hat{\mathbf{z}}$ achieves the following objective value:

$$\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s^D} g^u z^u = \sum_{r \in \mathcal{R}^{TS}} \sum_{u \in \mathcal{U}_r} g^u z^u = \sum_{r \in \mathcal{R}^{TS}} \hat{z}_r \sum_{u \in \mathcal{U}_r} g^u = \sum_{r \in \mathcal{R}^{TS}} \hat{z}_r \hat{g}_r = \sum_{r \in \mathcal{R}} \hat{z}_r \hat{g}_r.$$

Conclusion. We have shown that [DAR–VCC–SP] = [DAR–VCC] and that [DAR–VCC–SP] – $c \cdot n \cdot \kappa \leq$ [DAR–VCC–TS] \leq [DAR–VCC–SP]. Hence, [DAR–VCC] – $c \cdot n \cdot \kappa \leq$ [DAR–VCC–TS] \leq [DAR–VCC]. This completes the proof. \square

EC.4.5. DAR–VCC results in the Manhattan space and in the discretized space

We implement our DAR–VCC algorithm in continuous Manhattan space and in the discretized space, using the same experimental setup as in Section 5.3. In the discretized space, we sample 10 candidate stops (including each customer’s home location) within the square defined by the walking distance limit, and we use, arbitrarily, the Manhattan distance between any pair of locations (any theoretical or empirical metric could be used). Recall that, in the continuous Manhattan space, the MSO–VCC algorithm is exact (Theorem 1), and so is the DAR–VCC algorithm. In contrast, the MSO–VCC algorithm, hence the DAR–VCC algorithm, does not have guarantees of global optimality in the discretized space (see EC.2.3). Nonetheless, the results reported in Tables EC.6 and EC.7 confirm that our DAR–VCC algorithm exhibits much better scalability than

direct CPLEX implementations, yielding superior solutions in much shorter computational times—namely, CPLEX does not scale beyond 8 customers, whereas our algorithm can handle instances with up to 200 customers in minutes. These results establish the robustness of the findings from Section 5 to the structure of operations.

Table EC.6 Average computational results for DAR–VCC (3 vehicles, Manhattan distance).

Instances	CPLEX					CPLEX+H			Our algorithm	
	Sol.	UB	Opt. gap	Gap vs. alg.	CPU (s)	Sol.	Gap vs. alg.	CPU (s)	Sol.	CPU (s)
p_5_3_0.5	21.9	21.9	0%	0%	1	21.9	0%	3	21.9	<1
p_6_3_0.3	12.7	12.7	0%	0%	8	12.7	0%	5	12.7	<1
p_7_4_0.3	19.6	19.6	0%	0%	251	19.6	0%	243	19.6	<1
p_8_4_0.4	28.8	31.5	9%	0%	>3,600	28.4	1%	>3,600	28.8	<1
p_9_5_0.5	28.4	33.4	18%	2%	>3,600	26.2	11%	>3,600	29.0	<1
p_10_6_0.5	18.0	32.8	82%	0%	>3,600	18.1	0%	1,133	18.1	<1
p_12_4_0.3	27.2	51.2	88%	3%	>3,600	22.6	23%	2,134	27.9	<1
p_15_5_0.5	38.1	65.7	73%	17%	>3,600	33.8	31%	>3,600	44.4	<1
p_20_3_0.5	32.2	79.4	146%	19%	>3,600	27.0	42%	>3,600	38.4	<1
p_25_3_0.2	16.9	112.3	566%	81%	>3,600	29.0	6%	2,935	30.6	<1
p_25_5_0.4	20.6	112.3	445%	160%	>3,600	35.4	51%	>3,600	53.5	<1
p_50_4_0.2	3.7	213.1	5,721%	875%	>3,600	25.7	39%	>3,600	35.7	<1
p_50_6_0.2	-	-	-	-	-	25.7	39%	>3,600	35.7	<1
p_50_4_0.4	-	-	-	-	-	28.9	115%	>3,600	62.0	20
p_50_6_0.4	7.0	212.1	2,946%	919%	>3,600	29.0	145%	>3,600	71.0	96
p_75_6_0.4	-	-	-	-	-	31.0	124%	>3,600	69.3	86
p_75_6_0.5	-	-	-	-	-	31.0	183%	>3,600	87.6	947
p_100_4_0.2	-	-	-	-	-	31.5	62%	>3,600	51.1	7
p_100_4_0.3	-	-	-	-	-	33.2	98%	>3,600	65.9	90
p_100_4_0.4	-	-	-	-	-	33.2	125%	>3,600	74.7	857
p_125_5_0.2	-	-	-	-	-	30.9	97%	>3,600	60.8	78
p_125_5_0.4	-	-	-	-	-	27.9	193%	>3,600	81.8	1,213
p_150_4_0.3	-	-	-	-	-	31.8	112%	>3,600	67.3	256
p_150_6_0.3	-	-	-	-	-	31.7	146%	>3,600	78.1	1,089
p_175_5_0.2	-	-	-	-	-	32.8	112%	>3,600	69.6	348
p_175_4_0.4	-	-	-	-	-	27.4	179%	>3,600	76.4	2,090
p_200_3_0.3	-	-	-	-	-	32.1	80%	>3,600	57.8	325
p_200_4_0.3	-	-	-	-	-	30.7	138%	>3,600	72.9	1,939

