# An Online Cost Allocation Model for Horizontal Supply Chains

Final Report
METRANS Project

April, 2018


**Principal Investigator:**
**John G. Carlsson**
**Co-Principal Investigator:**
**Maged M. Dessouky**

**Ph.D. Graduate Students:**
**Han Zou**
**Shichun Hu**

**Daniel J. Epstein Department of Industrial and Systems Engineering**
**University of Southern California**
**Los Angeles, California**


METRANS Transportation Center USC | CSULB

# Disclaimer

# Abstract

The problem we study in this report focuses on routing in real time a fleet of capacitated vehicles to satisfy requests submitted by a set of customers while assigning the service cost fairly among the requested customers. During each operation, only a subset of the customers request service with some of them known at the beginning of the operation and the rest arriving dynamically during the day. The exact time points of these dynamic requests are unknown at the beginning of the day.

We propose a Hybrid Proportional Online Cost Sharing (HPOCS) mechanism to tackle the cost sharing problem and analyze its performance using simulation instances. Although HPOCS does satisfy the desirable properties, namely online fairness, budget balance, immediate response, individual rationality and ex-post incentive compatibility, it has certain drawbacks when the number of dynamic customers is small and does not give sufficient incentive for customers to request early. Therefore, we make two extensions to HPOCS: 1) we extend it to introduce the idea of discounts to encourage customers to submit their request in advance to better facilitate efficient vehicle routing; 2) we extend it to incorporate a dynamic vehicle routing framework that periodically re-optimizes the current vehicle routes. Both extensions include performance analysis and the tradeoff between the performance and the loss of certain desirable properties.

In general, our proposed mechanism, along with its extensions can generate efficient cost sharing solutions that satisfy desirable properties, reduce overall operating cost (mainly vehicle miles travelled) and provide sufficient incentives to customers to request service early in support of horizontal cooperation.

# Disclosure

The project entitled "An Online Cost Allocation Model for Horizontal Supply Chains" was funded in entirety under this contract to the California Department of Transportation. The Principal investigator and the Co-Principal investigator of the project were Associate Professor John G. Carlsson and Professor Maged M. Dessouky of the Viterbi School of Engineering at the University of Southern California and was carried out during the period: 01/01/2017 – 03/31/2018. The total amount of funding was $100,000.

# Acknowledgement

We would like to thank METRANS for funding this research.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Background

Many industries deal with the task of transporting goods or delivering services in a timely, reliable, and cost-effective manner, including manufacturing, food, e-commerce, public transit, etc. Logistics has become the backbone that enables the productivity and mobility of these industries [38]. Indeed, growth in the transportation sector recently has been on par with the growth in the Gross Domestic Product (GDP) in the United States. According to statistics from the 2013 National Transportation Statistics report [39], expenditure on transportation activities amounted to 1,426 billion dollars in 2012, representing nearly 9 percent of the total US GDP. However, the logistics sector as it is today functions in a way that is economically, environmentally, and socially unsustainable [37]. In order to compete effectively against their peers, companies have relied on internal optimization to reduce operating costs, but have overlooked opportunities for external cooperation. As a result, the logistics sector has become highly fragmented, with each supplier developing and operating its own distribution network that sees low capacity usage, high energy consumption, and high greenhouse gas emission across the entire system [37]. The increasing amount of freight transportation also aggravates its impact on traffic congestion, and poses threats on the safety and efficiency of passenger traffic and other social functions that share the same road infrastructure. This phenomenon becomes more significant in densely populated urban areas, like Los Angeles.

As opportunities for internal optimization are becoming fully exploited, fierce competition drives companies to focus on reducing costs of non-value adding activities [50], especially logistic activities. The concept of horizontal cooperation sees both theoretical development in the literature [8] and successful application in industry [20]. It formally refers to the cooperation between businesses operating at the same level(s) in the market. When applying to the logistics sector, horizontal cooperation could refer to the pooling of freight transportation networks and sharing of customers. External cooperation allows consolidation of vehicle capacity, delivery routes, and shipment orders among different suppliers or logistic service providers, thus creating a unified logistics network that sees increased capacity usage, reduced energy usage, pollution, and operating costs. For example, a case study of the Swedish forest industry has shown that potential savings of cooperation among several forest companies operating in the same region are large, often in the range of 5 to 15 percent [20]. A shared transportation network also reduces the total truck miles, which in turn reduces the usage of the road infrastructure that it shares with passenger traffic. Similarly, reduced freight traffic helps alleviate traffic congestion and the safety threat it poses on passenger traffic. Horizontal cooperation would not only generate savings for companies already in business, but also lower the potential barrier for new (and possibly small) businesses to enter the market.

Besides, operations in any real world transportation network contain a fairly high level of uncertainties including variable waiting and travel times due to traffic congestion, arrival of new service requests, cancellation of existing requests, unknown demand sizes, etc. Under changing and gradually revealed information, the problem of designing real-time collection and/or delivery routes from one or several depots to a set of geographically dispersed customers falls in the scope of the Dynamic Vehicle Routing Problem (DVRP). The DVRP derives from the Vehicle Routing Problem (VRP) when some element of the problem becomes non-deterministic. Given the advances in information technologies, the transportation industry, like many others, has undergone significant changes in the last decade. In particular, the increasing performance and the lowering cost of the computational devices, vehicle positioning systems, real-time information and communication networks have made real-time dynamic routing of the fleet a very real possibility.

One crucial component of a shared transportation system is the method used to allocate costs and/or savings to each participant in the system. A cost-sharing mechanism serves as the basis for any economic analysis of horizontal cooperation. However, the cost allocation problem in the vehicle routing context remains rarely studied in the literature, especially for the dynamic case discussed above. For a "static" cost sharing problem in which the set of players and the cost function are both known and deterministic, Moulin mechanisms [40] and acyclic mechanisms [35] are among the most studied families of cost-sharing mechanisms. In the context of vehicle routing problems, a "static" cost sharing problem means that the set of customers to be served is known and the optimal total cost can be calculated. Unfortunately, neither of these two assumptions holds in the dynamic vehicle routing problem we study.

Little work has been conducted on designing online cost-sharing mechanisms that work when the set of players are gradually revealed, instead of known beforehand. Even less work on cost allocation has been done in the vehicle routing context. The majority of this subset of work has assumed a static operating environment, in which the tasks of designing vehicle routes and allocating costs can be tackled separately and independently. This separation does not fully justify challenges faced by logistic service providers under today's lean manufacturing and JIT delivery constraints which partially motivates the problem we study. Thus, there is a need for a unified solution approach that combines dynamic vehicle routing with online cost allocation for dynamic cost sharing transportation systems.

## 1.2 Problem Description

In this research, we aim to model and solve a cost allocation problem in a real-time cost sharing transportation system, which results from horizontal cooperation among multiple suppliers. The underlying vehicle routing problem represents the daily operation of many logistic service providers, especially those who consolidate shipments from multiple suppliers. Suppose a trucking company operates a fleet of homogeneous vehicles to collect shipments from a known set of suppliers and transport the shipments back to a central depot. These suppliers can be seen as registered customers of the company. Their locations and service time windows are known and fixed. However, each customer may not request service on each day. How often each customer requests service is determined by his/her own operation schedules, and can be seen as a given parameter in our problem. If a customer requests service, it can either do so at the beginning of the day (before the vehicles leave the depot), or at any time during the day. Customers who have requested service at the beginning of the day are called advance customers and must be serviced. All other customers, called dynamic customers, may potentially request service, but the company does not know whether and when they will do so. In our previous research, we have modeled and solved this dynamic vehicle routing problem using a novel look-ahead dynamic vehicle routing framework [13].

Building upon our previous work, this research focuses on how to allocate the cost to each new customer at the time of request without knowing the future customer requests and the total cost of the service. It is important to point out that the problem of dynamically routing vehicles and the problem of real-time cost allocation are highly interdependent and must be considered simultaneously. In particular, the vehicle routes depend on whether the new customers accept or decline the quote for service, and the quote (cost share) in turn depends on how vehicle routes are designed and what is the expected total cost of such routes.