“>3,600” means that the instance cannot be solved to optimality within one hour.

Opt. gap: (CPLEX upper bound – CPLEX solution)/CPLEX solution.

Gap vs. alg.: (Solution with our algorithm – benchmark)/benchmark, where “benchmark” refers CPLEX or CPLEX+H.

EC.4.6. Impact of walking distance

We investigate in Table EC.8 the sensitivity of the performance of our DAR–VCC algorithm with respect to the customers’ maximum walking distance W_j (from 100 meters to 900 meters). We break down the DAR–VCC computational times into the generation of the time-space network via dynamic programming (“CPU:DP”) and the solution of the network optimization (“CPU:TSNO”).

Note that the runtimes increase with the maximum walking distance. As the flexibility induced by vehicle-customer coordination increases, more time-space arcs need to be generated via dynamic programming. The time-space network optimization model, although larger, always terminates in less than one second. Yet, the overall increase remains within one order of magnitude as the walking distance increases from 100 meters to 900 meters. Recall, also, that increasing the maximum walking

Table EC.7 Computational results for DAR–VCC (3 vehicles, discretized space).

Instances	CPLEX					CPLEX+H			Our algorithm	
	Sol.	UB	Opt. gap	Gap vs. alg.	CPU (s)	Sol.	Gap vs. alg.	CPU (s)	Sol.	CPU (s)
p_5_3_0.5	23.9	23.9	0%	0%	1	23.9	0%	2	23.9	<1
p_6_3_0.3	9.8	9.8	0%	0%	4	9.8	0%	3	9.8	<1
p_7_4_0.3	19.5	19.5	0%	0%	121	19.5	0%	285	19.5	<1
p_8_4_0.4	28.0	28.0	0%	0%	1,260	28.0	0%	1,752	28.0	<1
p_9_5_0.5	32.1	36.1	12%	1%	>3,600	32.4	0%	>3,600	32.4	<1
p_10_6_0.5	13.3	25.9	95%	0%	>3,600	13.3	0%	18	13.3	<1
p_12_4_0.3	26.3	59.4	125%	0%	>3,600	23.1	14%	213	26.3	<1
p_15_5_0.5	37.4	62.6	67%	0%	>3,600	37.4	0%	>3,600	37.4	<1
p_20_3_0.5	27.8	76.5	175%	0%	>3,600	27.3	2%	>3,600	27.8	<1
p_25_3_0.2	13.7	110.1	701%	41%	>3,600	19.4	0%	66	19.4	<1
p_25_5_0.4	30.2	110.1	264%	22%	>3,600	29.2	26%	>3,600	36.7	<1
p_50_4_0.2	4.7	208.3	4,324%	438%	>3,600	21.6	18%	2,020	25.3	<1
p_50_6_0.4	14.6	208.3	1,323%	284%	>3,600	34.8	62%	>3,600	56.3	7
p_75_6_0.4	-	-	-	-	-	36.6	48%	>3,600	54.2	7
p_75_6_0.5	-	-	-	-	-	29.6	110%	>3,600	62.1	28
p_100_4_0.2	-	-	-	-	-	20.7	41%	540	29.2	<1
p_100_4_0.3	-	-	-	-	-	23.1	115%	>3,600	49.7	6
p_100_4_0.4	-	-	-	-	-	29.6	105%	>3,600	60.5	50
p_125_5_0.2	-	-	-	-	-	36.9	40%	>3,600	51.5	7
p_125_5_0.4	-	-	-	-	-	21.6	197%	>3,600	64.1	93
p_150_4_0.3	-	-	-	-	-	29.2	59%	>3,600	46.5	33
p_150_6_0.3	-	-	-	-	-	29.2	59%	>3,600	46.5	39
p_175_5_0.2	-	-	-	-	-	28.3	33%	>3,600	37.7	8
p_175_4_0.4	-	-	-	-	-	34.9	77%	>3,600	61.9	171
p_200_3_0.3	-	-	-	-	-	25.6	103%	>3,600	51.9	79
p_200_4_0.3	-	-	-	-	-	25.6	135%	>3,600	60.1	137

">3,600" means that the instance cannot be solved to optimality within one hour.

Opt. gap: (CPLEX upper bound – CPLEX solution)/CPLEX solution.

Gap vs. alg: (Solution with our algorithm – benchmark)/benchmark, where "benchmark" refers CPLEX or CPLEX+H.

Table EC.8 Computational times (in seconds) of the DAR–VCC as a function of W_j (3 vehicles).

Customers	W_j	Manhattan distance				Euclidean distance			
		Sol	CPU	CPU:DP	CPU:TSNO	Sol	CPU	CPU:DP	CPU:TSNO
100	0.1	55.2	14	13	<1	46.4	43	42	<1
	0.3	57.5	42	41	<1	49.1	117	116	<1
	0.5	61.1	68	67	<1	51.3	134	133	<1
	0.7	61.7	72	71	<1	51.3	136	135	<1
	0.9	61.7	73	72	<1	51.3	136	135	<1
150	0.1	51.6	71	70	<1	46.9	139	138	<1
	0.3	69.1	276	275	<1	53.5	411	410	<1
	0.5	79.4	486	485	<1	65.0	487	486	<1
	0.7	79.4	537	536	<1	65.0	494	493	<1
	0.9	79.4	544	543	<1	65.0	495	494	<1
200	0.1	67.6	306	305	<1	58.8	589	588	<1
	0.3	77.7	1,284	1,283	<1	67.2	1,949	1,948	<1
	0.5	81.5	2,429	2,428	<1	67.3	2,464	2,463	<1
	0.7	83.2	2,783	2,782	<1	67.3	2,507	2,506	<1
	0.9	83.2	2,823	2,822	<1	67.3	2,514	2,513	<1

distance has diminishing returns on profitability. Together, these results suggest that the algorithm retains tractability for an acceptable range of values of the maximum walking distance.

Appendix EC.5: Details on O–DAR–VCC (Section 6)

EC.5.1. Experimental setup

We implement O–DAR–VCC using real-world data from Manhattan. We construct the road network and compute the walking and driving times from OpenStreetMap (2021). We use demand data from the NYC Taxi & Limousine Commission (2021). For each trip, the dataset records the pickup time, the pickup “zone”, the dropoff time, the dropoff “zone”, the fare, and the number of riders. We filter the trips originating between 5:00 PM and 6:30 PM from December 1, 2019 to December 14, 2019. Like Bertsimas et al. (2019), we remove outliers with a duration outside of [30 seconds, 3 hours], a distance outside of [250 m, 200 km], or an average speed outside of [2 km/h, 110 km/h]. We end up with around 16,000 customer requests over the 90-minute planning horizon.

We make additional manipulations to reconstruct realistic O–DAR–VCC instances based on available information. For each trip, we uniformly sample a location within each pickup (resp. dropoff) zone as the origin (resp. destination). We assign a request time v_j^R equal to the pickup time minus a wait time uniformly sampled between 0 and 10 minutes. To cap the number of passengers per request, we duplicate a trip if it contains more than two passengers. We set the deadline \bar{T}_j by allowing a 50% increase in trip time, as compared to the ideal situation with no wait time and no detour (we vary this parameter in EC.5.4). Finally, we set the number of vehicles to 1,000, 2,000 and 3,000, with a capacity of 4 passengers per vehicle.

The profit comprises fare revenue minus vehicle costs. We use trip-level fare information reported in the data from the NYC Taxi & Limousine Commission (2021). We assume a unit vehicle cost of \$3.2 per hour, as in the offline setting. This assumes a unit vehicle cost of \$0.26 per mile including fuel, repair, maintenance, depreciation, etc. (source: www.mymoneydesign.com/personal-finance-2/savings-budgeting/what-is-the-real-cost-of-driving-per-mile/) and an average speed of 20 km/h (source: Huang et al. (2018)).