## 1.3 Motivation

In a dynamic operation environment, critical problem information is revealed over time, meaning that the complete realization of the problem is only known at the end of the planning horizon. The optimal total cost of service can only be approximated at any time during the planning horizon due to incomplete information. Thus, the vehicle routing problem and the cost allocation problem become highly interdependent. The routing schedule depends on whether new customers can be accommodated due to feasibility constraints, and whether the customers accept or decline the service based on the price quotes and their willingness-to-pay values. Reversely, the price quote offered to each dynamic customer must consider the expected total operating cost, which in turn depends on the routing schedule. The price quote should also consider possible future customer requests and the ability of the current schedule to accommodate them.

One of the most heavily studied and widely used approaches to tackle the cost-sharing problem is Cooperative Game Theory (CGT). CGT provides a general framework for studying the allocation of costs and/or savings to a group of participants who require a common resource to accomplish tasks. Traditional CGT-based solution concepts require that the grand coalition (the entire set of players) is known in advance and that the optimal cost function is well-defined on each coalition/subset of players. Neither of these two assumptions hold in the dynamic environment. The majority of work in the current literature have assumed a static vehicle routing environment, and have treated the cost allocation problem separately from the vehicle routing problem. For example, Lozano et al. [34] studied the problem of finding the optimal way to form coalitions and to share the total cost among a set of companies who are interested in consolidating their transportation needs. In particular, a Mixed-Integer Linear Program (MILP) was formulated to estimate the optimal total cost of serving a set of companies. The problem was first solved considering the transportation demands of each company independently. A second problem was formulated by merging the transportation demands of every coalition of two of the companies. Then the model considered the coalitions of three companies, and so on, until reaching the grand coalition. Given the estimated total cost of serving each subset/coalition of the companies, several cost allocation methods based on CGT concepts were then implemented and compared. Similar two-stage approaches that decouple the cost allocation problem from the vehicle routing problem are common in the literature.

It can be easily shown that typical cost-sharing mechanisms such as proportional cost sharing and marginal cost sharing fail to possess desired properties when adapted naively to the dynamic setting. Indeed, the problem of allocating costs in a real-time cost sharing transportation system is highly nontrivial and is ranked among the top impediments for successful horizontal cooperation [9]. The research on designing online and dynamic cost-sharing mechanisms for transportation systems have been very limited. A major line of research considering the competitive pricing problem in a dynamic transportation system is due to Figliozzi, Mahmassani, and Jaillet [17, 19, 18]. The problem is framed as a sequential auction marketplace where new customer orders arrive stochastically and the logistics service provider must offer a competitive price bid to win the order from its competitors. New orders arrive at the same time when existing orders are being served. Each order served generates a reward. The objective is to maximize the profit as measured by the total rewards collected minus the total transportation cost. The authors developed a stochastic dynamic programming-based formulation that solves for the optimal price whenever a new order arrives.

The work by Furuhata et al. [21] is concerned with a demand-responsive transport (DRT) system where new service requests are submitted sequentially over time, but all of them are still submitted before the vehicles start service. The authors developed a mathematically precise and concrete cost-sharing mechanism, namely the Proportional Online Cost Sharing (POCS), that handles sequential customer submissions. POCS draws upon features of proportional and marginal cost sharing and has been proved to satisfy a list of desirable properties, including online fairness, immediate response, individual rationality, budget balance, and ex-post incentive compatibility. POCS is a flexible framework in the sense that no specific cost function is defined. All of the desired properties hold as long as the cost function of choice satisfies the following two properties: 1) total cost is non-decreasing over time (over order submissions); 2) total cost is independent of the submit order of customers who have already submitted their requests.

Although POCS represents a step forward in the research on cost-sharing mechanism design because it relaxes the constraint that the entire set of players must be known at once, limitations still remain. POCS assumes that all customers submit their service requests before vehicle operations start. In the dynamic vehicle routing environment we study, customer request submissions and vehicle operations take place simultaneously. The second assumption that the total cost is independent of the submit order of customers does not hold trivially.

In this research, we focus on studying a category of the dynamic vehicle routing problem where only part of the customers are known in advance, and the rest become known in real time. Based on a dynamic routing framework, we develop an online cost-sharing mechanism that is capable of dynamically allocating cost to each customer as it is realized. Our approach combines two cost-sharing mechanisms originally designed for the static and the online environment, respectively. With specially designed cost functions and routing schedules, the hybrid mechanism is shown to possess all of the five properties originally proposed in [21], namely online fairness, immediate response, individual rationality, budget balance, and ex-post incentive compatibility. We extend our work by proposing several variations of the baseline mechanism which can be formulated by relaxing some of the model assumptions. We compare and contrast different variations of the mechanism through extensive numerical simulations.

### 1.4 Structure of the Report

The rest of the report is organized as follows. In Section 2, a literature review of the relevant problems is presented. Section 3 formally defines the problem and describes the dynamic routing framework. In Section 4, we introduce the Hybrid Proportional Online Cost Sharing (HPOCS) mechanism and illustrate it with two examples. We prove that HPOCS satisfies all of the desirable properties we propose. We then analyze simulation results under various demand conditions. Section 5 and Section 6 present two extensions of the HPOCS mechanism that improve the performance of the baseline model. We study the two extension mechanisms both theoretically and via experiments. Section 7 describes how to implement the proposed mechanism and we conclude in Section 8.

# 2 Literature Review

In this section, we review the literature relevant to our research. We first focus on previous work on cost-sharing mechanism design, then review studies on cost allocation problems in the domain of transportation.

## 2.1 Cost-sharing Mechanism Design

A cost allocation problem specifies a set of players who request service that require a common and limited resource. Each player has a private, non-negative valuation for the service. This valuation is sometimes referred to as the willingness-to-pay value or the bid of the player. A cost function is defined on all subsets of players. The value of the function usually denotes the minimum total cost of serving the corresponding subset of players. The objective is to determine the cost allocated (or the price charged) to each player and the subset of players who are willing to participate in the contention given the prices. The final solution needs to not only specify the membership of the contention, but also provide exact ways to facilitate such a contention in the context of the problem [35]. For example, to solve the cost allocation problem corresponding to a vehicle routing problem, the final solution needs to specify the group of customers to participate in the cooperation, a routing schedule that accommodates the same group of customers, and the exact cost share for each customer in the group. Depending on the context of the problem, a "binary demand game" refers to a situation where each player either receives service fully or not receive any service at all. In general demand games (sometimes referred to as "multi-parameter demand games"), each player may receive one of several levels of service or receive no service at all. Both binary and general demand games are common in vehicle routing problems. For example, if the operation is concerned with the shipments of goods or supplies, partial shipments or refills may be acceptable. In other situations, including the case of demand-responsive transport (DRT) systems, the entire demand of one customer must be fully fulfilled or not fulfilled at all. Popular classes of cost sharing problems include facility location problems [46, 30, 14], set covering problems [14, 27], and network planning problems (including the Steiner tree (ST) problem) [25, 46, 26].

One of the most heavily studied approaches to tackle the cost allocation problem is Cooperative Game Theory (CGT). CGT provides a general framework for studying the allocation of costs and/or savings to a group of participants who require a common resource to accomplish tasks. This approach focuses on what a group can achieve and whether it is possible to coordinate the group to achieve the goal by properly allocating costs. Many solution concepts have been proposed within the CGT framework. For example, the core [22] of a problem consists of allocations that recover the cost incurred by all of the players and ensures that no individual or a group of players can benefit by defecting. Whether the core of a problem is empty or not is often used as a proxy for the possibility of cooperation. Other CGT related solution concepts include the Shapley value [49], the nucleolus [48], and the $\tau-$value methods [53].

Another approach for solving the cost allocation problem is to design a cost-sharing mechanism, which is the approach we adopt in this report. Instead of investigating what can be achieved by a cooperation, cost-sharing mechanism design focuses on finding a good way to allocate the cost to all potential players and to incentivize all players to participate in the cooperation. A cost-sharing mechanism needs to define an algorithm to calculate the shared cost for each player, and a process to determine the subset of players who end up participating in the cooperation. During this process, the algorithm compares the shared cost of each player with its willingness-to-pay value; only the players whose quotes are no larger than their willingness-to-pay values accept the quotes and receive service.