This setup follows the literature on routing optimization with the NYC taxi data (e.g., Santi et al. 2014, Alonso-Mora et al. 2017, Vazifeh et al. 2018, Bertsimas et al. 2019). In addition, to capture vehicle-customer coordination, we define the set of candidate pickup (resp. dropoff) locations as the ten closest locations from each customer’s origin (resp. destination). We do not impose extra limits on the customers’ walking distance.

EC.5.2. Algorithmic adjustments

At each epoch t , the O–DAR–VCC algorithm follows the static DAR–VCC algorithm. However, we perform several adjustments to the SSO–VCC algorithm, the generation of the time-space nodes, the generation of the time-space arcs, and the time-space network optimization model itself. In addition, we implement a heuristic acceleration to reduce the number of time-space arcs generated at each iteration—in order to consistently generate solutions within 5 seconds at each iteration. At

the end of each iteration, we also conduct a simple heuristic to support empty vehicle repositioning based on historical data, to alleviate the impact of myopic decision-making.

Additional notations

- **Road networks.** Let us define a set of (spatial) nodes \mathcal{N}^R , each corresponding to road intersections and other convenient stopping locations. For each $i_1, i_2 \in \mathcal{N}^R$, we denote by $\tau^W(i_1, i_2)$ and $\tau^V(i_1, i_2)$ the walking and driving times from i_1 to i_2 , respectively.
- **Demand inputs.** At epoch t , let \mathcal{P}_{new}^t and \mathcal{P}_{back}^t be the set of new and backlogged requests, respectively: each $j \in \mathcal{P}_{new}^t$ is such that $(t-1) \cdot \delta v < v_j^R \leq t \cdot \delta v$, and each $j \in \mathcal{P}_{back}^t$ is such that $v_j^R \leq (t-1) \cdot \delta v$. Let $\mathcal{P}^t = \mathcal{P}_{new}^t \cup \mathcal{P}_{back}^t$, and let \mathcal{D}^t store the corresponding dropoffs. For each request $j \in \mathcal{P}^t$ (resp. \mathcal{D}^t), $\widehat{\mathcal{P}}_j^t$ (resp. $\widehat{\mathcal{D}}_j^t$) denotes the candidate pickup (resp. dropoff) locations.
- **Supply inputs.** At epoch t , we denote by v_r^t the time at which vehicle $r \in \mathcal{V}$ will next be available, and by O_r^t the corresponding location. For idle vehicles $(t+1) \cdot \delta v$, v_r^t is equal to $(t+1) \cdot \delta v$ and O_r^t is defined as their location at time v_r^t . For busy vehicles, v_r^t is the time when their ongoing trip will be completed and O_r^t is the location of the trip's last dropoff.
- **Commitment.** We denote the subset of accepted customer requests at epoch t by $\overline{\mathcal{P}}^t \subseteq \mathcal{P}_{new}^t$.
- **Dispatch.** At epoch t , the set of customers on the path of a dispatched vehicle is denoted by $\widetilde{\mathcal{P}}^t \subseteq \overline{\mathcal{P}}^t \cup \mathcal{P}_{back}^t$. The next set of backlogged customer requests becomes:

$$\mathcal{P}_{back}^{t+1} = \left(\bigcup_{\tau \leq t} \overline{\mathcal{P}}^\tau \right) \setminus \left(\bigcup_{\tau \leq t} \widetilde{\mathcal{P}}^\tau \right).$$

Time-space nodes. We divide the planning horizon into discrete time intervals of length κ (10 seconds). At epoch t , the set of time-space nodes is the union of three subsets:

- \mathcal{S}_t^O : time-space nodes corresponding to vehicles' current locations. It is the collection, across all vehicles $r \in \mathcal{V}$, of the nodes defined by location O_r^t and time $t(s) = \lceil v_r^t / \kappa \rceil \kappa$.
- s^D : time-space node of vehicles' (dummy) destination and the final period $t(s^D) = \lceil \bar{T} / \kappa \rceil \kappa$.
- \mathcal{S}_t : for each dropoff $j \in \mathcal{D}^t$, we define a node s such that $L(s) = j$ for each $t(s)$ in $\{ \lfloor \underline{T}_{j-n} / \kappa \rfloor \kappa, \lfloor \underline{T}_{j-n} / \kappa \rfloor \kappa + \kappa, \dots, \lceil \bar{T}_{j-n} / \kappa \rceil \kappa \}$, where \underline{T}_{j-n} and \bar{T}_{j-n} denote the earliest and latest arrival times of customer $j - n$, respectively ($\underline{T}_{j-n} = \min_{r \in \mathcal{V}} \min_{M \in \widehat{\mathcal{P}}_{j-n}^t} \{v_r^t + \tau^V(O_r^t, M) + \tau^V(M, G_j)\}$).