Researchers have focused on studying three desired properties of cost-sharing mechanisms, namely truthfulness (strategyproofness), budget balance, and economic efficiency [40, 35]. Truthfulness (strategyproofness) requires that no player can strictly increase its utility by misreporting its valuation for the service. Equivalently speaking, it is optimal for individual players or groups of players to make their decisions based on their true valuations. The budget balance property requires that the sum of the prices charged to each participant equals to the total operating cost of facilitating the cooperation. Economically efficient mechanisms are those maximizing the welfare of all players in the problem, not only those who end up participating in the contention [35]. Unfortunately, no mechanism could simultaneously satisfy all of the above mentioned constraints, as has been proved by Green et al. and Roberts [24, 44]. Thus researchers have focused on developing cost-sharing techniques that relax at least one of the constraints. Approximate measures have also been proposed on budget balance and economic efficiency [47].

The only known general technique for designing truthful and approximately budget-balanced cost-sharing mechanisms is due to Moulin [40, 41]. Roughly, a Moulin mechanism simulates an ascending iterative auction, in which players receive non-descending prices in each iteration and only the players who accept the price remain in the game. The algorithm halts when all remaining players accept their prices offered in the current iteration. A function regarded as the cost-sharing method calculates the cost share for each player given the entire set of players remaining in the current iteration. Truthfulness is guaranteed by requiring that the underlying cost-sharing method (function) satisfies the cross-monotonic property. Budget-balance is achieved only approximately by offering costs in each iteration that would in total approximately cover the cost incurred if the current iteration were to be the last one. Despite the fact that designing such mechanisms is highly non-trivial, Moulin mechanisms have gained significant attention and seen applications in a wide range of cost-sharing problems including scheduling [5, 4], network design [2, 25, 26], facility location [14, 28, 30, 43], and logistics [32]. However, recent work in the literature have criticized their poor performance in terms of budget-balance and economic efficiency [35, 27, 45].

New families of cost-sharing mechanisms have been proposed, among which is the acyclic mechanism [35]. Different from Moulin mechanism, acyclic mechanism introduces an ordering of players. In each iteration of the algorithm, prices are offered to players in the predefined sequence rather than simultaneously. This design relaxes the cross-monotonicity requirement of the underlying cost-sharing method while still maintaining the truthfulness property of the mechanism. Acyclic mechanisms have seen applications mainly in scheduling [3, 6]

All of the approaches discussed above have been proposed for static cost allocation problems, in which all the problem information is known and deterministic. The entire set of players is known and fixed, and the cost function defined on any subsets of the players can be calculated deterministically. In the cost allocation problem associated with the dynamic vehicle routing problem, critical problem information is revealed dynamically over time. For example, in the vehicle routing problem with dynamic customers, only part of the customers is known at the beginning of the planning horizon, and the rest of the customers arrive dynamically over time. In such cases, the entire set of customers that become realized can only be known by the end of the planning horizon. At any point during the horizon, the total cost of serving all customers can only be approximated. Generally speaking, the problem of allocating cost to a set of players under dynamically revealed information is called an online cost allocation problem, which can be solved by an online cost-sharing mechanism. An online mechanism adapts to newly revealed problem information and iteratively resolves the cost allocation problem as necessary.

An online environment brings new challenges to the design of cost-sharing mechanisms, and additional properties that are important in an online environment have been introduced [21]. In particular, the individual rationality property states that the shared cost value for any customer never exceeds its willingness-to-pay level once the customer has been accepted into the cooperation.

The online fairness property requires that players who join the cooperation late should never receive a lower shared cost than those that join early. The immediate response property requires that when a new player becomes realized, it should be provided with an initial quote for the service immediately, so that the player could make the decision on whether to participate in the cooperation or not. The quotes have to be offered without knowledge on future player realizations and the final total cost of service. Besides, some of the properties originally defined for static cost allocation problems have been extended to the online environment. For example, the ex-post incentive compatibility property builds on the truthfulness property for static problems, and states that the optimal strategy for each player is to make its request known at the earliest time possible.

## 2.2 Cost Allocation in Transportation

As transportation costs continue to increase due to increased competition, lower inventory levels, and higher service level requirement by customers, horizontal collaboration in the logistics sector has received increasing attention from both the research community and players in industry. In the context of supply chain management and transportation, horizontal cooperation refers to the pooling of transportation capacity and customer demands among businesses operating at the same level(s) in the market [8]. A cost sharing transportation system is formed as a result. One crucial component of such a system is the allocation of total operating costs and/or savings to each participant in the system. A cost-sharing mechanism serves as the basis for any economic analysis of horizontal cooperation.

The work by Anderson and Claus represents one of the earliest attempts to study the cost allocation problem in transportation collaboration [1]. The authors studied and compared multiple basic cost allocation methods as applied to a minimum cost network problem. In particular, the authors showed that the average cost sharing, unit (per mileage) cost sharing, and marginal cost sharing all suffer from various inefficiencies when applied naively. For example, average cost sharing cannot guarantee that each rational player will participate in the cooperation, while unit mileage pricing cannot prevent subgroups of users to form coalitions outside the grand coalition.

CGT appears to be one of the popular approaches for solving cost allocation problems in transportation research. Many CGT solution concepts have been studied, including the Shapley value [49, 29], the core and related concepts [23, 15, 16], the nucleolus [48, 33], and the $\tau-$value methods [53].

Other streams of research exist that study the cost allocation problem in transportation outside the scope of CGT. Özener et al. studied the cost allocation problem in a vendor managed inventory (VMI) system [42]. In a VMI model, the supplier is responsible to manage the inventory level of its customers and decides when and how much to replenish each customer. The research focuses on how to calculate the cost-to-serve for each customer, which is useful for both marketing and distribution planning needs.

Lewczuk and Wasiak studied the problem of how to the allocate the transportation cost to clients served by a material delivery system [31]. For practical reasons, a transparent and easy-to-understand cost allocation scheme is desired. The proposed method is based on determining the replacement cost of service of each client. Two major drivers of the replacement cost include vehicle usage as measured by ether distance or time.

The POCS mechanism introduced by Furuhata et al. [21] solves the online cost allocation problem associated with a demand-responsive transport (DRT) system, where new service requests are submitted sequentially over time, rather than all known at the same time. The POCS mechanism adapts proportional and marginal cost sharing methods into the online setting. In particular, customers that have consecutive request orders can choose to form coalitions. The shared costs among customers within the same coalition is proportional to their demand, while the sum of the shared costs of all customers in a coalition equals to the sum of their marginal costs. POCS is a flexible framework in the sense that no specific cost function is defined. All of the desired properties hold as long as the cost function of choice satisfies the following two properties: 1) total cost is non-decreasing over time (over order submissions); 2) total cost is independent of the submit order of customers who have already submitted their requests.

The POCS mechanism relaxes one of the constraints of static cost allocation problems, namely that the entire set of players must be known at once. However, one limitation still remains. POCS assumes that all customers submit their service requests before vehicle operations start, so that the routing schedule can be recalculated each time a new customer submits its request. In the dynamic vehicle routing environment we study, customer request submissions and vehicle operations take place simultaneously. The portion of the routing schedule that has been implemented cannot be reversed. Thus the vehicle routing problem and the cost allocation problem become highly interdependent. The assumption that the total cost is independent of the submit order of customers does not hold trivially. In this report, we extend beyond this limitation by developing novel ways to construct routing solutions, calculate total costs, and dynamically route the fleet.

## 3 A Motivating Example

One of the central themes of this report is the ability to reduce the costs that participants must pay by leveraging an *economy of scale*. Specifically, by planning a service vehicle's route in an efficient way, it is possible to provide service jointly to a large set of customers with a lower cost than providing service to those customers individually. The complicating factor, as we have already mentioned, is the uncertainty in the dynamic arrivals of future customers. We will address this this using computational simulations in subsequent sections, but for now, we give a simple "back of the envelope" style calculation to emphasize the key concepts that arise in this area. We describe a simple model using the *continuous approximation paradigm* [10], which allows one to quantify system performance metrics with minimal data and problem assumptions. These methods rely on concise summaries of information and simplified analytical forms, including closed form expressions. In routing problems, such methods have been introduced to approximate the length of a vehicle route using simple analytical forms based on the area of the service region and the spatial density of average demand realizations, and we shall do so presently.