Time-space arcs. We apply the label-setting dynamic programming to generate time-space arcs. We make two modifications in the algorithm to capture O–DAR–VCC characteristics:

Asynchronous operations. In the offline setting, operations are asynchronous: all customers can start walking upon placing their requests (at time 0) and vehicles depart from location $L(s)$ at time $t(s)$. In the O–DAR–VCC, however, we do not let customers start walking before the vehicle is dispatched—to allow for subsequent re-optimization. Operations are thus synchronous: customers and vehicles start their trips at time $t(s)$. Mathematically, this translates into

- Initial state: $v(i^0) = 0$ in O–DAR–VCC, as opposed to $v(i^0) = t(s)$ in DAR–VCC.
- Operating cost: $c \cdot v(i^l)$ in O–DAR–VCC, as opposed to $c \cdot (v(i^l) - v(i^0))$ in DAR–VCC. The profit is then given by $\Pi(i^l, j^l, Q^l, \mathbb{S}_1^l, \mathbb{S}_2^l) = \sum_{j \in \mathbb{S}_1^l} g_j - c \cdot v(i^l)$.

Flexible dropoffs. In the DAR–VCC, we assumed that vehicle-customer coordination only applied to pickups. In the O–DAR–VCC, we assume that both pickup locations and dropoff locations are flexible. Therefore, we apply the MSO–VCC algorithm over extended customer sequences. In the offline setting, any time the state corresponded to a dropoff, we traced it back to the previous dropoff and re-optimized, via MSO–VCC, the intermediate pickups. In the online setting, we need to “wait” until the end of the subpath to re-optimize all pickups and dropoffs. Note that this procedure leverages our restriction that the vehicle drops off the last customer at their destination. We make this adjustment to compute operating costs as part of the reward function, and to perform our feasibility check to enforce customer deadlines. In addition, since we cannot derive travel times or rewards in intermediate dropoff states, we do not apply any dominance rule.

Time-space optimization. The O–DAR–VCC exhibits two differences with the DAR–VCC. First, vehicles start from different locations O_r^t and times v_r^t . Equations (EC.123) and (EC.124) model the new “inflow” constraints. Second, backlogged customers (who have been accepted but remain unserved) will need to receive a service, which is enforced via Equation (EC.127). All other equations are unchanged. In the equations below, \mathcal{U}_t denotes the set of all time-space arcs at epoch t , and \mathcal{U}_{ts}^+ and \mathcal{U}_{ts}^- denote the subsets starting from and ending in time-space node s .

$$\max \sum_{u \in \mathcal{U}_t} g^u z^u, \quad (\text{EC.121})$$

$$\text{s.t. } \sum_{u \in \mathcal{U}_{ts}^+} z^u - \sum_{u \in \mathcal{U}_{ts}^-} z^u = 0, \quad \forall s \in \mathcal{S}_t, \quad (\text{EC.122})$$

$$\sum_{u \in \mathcal{U}_{ts}^+} z^u - \sum_{u \in \mathcal{U}_{ts}^-} z^u \leq 1, \quad \forall s \in \mathcal{S}_t^O, \quad (\text{EC.123})$$

$$\sum_{s \in \mathcal{S}_t^O} \left(\sum_{u \in \mathcal{U}_{ts}^+} z^u - \sum_{u \in \mathcal{U}_{ts}^-} z^u \right) = \Lambda, \quad (\text{EC.124})$$

$$\sum_{u \in \mathcal{U}_{ts}^+} z^u - \sum_{u \in \mathcal{U}_{ts}^-} z^u = -\Lambda, \quad \forall s \in \{s^D\}, \quad (\text{EC.125})$$

$$\sum_{u \in \mathcal{U}_t} a^{uj} z^u \leq 1, \quad \forall j \in \mathcal{P}_{new}^t, \quad (\text{EC.126})$$

$$\sum_{u \in \mathcal{U}_t} a^{uj} z^u = 1, \quad \forall j \in \mathcal{P}_{back}^t, \quad (\text{EC.127})$$

$$0 \leq \Lambda \leq |\mathcal{V}|, \quad (\text{EC.128})$$

$$z^u \in \{0, 1\}, \quad \forall u \in \mathcal{U}_t. \quad (\text{EC.129})$$

Heuristic acceleration. The scale of operations in New York City may lead to runtimes exceeding the 5-second limit at some iterations. In response, we implement a heuristic acceleration to restrict the search over a subset of “synergistic” customers in each node. The intuition is that, when we design subpaths for a vehicle located in one area (e.g., downtown Manhattan), we ignore customers traveling in distant parts of the city (e.g., uptown Manhattan). As such, our heuristic aims to enhance scalability at a small cost in terms of optimality.

Specifically, our heuristic acceleration is to consider, for each node, a subset of “backlogged” customers that are most likely to create synergies at the node under consideration. In our label-setting dynamic programming algorithm, this heuristic acceleration is implemented by restricting the set of relevant pickups and dropoffs in the admissible state transitions, as follows:

- For every node associated with a vehicle r (e.g., an idle vehicle or a vehicle with previously-assigned customers), we consider the “closest” vehicles \mathcal{V}_r , and define the set of customers including: (i) all customers previously assigned to vehicle r (empty if r was idle); (ii) all customers previously assigned to the “closest” vehicles in \mathcal{V}_r ; and (iii) all “new” customers. In other words, “synergistic” customers include all customers previously assigned to nearby vehicles. To define the “closest” vehicles, we use a function capturing the travel time between the two vehicles’ starting locations plus the difference between their earliest departing times. We control the size of the set \mathcal{V}_r to ensure that the number of “synergistic” customers ((i) and (ii) above) does not exceed a pre-specified parameter K ($K = 50$ in our experiments).
- For every node associated with a customer (that is, a dropoff point), we first find the closest vehicle and then proceed as above.

Vehicle repositioning. Given the scope of the paper, we do not fully optimize repositioning decisions or model driver behaviors. Instead, we capture first-order effects of vehicle repositioning via a simple history-based strategy orienting idle vehicles toward high-demand areas:

- Repositioning or not? Each idle vehicle will start a repositioning trip if its current location “covers” less than 0.5% of the previous customer requests. To define “coverage”, we compute the distance between the vehicle’s current location and the origin of the 1,000 previous requests. We define \bar{T} as the average distance between vehicles’ locations at the beginning of their trips, and the origins of the requests they serve. The vehicle will reposition itself if less than 5 out of the previous 1,000 requests start within \bar{T} of its current location.
- Repositioning location. For each repositioning vehicle, we consider all candidate repositioning locations within 1–2 minutes of driving, and select the one with the most customers within \bar{T} (among the previous 1,000 customers). If there is no acceptable location within 1–2 minutes, we repeat this procedure with all nodes within 2–3 minutes, 3–4 minutes, etc.

EC.5.3. Computational tractability

Recall that our O–DAR–VCC algorithm proceeds via batching and optimization, using a batching window of 10 seconds—hence, optimization has to be performed in 5 seconds at each iteration. In this appendix, we show that our algorithm, combined with our heuristic acceleration, consistently satisfies these practical requirements.

Specifically, Figure EC.8 reports the number of customers (Figure EC.8a), the number of time-space nodes (Figure EC.8b), the number of time-space arcs (Figure EC.8c), and the computational times (Figure EC.8d) over all 7,560 epochs. Each problem instance involves 50–150 customers, thousands of time-space nodes, and tens of thousands of time-space arcs (with 3,000 vehicles). Yet, the vast majority of them are solved within a second—including the times to generate the time-space network via dynamic programming and to solve the time-space network optimization. Moreover, the longest runtimes fall within the five-second limit. These results show that the O–DAR–VCC algorithm can support real-time routing operations with vehicle-customer coordination in very large-scale real-world networks, of the size of the full taxi system in Manhattan.

Finally, Table EC.9 compares the number of time-space arcs and the computational times without and with the heuristic acceleration. As expected, the heuristic acceleration significantly reduces the number of time-space arcs. Without the heuristic acceleration, around 2% of the instances take more than 5 seconds, with some instances taking up to 40 seconds—violating our five-second limit. In this instance, the total profit is \$94,873 without the heuristic and \$94,272 with the heuristic, suggesting a loss of 0.6%. Ultimately, our heuristic consistently yields solutions within the five-second threshold at minimal cost in terms of performance.