Consider a set of $\mathcal{N}$ customers that are uniformly distributed in a geographic region with area $\mathcal{A}$, together with a vehicle located at a central depot. We seek the shortest vehicle route that originates at the vehicle, visits all of the $\mathcal{N}$ customers, and returns to the depot. A well-known result due to [12] establishes that the length $L$ of the tour is approximately

$$L \approx \kappa \sqrt{\mathcal{A}\mathcal{N}}$$

for large $\mathcal{N}$, where $\kappa$ is a constant that depends on the connectivity of the road network; for typical urban settings one typically sees $\kappa \approx 1.5$; see for example [36]. If we assume that the cost of providing service is equal to the length of the tour (disregarding linear proportionalities for the sake of notation), we see that the average cost per customer is equal to

$$\frac{L}{\mathcal{N}} \approx \kappa \sqrt{\mathcal{A}} \cdot \frac{1}{\sqrt{\mathcal{N}}} \tag{1}$$

which is decreasing and approaches zero as $\mathcal{N} \to \infty$. We see that, for example, if the number of customers doubles, then the average cost per customer reduces by a factor of $1/\sqrt{2} \approx 0.71$.

The above analysis does not take into account the vehicle capacities; suppose now that the vehicle can visit a limited amount of customers, say $C$, before it must return to the central depot.

When this is the case, [11] shows that the length of the optimal tour is approximately

$$L \approx 2r\frac{\mathcal{N}}{C} + \kappa\sqrt{\mathcal{A}\mathcal{N}}\,,$$

where $r$ represents the average distance between a customer and the central depot. We now see that the average cost per customer is

$$\frac{L}{\mathcal{N}} \approx \frac{2r}{C} + \kappa\sqrt{\mathcal{A}}\cdot\frac{1}{\sqrt{\mathcal{N}}} \tag{2}$$

which is decreasing and approaches $2r/C$ as $\mathcal{N} \to \infty$. We therefore see that vehicle capacities have a significant impact on the extent to which cost-sharing can provide a benefit when we compare the above result to the uncapacitated case.

## 3.1 The impact of dynamic customers

The key problem component that motivates this report is the issue of demand that arrives *dynamically* over a time horizon. This means that the vehicle's route may update over time, as will the costs per customer. In addition there is also the concern of *feasibility constraints*, since a customer may not be able to fit into the vehicle's current tour.

Our recent paper [7] presents a formula, similar to (1) and (2), for addressing the above issues. At any time $t$, let $C(t)$ denote the set of customers who have requested service by time $t$. Of course, by the preceding analysis, we see that the length of a tour that visits all of the customers will be approximately $\kappa\sqrt{\mathcal{A}|C(t)|}$. If this tour is infeasibly long, then the vehicle should instead seek the path that visits as many customers as possible without violating the length threshold. We show in [7] that if the vehicle seeks to visit as many customers as possible with a tour of length at most $\ell$, then the number of customers covered is approximately

$$\frac{\ell}{\kappa\sqrt{\mathcal{A}}}\cdot\sqrt{C(t)}$$

as $t \to \infty$. Moreover, the cost per customer is

$$\frac{\ell}{\kappa\sqrt{\mathcal{A}}}\cdot\frac{1}{\sqrt{C(t)}}\,,$$

which is again decreasing and approaches zero as $C(t) \to \infty$. Thus, we see that, while a vehicle capacity constraint (the parameter $C$ in (2)) imposes a positive lower bound on the cost per customer, a vehicle distance threshold does not, although this comes at a separate cost, namely that a set of customers is neglected.

# 4 The Online Cost Allocation Problem

The first step to tackle the cost allocation problem in a real-time cost sharing transportation system is to model the underlying dynamic vehicle routing problem. The second step in building a real-time cost sharing transportation system is to study how the total operating cost is allocated to each participant in the cooperation. To study a static cost allocation problem, one needs to define the set of players, the total cost function, and the calculation of the shared costs. In the online cost allocation problem we study, the key challenge lies in how to incorporate the time dimension into a cost-sharing mechanism. In particular, we need to specifically design how the set of players, the total cost function, and the calculation of shared costs evolve over time, as more problem information becomes available.

In this section, we first formulate the vehicle routing problem with dynamic customer requests and introduce the notation. We then state the online cost allocation problem in the DVRP and discuss a list of desirable properties for online cost-sharing mechanisms. We also use examples to illustrate how typical cost-sharing methods tend to fail in the online environment.

## 4.1 Dynamic Vehicle Routing Problem Definition

Suppose that the operation consists of routing a fleet of capacitated vehicles to collect shipments from a set of customers and transport them to a central depot. The length of the planning horizon is $T_{max}$ and can be discretized into time steps of unit length. There are $\mathcal{N}$ potential customers. Each customer has a fixed location, a known demand size, a known service time window and a service time of fixed length. The service time window specifies the earliest and latest times when service can be started at the corresponding customer and cannot be violated. Each customer requests service at most once during the planning horizon. The uncertainty lies in the fact that not all customers would request service. Some customers request service in advance (prior to the beginning of the planning horizon), and are called advance customers. The rest of the customers are called dynamic customers, who may or may not request service during the planning horizon. We assume that the probability a dynamic customer requests service can be estimated from historical information. The time when a dynamic customer requests service is called its request time. It is also the time when it becomes certain that the customer needs to be served. The objective is to minimize the total travel distance of all vehicles.

The following notations are used for model parameters and decision variables. Generally, $i$

and $j$ are used to index customers, $k$ to index vehicles/routes, and $t$ to index time.

| | |
|---|---|
| $\mathcal{N}$ | total number of customers |
| $\mathbb{AC}$ | set of advance customers |
| $\mathbb{DC}$ | set of dynamic customers |
| $d_i$ | demand of customer $i$ |
| $s_i$ | service time of customer $i$ |
| $e_i$ | the earliest time that service can begin at customer $i$ |
| $l_i$ | the latest time that service can begin at customer $i$ |
| $v_i$ | request deadline of customer $i$ |
| $u_i$ | actual request time of customer $i$ |
| $t_{i,j}$ | minimum travel time between location $i$ and $j$ |
| $\mathcal{K}$ | total number of vehicles |
| $\mathcal{C}$ | capacity of each vehicle |
| $r_{k,t}$ | partial route for vehicle $k$ at time $t$ |
| $n_{i,k,t}$ | the $i$-th customer scheduled on vehicle $k$ at time $t$ |
| $a_i$ | time of arrival at customer $i$ |
| $b_i$ | time of departure from customer $i$ |
| $n_{0,k,t}$ | the location from where vehicle $k$ would start its new route if diverted at time $t$ |
| $a_{0,k,t}$ | the time when vehicle $k$ would become available to start its new route if diverted at time $t$ |

It is assumed that all vehicles travel at unit speed. Thus, the travel time is equatable with travel distance between corresponding locations. It is also assumed that no preemption in vehicle routes is allowed, meaning that a vehicle cannot be diverted while en route to its current scheduled customer. The vehicle can only be diverted after it reaches and finishes service at its current customer. The request time $u_i$ of dynamic customer $i$ represents the time when it becomes certain that customer $i$ needs to be serviced. $u_i$ is modeled as a random variable taking values on the interval $[0, v_i]$. The request deadline $v_i$ denotes the latest time that the customer must make the decision on whether it needs to be serviced or not. Generally speaking it is reasonable to set $0 < v_i \leq e_i$. In addition, we assume that real-time two-way communication capability is established between the central decision making unit and each vehicle. At any point in time, the decision maker is aware of the complete fleet status including current locations, directions, and remaining capacities. This enables dynamic real-time routing of the vehicles.

There are two issues that are uncertain about dynamic customer requests. First, whether the customer requests service at all during the planning horizon. Second, when will the customer request service given that it will do so. From a historical perspective, the probability that a customer requests service on any day can be estimated by the proportion of days that the customer has requested service among all the days of operation. We use $q_i$ to denote this probability. For the second issue, a distribution on request time can be estimated by the actual request times of the customer on the days when it actually requested service. By definition, this distribution is conditional on the fact that the customer requests service. Let $f_i(t)$ be the conditional probability density function of request time $u_i$. Recall that $u_i$ is defined on $[0, e_i]$, thus we have $\int_0^{e_i} f_i(t)dt = 1, \forall i$. Given this setup, the probability that a dynamic customer $i$ requests service during the time interval $[t_1, t_2], 0 \leq t_1 \leq t_2 \leq T_{max}$ on any day can be calculated as

$$P\left(i \text{ requests during } [t_1, t_2]\right) = P\left(i \text{ requests}, \ i \text{ requests during } [t_1, t_2]\right) \tag{3}$$

$$= P\left(i \text{ requests during } [t_1, t_2] \,|\, i \text{ requests}\right) * P\left(i \text{ requests}\right) \tag{4}$$

$$= \int_{t_1}^{t_2} f_i(t)dt * q_i \tag{5}$$

In a dynamic vehicle routing context, problem information are revealed gradually over time. In other words, the full set of customers cannot be know until the end of the planning horizon. At any time $t$ in the planning horizon, only the set of advance customers and a subset of dynamic customers who have already requested service are known. A problem consisting of only partial information is called a partial vehicle routing problem $P_t$. The solution to a partial problem at time $t$ is called a partial solution $S_t$, which consists of a collection of partial routing schedules, $S_t = \{r_{k,t}\}$ where $k = 1, \ldots, \mathcal{K}$. Besides, the sequence of customers alone does not uniquely determine an operational schedule. We also need to specify the exact arrival and departure times at each location along the route. Let $a_i$ and $b_i$ denote the arrival and departure times at customer $i$ respectively. A partial routing schedule for vehicle $k$ specifies the sequence of customers scheduled for the vehicle, together with the arrival and departure times at each customer. $r_{k,t} = \{n_{1,k,t}, \ldots, n_{|r_{k,t}|,k,t}, n_{|r_{k,t}|+1,k,t}\}$ where $|r_{k,t}|$ denotes the total number of customers scheduled on route $k$ at time $t$. $n_{|r_{k,t}|+1,k,t} = 0, \forall k, t$ is a dummy place holder variable representing the constraint that all vehicles must return to the depot by the end of the planning horizon.

$n_{0,k,t}$ denotes the location from where vehicle $k$ would start its new route if it was diverted at time $t$. It can be loosely interpreted as the "available position" of vehicle $k$. At any time $t$, vehicle

$k$ must be in exactly one of the following two states. State I: serving or idling at some customer $i$. State II: en route to some customer $i$. In either case, if a new routing schedule were to be constructed at the moment and the vehicle is diverted, the new route must start at location $i$ (no preemption assumption). Hence in either case, we have $n_{0,k,t} = i$. In fact, during the implementation of vehicle routes, the $n_{0,k,t}$ variable should be updated once the vehicle starts to travel to its next customer based on the no preemption rule.

Similarly, $a_{0,k,t}$ denotes the time when vehicle $k$ would become available to start its new route if it were to be diverted at time $t$. It can be loosely interpreted as the "available time" of vehicle $k$. At any time $t$, if vehicle $k$ is currently servicing customer $i$, then $a_{0,k,t} = a_i + s_i$; if the vehicle is idle, then $a_{0,k,t} = t$; if the vehicle is traveling to service customer $i$, then $a_{0,k,t} = a_i + s_i$.

## 4.2 Cost Allocation Problem Statement

### 4.2.1 Problem Definition

In the dynamic vehicle routing problem we study, the uncertainty lies in the fact that part of the customers are confirmed at the beginning of the planning horizon while the rest are revealed over time when the vehicles are in operation. The entire set of customers who may potentially request service is known and fixed. Among these customers, some know for sure that they require service and have confirmed so at the beginning of the planning horizon, before the routing schedules are calculated. These customers are called advance customers. The rest of the customers are called dynamic customers, who may or may not request service during the planning horizon. The time when a dynamic customer becomes certain that it requires service is called its truthful request time. We assume that the service provider enforces a deadline for each customer to request service. The deadline for each customer may be different, and is always no later than the beginning of the service time window of the customer. The truthful request time of each customer is a random variable taking values on the interval ranging from the beginning of the planning horizon to its request deadline. We assume that the passenger cannot request service prior to its truthful request time, but may choose to delay its request in anticipation to take advantage of a possibly lower shared cost. In such cases, we distinguish its truthful request time, which is its earliest possible request time, from its actual, perhaps delayed, request time. Similarly, we assume that the actual request time of each dynamic customer cannot be later than its request deadline. That is to say, the request deadline of each dynamic customer aligns with the latest time when we could know with certainty whether the customer requires service or not. It is assumed that no two dynamic customers can have the same request time, either the truthful request time or actual request time. Once a dynamic customer requests service, it is called a realized dynamic customer.

The solution of a cost allocation problem usually comes in the form of a cost-sharing mechanism, which takes the set of customers as the input and generates the shared cost of each participant as the output. A cost-sharing mechanism should specify at least two cost functions: a total cost function that returns the total transportation cost of serving the set of customers, and a shared cost function that returns the shared cost of each individual participant. In the online cost allocation setting, however, the shared cost of each participant usually changes over time, possibly due to realization of new customers, cancellation of existing customers, and changes in network conditions that affects the total operating cost. An online cost-sharing mechanism should instead re-calculate the total operating cost and the shared cost of each customer whenever any of these changes happens. The mechanism should also record the sequence of shared costs over time for each customer.

When a dynamic customer requests service, the total cost of serving all customers may change, so does the shared cost of each existing customer. The dynamic customer should be immediately considered in the cost allocation problem and be offered a shared cost. The shared cost that a customer receives at the time of its request serves as its initial quote. Each customer may have a willingness-to-pay value that aligns with its valuation of the service received. The initial quote is the price that the customer would have to compare with its willingness-to-pay value to make the decision of whether to accept or decline the service.

How the total transportation cost should be calculated and shared among both advance and realized dynamic customers over time is a non-trivial problem for the following reasons: First, advance customers become known at the beginning of the planning horizon and should be offered their initial quotes at the same time, without knowledge on how many and which dynamic customers would request service. The way cost is shared among advance customers should obey standards typically required in static cost allocation problems, including fairness, budget balance, etc. As the planning horizon rolls out, the shared costs for advance customers together with the shared costs for realized dynamic customers should obey the properties required in the online setting. Second, customers should be given incentives to request service as early as possible to allow more time for calculating routing schedules. This suggests that an ideal mechanism should ensure that the best strategy for each individual customer to achieve the lowest possible shared cost is to request service at its truthful request time. For the same reason, a good mechanism should be able to demonstrate that it is more advantageous for each customer to make its service request known early as an advance customer than to request late as a dynamic customer. For example, consider the extreme case where all of the customers are advance customers and are known before the fleet starts its operation. Then the situation practically becomes a static vehicle routing and cost allocation problem. The routing schedule and shared costs can be solved without the complexity caused by uncertainties. Last but not least, the initial quote provided to each customer should serve as an upper bound on the final shared cost of the customer, which is the shared cost value for the customer at the end of the planning horizon.

### 4.2.2 Typical Cost-sharing Methods

In a static cost allocation problem, where the entire set of players is known and the total cost of serving each subset of players is well defined, the most intuitive and fair way to share the cost is proportional cost sharing [54, 52], where the total cost is distributed among all customers proportionally to their demand of the common resource. Now consider the online cost allocation problem we study, where dynamic customers request service sequentially and each customer may or may not become realized. In this online setting, the most intuitive way of sharing cost is incremental cost sharing [40], where the shared cost of each new player equals to the marginal cost generated from including the new player. Under incremental cost sharing, the shared cost of each customer will remain the same through the planning horizon, and thus the final shared cost always equals to the initial quote for each customer. Another strategy is to naively adapt proportional cost sharing to the online setting by re-calculating shared costs each time a dynamic customer requests service. That is to say, the shared cost of each customer may change each time an additional customer enters the system, and there is no guarantee that the shared cost for any customer will not increase over time. Lastly, we could incorporate customer request forecasting to generate anticipated customers and take them into consideration when calculating shared costs. We call this strategy proportional cost sharing with forecast.

We now use a simple example to illustrate how these typical cost-sharing methods behave in an online setting. Suppose there are a total of 2 customers in the system and their locations are shown in Figure 1. The direct distances between customer $A$ and the depot, between customer $B$ and the depot, and between customer $A$ and $B$ are 8, 6, and 4 respectively. We assume that the unit travel cost is generated per unit distance traveled. We also assume that the demand for the transportation resource of each customer can be represented by its direct distance from the depot. Let customer $A$ be an advance customer who has been confirmed. Customer $B$ represents a dynamic customer who has a high chance of requesting service but has yet to do so. There are two possible outcomes concerning the randomness in this system, depending on whether customer B requests service or not. In case 1 customer B ends up not requesting service while in case 2 it does. The final routing schedule for both cases are drawn in Figure 1. Under both cases, we calculate the initial quote at the time of service request and final shared cost for each customer using the three cost-sharing methods discussed above, namely incremental cost sharing, proportional cost sharing without forecasting, and proportional cost sharing with forecasting. We assume that budget balance is always satisfied in both cases under all three cost-sharing methods, meaning that the sum of the final shared cost(s) of all realized customer(s) equals to the total travel cost corresponding to the routing schedule represented in Figure 1.