Table EC.9 Performance of O–DAR–VCC with and without heuristic (December 1, 2019, 3,000 vehicles).

O–DAR–VCC without heuristic		O–DAR–VCC with heuristic	
	Time-space arcs	CPU (ms)	Time-space arcs
Min	2,398	16	2,092
5th	7,211	109	4,644
50th	14,228	469	9,277
95th	25,429	2,111	16,649
Max	58,472	38,989	29,486

EC.5.4. Sensitivity and robustness

We conclude with an analysis of the sensitivity of our results to the maximum increase time that customers are willing to tolerate as compared to an ideal trip with no wait, no walk and no detour (set to 50% in the main paper). The results reported in Table EC.10 yield two takeaways. First, the benefits of vehicle-customer coordination are robust to delay acceptability, and are thus not driven by the choice of a 50% maximum trip increase in our paper. To see this, note that the profit

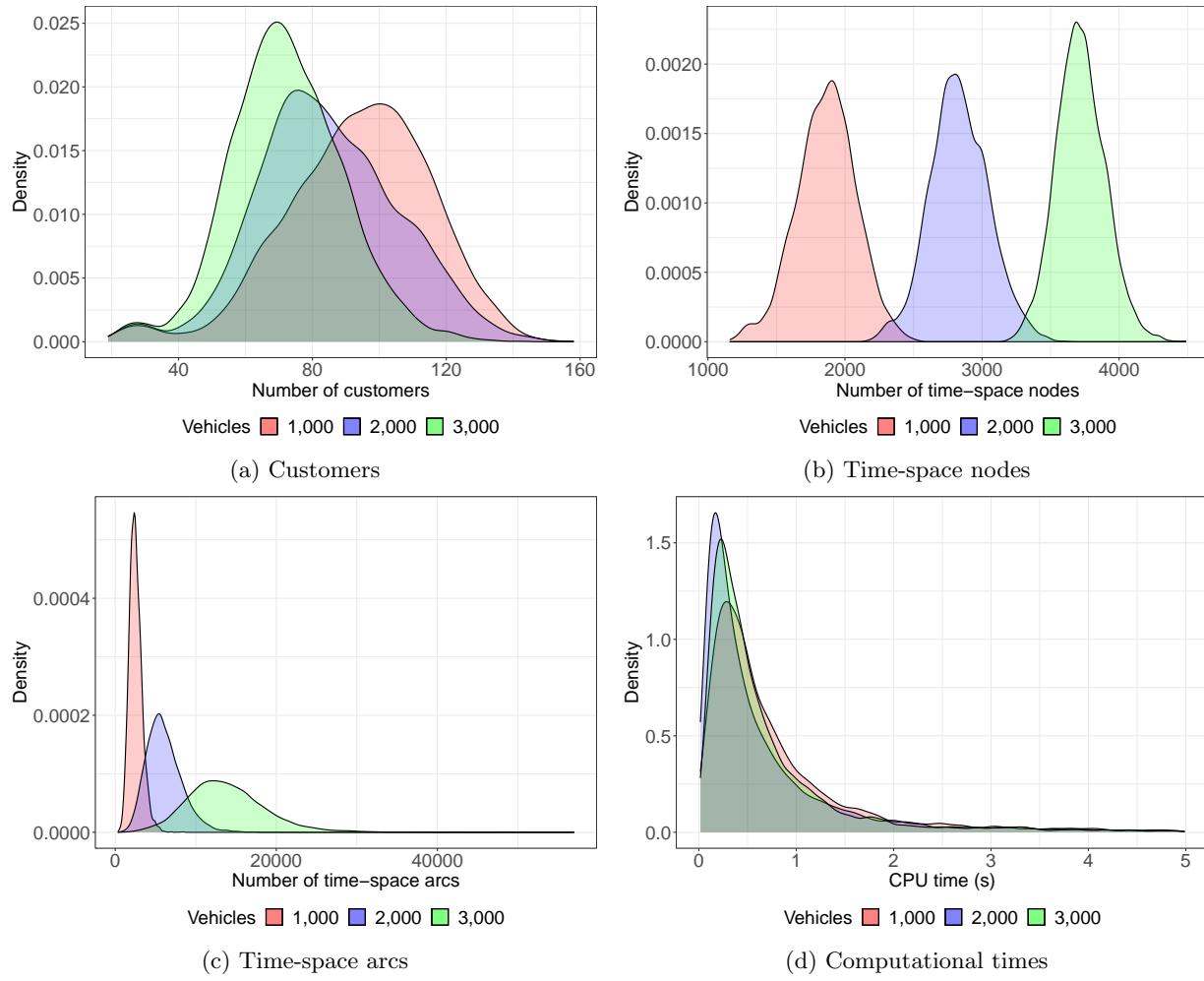


Figure EC.8 Distribution of problem size and computational times over all 7,560 epochs (14 days).