The results are shown in Table 1. For example, under the proportional cost sharing without forecasting, the initial quote provided to customer A is 16, which equals to the total travel cost when A is the only customer in the system. Under case 2, when customer B requests service, the total travel cost becomes 18. The shared costs for customer A and B are $^{144}/_{14}$ and $^{108}/_{14}$ respectively, in proportion to their direct distance. These cost values represent the final shared costs since all of the randomness in the system has been realized. Under the proportional cost sharing with forecasting, customer B is considered in the cost allocation problem since it has a high probability of becoming realized. The initial quote for customer A is $^{144}/_{14}$, and is calculated based on the routing schedule that customer B realizes, where the total travel cost is 18. However, in the case that customer B ends up not requesting service, the final shared cost of customer A has to be increased to 16 in order to recover the total travel cost of serving only customer A.

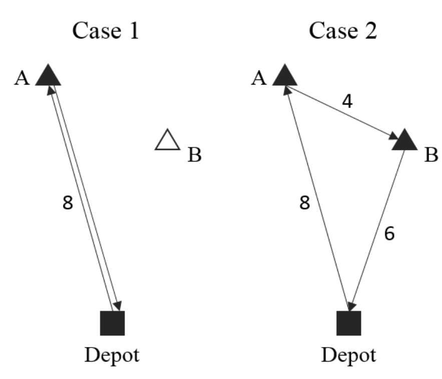


Figure 1: Example I: Typical cost-sharing methods

| | Case 1 | | Case 2 | | | |
|---|---|---|---|---|---|---|
| | Customer A | | Customer A | | Customer B | |
| | Initial | Final | Initial | Final | Initial | Final |
| Incremental Cost Sharing | 16 | 16 | 16 | 16 | 2 | 2 |
| Proportional Cost Sharing without Forecasting | 16 | 16 | 16 | $^{144}/_{14}$ | $^{108}/_{14}$ | $^{108}/_{14}$ |
| Proportional Cost Sharing with Forecasting | $^{144}/_{14}$ | 16 | $^{144}/_{14}$ | $^{144}/_{14}$ | $^{108}/_{14}$ | $^{108}/_{14}$ |

Table 1: Shared costs under typical cost-sharing methods

We now discuss whether each cost-sharing method shows desirable properties as discussed in the previous section. Under incremental cost sharing, customer B receives a much lower shared cost than customer A proportional wise based on their demand values, which is a clear violation of the fairness rule. In general, customers who request late tend to get lower shared costs since the marginal cost of service is almost always lower than the stand alone cost offered to the first customer. This unfairness may incentivize untruthful request times as customers wait to request service late in anticipation for possibly lower marginal costs. The other two proportional cost sharing-based methods do not suffer from the loss of fairness since the total cost is always shared proportionally among existing customers at any point in time. However, in the case that customer B does request service, the proportional cost sharing without forecasting method provides customer A with an initial quote of 16 that is much higher than its final shared cost of $\frac{144}{14}$. Recall that the initial quote is the value that the customer would have to use to make the decision on whether to accept the service or not. Even though the final shared costs is fair and low for customer A, the high quote may turn away the customer in the first place. On the contrary, if forecasting for customer B is used at the time of quoting customer A, the initial quote and final shared cost for either customer will be the same as in the case that customer B does become realized. On the other hand, if customer B does not request service, the final price charged for customer A would have to increase to 16 from its initial quote of $\frac{144}{14}$ in order to recover the total transportation cost and maintain the budget balance property. This may not be practical since customer A may not agree to pay for a higher price. There may be a loss in good faith if the customer accepts and pays for the higher price, or there may be wasted traveling if the customer chooses to drop service and the vehicle has already been en route. An alternative solution would be to keep customer A's initial quote as its final price and generate a budget deficit at the same time. Either situation is clearly undesirable for both the service provider and the customer.

### 4.2.3 Desirable Properties

It is evident from the example above that none of the cost-sharing methods discussed so far is well-suited for the dynamic vehicle routing problem. Before we develop a new mechanism, we first discuss a list of properties for an ideal online cost-sharing mechanism. Some of the properties correspond to their counterparts for static problems, such as fairness and budget balance. The rest are derived specifically for the online environment and are based on the shortcomings of typical cost-sharing methods discussed above.

**Online Fairness.** At any time during the planning horizon, the shared cost per demand value of any customer is never lower than those of customers who have requested service prior to the customer. The property has two implications. First, since requests by advance customers are all known at once, their request times are the same. There should not be any notion of early and late among advance customers. Thus, fairness for advance customers means that the shared cost per demand value of all advance customers should be the same. Second, since all advance customers request service before all realized dynamic customers, the shared cost per demand value of any advance customer should never be higher than that of any dynamic customer. It is important to point out the difference between shared costs and initial quotes. The shared cost of a customer usually changes over time as more customers enter the system. The initial quote, however, is the first shared cost valued provided to a customer at the time of its request. Thus the online fairness property does not require that the initial quote per demand value provided to any customer to be never higher than the one provided to a subsequent customer. In other words, it can happen that a customer who requests service late receives a lower initial quote per demand value than a prior customer. Nevertheless, in such a situation it is guaranteed that the current shared cost per demand value of the prior customer is never higher than the initial quote per demand value provided to a subsequent customer.

**Budget Balance.** At any time during the planning horizon, the sum of the shared costs of all customers equals to the total travel cost of the current routing schedule, including both traveled and untraveled portions of the schedule.

**Immediate Response.** Each customer should be provided with an upper bound on its final shared cost at the time of its service request. Since each customer has to make the decision of whether to accept or decline the service based on its willingness-to-pay level, this property guarantees that each customer only has to make that decision once at the time of its request, without having to worry about being charged against its will for a higher price than it previously agreed to.

**Individual Rationality.** At any time during the planning horizon, the shared cost of any customer who has accepted its initial quotes never exceed its willingness-to-pay level. Since a customer only remains in the cooperation as long as its shared cost does not exceed its willingness-to-pay level, individual rationality guarantees that no customer will drop out of the cooperation once it joins. This property also suggests that the initial quote serves as an upper bound on the final shared cost for each customer.

**Ex-Post Incentive Compatibility.** The best strategy of each customer is to request service truthfully at its earliest possible time, provided that all other customers do not change their request times and whether they accept or decline their initial quotes. This property has two implications. First, an advance customer cannot decrease its final shared cost by choosing to become a dynamic customer and not request service at the beginning of the planning horizon. Second, a dynamic customer cannot decrease its final shared cost by delaying its actual request time to be later than its truthful request time. For similar reasons as discussed under the online fairness property, this property is concerned with the final shared costs rather than initial price quotes. Thus it is possible for a customer, either an advance customer or a dynamic customer, to delay its actual request time and receive a lower initial quote than it would have received at its truthful request time. Even if it happens, the final shared cost of the same customer in the delayed request case is guaranteed to be no lower than in the truthful request case.

# 5 Hybrid Proportional Online Cost Sharing (HPOCS)

In this section, we formally define the Hybrid Proportional Online Cost Sharing (HPOCS) mechanism. We begin by defining relative terminologies. Then we explain how the shared costs are calculated and updated over time in the dynamic vehicle routing problem. We illustrate the mechanism using a simple DVRP example. We prove that HPOCS satisfies all of the desirable properties discussed in the previous section. In the end, we analyze experimental results of the baseline HPOCS mechanism.

## 5.1 Mechanism Design

We develop the HPOCS mechanism as an online cost-sharing mechanism that combines proportional cost sharing for solving static cost allocation problems and the Proportional Online Cost Sharing (POCS) mechanism [21] for handling sequential customer requests. In particular, proportional cost sharing is used to calculate the initial quotes for advance customers at the beginning of the planning horizon, while the POCS mechanism is used to handle dynamic customer requests. The idea behind POCS is that customers are partitioned into coalitions, where each coalition contains a sequence of customers who request service within given time intervals. At the time of its request, each customer first forms its own coalition. However, customers can choose to form coalitions with customers who request service directly after them to decrease their shared costs per alpha value. The formation of a coalition is determined by comparing the pooled marginal costs shared over subsets of customers each time a new customer enters the system. A set of specially designed total and marginal cost values for advance customers is used to initialize the POCS process for dynamic customers. This setup ensures that the coalition can be formed across both advance and dynamic customers. A routing technique together with the corresponding cost functions serves as the core of HPOCS.

In general, we use the same notations as in Section 4.1. Additional notations are introduced as necessary. Generally, $i$ and $j$ are used to index customers, $k$ to index vehicles/routes, $t$ to index time, and $n$ to index the request order.

Let $\mathbb{C}$ represent the grand set of potential customers, which is the union of the set of advance customers $\mathbb{AC}$ and the set of dynamic customers $\mathbb{DC}$ , $\mathbb{C} = \mathbb{AC} \cup \mathbb{DC}$, $|\mathbb{C}| = \mathcal{N}$. Let $C(t)$ represent the set of customers who have requested service by time $t$. By definition, $C(0) = \mathbb{AC}$ since none of the dynamic customer has requested service but all of the advance customers are already known at time $t = 0$. Let $c_{ij}$ represent the minimum travel cost between location $i$ and $j$ and it is assumed that the unit cost is the same as the unit distance traveled by any vehicle. Thus, the travel cost is equatable with travel distance between corresponding locations, $c_{ij} = t_{ij}$.

We now formally define terminologies related to the HPOCS mechanism.

**Definition 1.** The alpha value $\alpha_i$ of customer $i$ quantifies the demand of its service request. That is, how much of the transportation resource it requires. It can also be interpreted as the measure of inconvenience caused on accommodating the customer. The alpha value is assumed to be positive and independent of the request time of the passenger. Similarly, it is also independent of whether the customer is an advance customer or dynamic customer. We use

$$\alpha_i = c_{0,i} * d_i, \tag{6}$$

where $c_{0,i}$ represents the minimum travel cost between customer $i$ and the depot, and $d_i$ represents the demand of customer $i$.

**Definition 2.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, $\pi_t$ denotes a request order of the customers in $C(t)$. For $n \in [1, |C(t)|]$, $\pi_t(n)$ represents the $n^{th}$ customer to request service under request order $\pi_t$. For example, $\pi_t(n) = i$ means that customer $i$ is the $n^{th}$ customer to request service under request order $\pi_t$.

**Definition 3.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, $\bar{\pi}_t$ denotes the special request order based on the realization of the dynamic vehicle routing problem up to time $t$, where all realized dynamic customers are ordered after all advance customers. In particular, the first part of $\bar{\pi}_t$ consists of all of the advance customers. Since all of the advance customers become known at the same time, there should not be any notion of early and late among them. In fact, any ordering of advance customers can be used to build the first half of $\bar{\pi}_t$. The exact ordering does not affect the properties of HPOCS, which will be proved in later sections. The second part of $\bar{\pi}_t$ records the ordering of realized dynamic customers based on the ordering of their actual request times.

Suppose that $t_1$ and $t_2$ are two time points in the planning horizon with $0 \le t_1 \le t_2 \le T_{max}$, then the corresponding sets of customers $C(t_1)$ and $C(t_2)$ must satisfy $C(t_1) \subseteq C(t_2)$ and $|C(t_1)| \le |C(t_2)|$. In addition, let $n$ be any order index within the range $1 \le n \le |C(t_1)|$. Then we must have $\bar{\pi}_{t_1}(n) = \bar{\pi}_{t_2}(n)$. This is true because by Definition 3, the special request order is constructed by appending new customers who request service to the end of the current order. Thus the existing portion of the special request order is always preserved.

It is important to point out that $\pi_t$ is a general symbol used to represent any request order, while $\bar{\pi}_t$ is the request order uniquely defined by the realization of the DVRP. Nevertheless, given time $t \in [0, T_{max}]$, $\pi_t$ and $\bar{\pi}_t$ will always contain exactly the same set of customers, namely $C(t)$. Recall that $C(0) = \mathbb{AC}$, meaning that $\pi_0$ consists of all advance customers. The same is true for $\bar{\pi}_0$.

**Definition 4.** The grand schedule $\bar{S}$ is a complete routing solution to the static vehicle routing problem corresponding to the grand set of customers $\mathbb{C}$, which satisfies the following requirement. For any dynamic customer $i$, the time when the assigned vehicle is scheduled to leave from its predecessor location is no earlier than the request deadline of the dynamic customer, $v_i$. That is, all arrival and departure times are set based on the wait-first strategy. When a vehicle finishes service at its current customer and becomes idle, if the next customer on the schedule is a dynamic customer that has yet to request service, the vehicle should wait at its current location and only be allowed to travel either when the dynamic customer becomes realized or when its request deadline has been reached, whichever comes first. $\bar{S}$ takes the form of a set of vehicle routes each assigned to a single vehicle. $\bar{S} = \{r_k\}$ where $k = 1, \ldots, \mathcal{K}$. Each route $r_k$ specifies the sequence of customer visits as well as the exact arrival and departure times at each customer, which satisfies the corresponding time window constraints and the additional requirement discussed above.

**Definition 5.** Let $\bar{S}$ be a grand schedule corresponding to the set of customers $\mathbb{C}$, and let $C \subset \mathbb{C}$ be a subset of customers. $\bar{S}(C)$ is called the partial schedule induced by the grand schedule $\bar{S}$ and the set $C$, which is constructed by removing all of the customers not in $C$ from the grand solution $\bar{S}$. In particular, each customer that is not in $C$ is removed from the route, and its predecessor and successor scheduled on the same vehicle are connected with a direct link. For instance, $\bar{S}(\mathbb{AC})$ represents the partial schedule induced by the set of advance customers.

Given a feasible grand schedule $\bar{S}$ and any subset of customers $C \subset \mathbb{C}$, it can be easily shown that a feasible induced schedule $\bar{S}(C)$ is guaranteed to exist, based on the triangle inequality property of pairwise distances. It is also evident that such induced solutions are usually not unique. Besides, given the grand schedule $\bar{S}$, for any time $t \in [0, T_{max}]$, and any request order $\pi_t$, we use the notation $\bar{S}\left(\pi_t\left(n\right)\right)$ to represent the partial schedule induced by the set of first $n$ customers on the request order $\pi_t$. More specifically, $\bar{S}\left(\pi_t\left(n\right)\right)$ is an equivalent notation used to denote the same induced solution as $\bar{S}(C)$, where $C = \{\pi_t\left(1\right), \ldots, \pi_t\left(n\right)\}$.

**Lemma 6.** *For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers* *who have requested service $C(t)$, any integer $n \in [1, |C(t)|]$, and any two request orders $\pi_t$ and $\pi'_t$ satisfying that the sets of customers $\{\pi_t(1), \ldots, \pi_t(n)\} = \{\pi'_t(1), \ldots, \pi'_t(n)\}$, we have $\bar{S}(\pi_t(n)) = \bar{S}\left(\pi'_t(n)\right)$. Equivalently speaking, the induced partial schedule $\bar{S}(\pi_t(n))$ is independent of the request order among the customers it contains.*

*Proof.* The induced schedule $\bar{S}(\pi_t(n))$ is constructed by removing customers from the grand solution $\bar{S}$ rather than by inserting customers sequentially based on $\pi_t$. As a result, $\bar{S}(\pi_t(n))$ is only concerned with the set of customers that is removed (or otherwise remain), but not about the ordering of the customers specified by $\pi_t$. It then follows that $\bar{S}(\pi_t(n))$ and $\bar{S}\left(\pi'_t(n)\right)$ are exactly the same schedules. □

The following proposition states that given the set of customers who have requested service by time $t$, the induced partial schedule is independent from the request order among the customers within the set.