increase is positive and significant across all three instances. In fact, vehicle-customer coordination consistently leads to win-win-win outcomes: higher operating profits (by 4% to 8%), higher acceptance rate (by 2% to 6%), and less vehicle miles traveled (by 5% to 15%). Second, these benefits are driven by different mechanisms depending on the underlying level of delay acceptability. With tight deadlines, the ride-sharing operator relies almost exclusively on single-customer trips, so vehicle-customer coordination results primarily in more convenient pickup locations and shorter wait times. As a result, costs go down and the time savings enable the operator to serve more customers within their deadlines—resulting in a 30% reduction in the cost per request. With loose deadlines, the ride-sharing operator can use vehicle-customer coordination to rely more heavily on ride-pooling, freeing vehicle capacity to serve 6% more customers, resulting in a comparatively higher revenue increase. The benefits of vehicle-customer coordination are strong in both cases but “downstream” cost savings contribute more to these benefits with tight deadlines whereas “upstream” re-optimization contributes more with loose deadlines. Ultimately, these results underscore

the robustness of our main insights to the design of walking products in ride-sharing, as well as the different mechanisms at play for different levels of operating flexibility.

Table EC.10 Sensitivity to customer flexibility, measured as the maximum trip time increase from the ideal trip with no wait, no walk and no detour (December 1, 2019, 1,000 vehicles).

Time increase	Metric	No ride-pooling		Ride-pooling	
		No VCC	VCC:PU	No VCC	VCC:PU-DO
25%	Profit	\$47,300	\$51,440	\$45,778	\$49,519
	Profit increase	(base)	8.8%	(base)	8.2%
	Revenue increase	(base)	7.7%	(base)	6.3%
	Cost decrease	(base)	11.1%	(base)	23.1%
	Cost per request	\$0.68	\$0.54	\$0.74	\$0.52
	Downstream contribution	(base)	13.1%	(base)	22.9%
	Upstream contribution	(base)	86.9%	(base)	77.1%
	Acceptance rate	33%	37%	32%	36%
	VCC: Pickups away from origin	—	14%	—	15%
	VCC: Dropoffs away from destination	—	—	—	0%
	Pooled trips	—	—	2%	4%
	Vehicle miles traveled (total)	21,599	18,205	21,599	18,205
50%	Profit	\$63,956	\$65,718	\$68,998	\$71,688
	Profit increase	(base)	2.8%	(base)	3.9%
	Revenue increase	(base)	2.5%	(base)	3.5%
	Cost decrease	(base)	3.7%	(base)	7.5%
	Cost per request	\$0.45	\$0.41	\$0.38	\$0.33
	Downstream contribution	(base)	11.5%	(base)	11.3%
	Upstream contribution	(base)	88.5%	(base)	88.7%
	Acceptance rate	49%	52%	55%	57%
	VCC: Pickups away from origin	—	17%	—	19%
	VCC: Dropoffs away from destination	—	—	—	3%
	Pooled trips	—	—	15%	20%
	Vehicle miles traveled (total)	22,055	21,248	20,783	19,270
75%	Profit	\$74,069	\$78,682	\$78,992	\$85,641
	Profit increase	(base)	6.2%	(base)	8.4%
	Revenue increase	(base)	6.0%	(base)	8.0%
	Cost decrease	(base)	-0.4%	(base)	6.3%
	Cost per request	\$0.40	\$0.37	\$0.31	\$0.26
	Downstream contribution	(base)	4.1%	(base)	5.2%
	Upstream contribution	(base)	95.9%	(base)	94.8%
	Acceptance rate	60%	65%	65%	71%
	VCC: Pickups away from origin	—	18%	—	22%
	VCC: Dropoffs away from destination	—	—	—	7%
	Pooled trips	—	—	29%	37%
	Vehicle miles traveled (total)	23,975	24,059	19,986	18,876