**Proposition 7.** *For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, and any two request orders $\pi_t$ and $\pi'_t$, we have*

$$\bar{S}(\pi_t) = \bar{S}\left(\pi'_t\right) = \bar{S}(C(t)) \tag{7}$$

*Proof.* By Lemma 6, we have $\bar{S}(\pi_t(n)) = \bar{S}\left(\pi'_t(n)\right)$ for any $n \in [1, |C(t)|]$. Setting $n = |C(t)|$, we have that

$$\bar{S}(\pi_t(n)) = \bar{S}(\pi_t(|C(t)|)) = \bar{S}\left(\pi'_t(|C(t)|)\right) = \bar{S}\left(\pi'_t(n)\right) \tag{8}$$

which proves the first equality. For the second equality, we note that by definition both schedules $\bar{S}\left(\pi'_t\right)$ and $\bar{S}(C(t))$ are induced by the same set of customers, namely those customers that have requested service by time $t$. In addition, both solutions are constructed in the same way by removing customers not in $C(t)$ from the grand schedule $\bar{S}$. The membership and ordering of each customer on each vehicle route is preserved. It follows that $\bar{S}\left(\pi'_t\right)$ and $\bar{S}(C(t))$ are exactly the same schedules. Thus we have completed the proof. □

We now define the cost functions used by HPOCS. Some cost functions are based on their counterparts in the POCS mechanism [21], such as coalition cost per alpha and shared cost. In the original POCS formulation, it is assumed that customers request service sequentially, and no two customers will request service at the same time. In the DVRP we study, all of the advance customers request service at the same time. Thus we extend the definitions in POCS to accommodate both advance and dynamic customers.

**Definition 8.** For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, and any request order $\pi_t$, the $totalcost\left(\bar{S}\left(C(t)\right)\right)$ is the total travel cost of the induced partial solution $\bar{S}\left(C(t)\right)$. Equivalently, $totalcost\left(\bar{S}\left(\pi_t\right)\right)$ can be used to represent the same total cost since the underlying partial schedules are practically the same, as stated by Proposition 7. We define $totalcost\left(\bar{S}\left(\emptyset\right)\right) := 0$.

**Definition 9.** The advance cost per alpha value *acpa* is the average cost per alpha value across all advance customers. It is calculated by dividing the total travel cost of the partial schedule induced by the set of advance customers by the sum of alpha values of all advance customers

$$acpa = \frac{totalcost\left(\bar{S}\left(\mathbb{AC}\right)\right)}{\sum_{i \in \mathbb{AC}} \alpha_i} = \frac{totalcost\left(\bar{S}\left(C(0)\right)\right)}{\sum_{i \in \mathbb{AC}} \alpha_i} \tag{9}$$

It is important to note that *acpa* is a constant value given the set of advance customers $\mathbb{AC}$.

**Definition 10.** For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any integer $n \in [1, |C(t)|]$, $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n\right)\right)\right)$ is the total operating cost required to serve the first $n$ customers on request order $\bar{\pi}_t$. Since $C(0) = \mathbb{AC}$ and $C(0) \subseteq C(t)$ for $t \geq 0$, we must have $|C(t)| \geq |\mathbb{AC}|$. The total cost function is defined differently for advance and dynamic customers. For the case of advance customers, let $1 \leq n^* \leq |\mathbb{AC}|$, so that $\bar{\pi}_t(n^*)$ represents an advance customer. We define

$$totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n^*\right)\right)\right) = acpa \sum_{n=1}^{n^*} \alpha_{\bar{\pi}_t(n)} \tag{10}$$

which states that the total cost of serving a group of advance customers is defined as the product of the advance cost per alpha value and the sum of the alpha values of all advance customers in the group. At $n^* = |\mathbb{AC}|$, $\bar{\pi}_t(n^*)$ represents the last advance customer on request order $\bar{\pi}_t$. We define

$$totalcost\left(\bar{S}\left(\bar{\pi}_t\left(|\mathbb{AC}|\right)\right)\right) = acpa \sum_{n=1}^{|\mathbb{AC}|} \alpha_{\bar{\pi}_t(n)} \tag{11}$$

$$= \frac{totalcost\left(\bar{S}\left(\mathbb{AC}\right)\right)}{\sum_{i \in \mathbb{AC}} \alpha_i} \sum_{n=1}^{|\mathbb{AC}|} \alpha_{\bar{\pi}_t(n)} \tag{12}$$

$$= totalcost\left(\bar{S}\left(\mathbb{AC}\right)\right) \tag{13}$$

The second equality follows from the definition of *acpa*, and the third equality follows from the fact that $\sum_{i\in\mathbb{AC}}\alpha_i = \sum_{n=1}^{|\mathbb{AC}|}\alpha_{\bar{\pi}_t(n)}$. Equation 13 states that the total cost of serving all advance customers as defined above equals to the total cost of the partial schedule induced by $\mathbb{AC}$. The boundary condition is satisfied and the above definition is consistent with Definition 8. For the case of dynamic customers, assume that $|\mathbb{AC}| < |C(t)|$. Let $|\mathbb{AC}| < n^* \leq |C(t)|$, so that $\bar{\pi}_t(n^*)$ represents a realized dynamic customer. Then $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n^*\right)\right)\right)$ is defined as the total travel cost of the induced partial solution corresponding to the first $n^*$ customers on schedule $\bar{\pi}_t$. Similarly as in Definition 8, we define $totalcost\left(\bar{S}\left(\pi_t\left(0\right)\right)\right) := 0$.

**Definition 11.** For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, any request order $\pi_t$, any customer $i \in C(t)$, let $n$ be the index order of the customer on request order $\pi_t$. Equivalently, $\pi_t(n) = i$ for some $n \in [1, |C(t)|]$. $mc\left(\pi_t(n)\right)$ denotes the marginal cost of serving customer $i$ under request order $\pi_t$ and is defined as the increase in total cost due to its request. That is

$$mc\left(\pi_t(n)\right) := totalcost\left(\bar{S}\left(\pi_t\left(n\right)\right)\right) - totalcost\left(\bar{S}\left(\pi_t\left(n-1\right)\right)\right) \tag{14}$$

Since the total cost function is defined differently for advance and dynamic customers, the marginal cost is also defined differently. We now define the marginal costs under the special request order $\bar{\pi}_t$. For the case of advance customers, let $1 \leq n^* \leq |\mathbb{AC}|$, so that $\bar{\pi}_t(n^*)$ represents an advance customer. Based on equations 10 and 14, we define

$$\begin{aligned} mc\left(\bar{\pi}_t\left(n^*\right)\right) &= totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n^*\right)\right)\right) - totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n^*-1\right)\right)\right) && (15)\\ &= acpa\sum_{n=1}^{n^*}\alpha_{\bar{\pi}_t(n)} - acpa\sum_{n=1}^{n^*-1}\alpha_{\bar{\pi}_t(n)} && (16)\\ &= acpa \times \alpha_{\bar{\pi}_t(n^*)} && (17)\end{aligned}$$

which states that the marginal cost of an advance customer equals to the product of the advance cost per alpha value and its alpha value. For the case of dynamic customers, assume that $|\mathbb{AC}| < |C(t)|$. Let $|\mathbb{AC}| < n^* \leq |C(t)|$, so that $\bar{\pi}_t(n^*)$ represents a realized dynamic customer. The marginal cost of the customer is defined as the increase in total travel cost of the partial solutions induced by the corresponding sets of customers. That is

$$mc\left(\bar{\pi}_t\left(n^*\right)\right) := totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n^*\right)\right)\right) - totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n^*-1\right)\right)\right) \tag{18}$$

We now define the coalition cost per alpha value, how HPOCS calculates the shared cost of each customer, and the concept of coalition.

**Definition 12.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any two integers $n_1, n_2 \in [1, |C(t)|]$ with $n_1 \leq n_2$, the coalition cost per alpha value of customers $\{\bar{\pi}_t(n_1), \ldots, \bar{\pi}_t(n_2)\}$ at time $t$ under submit order $\bar{\pi}_t$ is

$$ccpa_{\bar{\pi}_t(n_1,n_2)} := \frac{\sum_{n=n_1}^{n_2} mc\left(\bar{\pi}_t(n)\right)}{\sum_{n=n_1}^{n_2} \alpha_{\bar{\pi}_t(n)}} \tag{19}$$

**Definition 13.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any customer $i \in C(t)$, let $n$ be the index order of the customer on request order $\bar{\pi}_t$. Equivalently, $\bar{\pi}_t(n) = i$ for some $1 \leq n \leq |C(t)|$. Then the shared cost of customer $i$ at time $t$ under request order $\bar{\pi}_t$ is defined as

$$cost_t\left(\bar{\pi}_t(n)\right) := \alpha_{\bar{\pi}_t(n)} \min_{n \leq n' \leq |C(t)|} \max_{1 \leq n'' \leq n'} ccpa_{\bar{\pi}_t(n'',n')} \tag{20}$$

**Definition 14.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any two integers $n_1, n_2 \in [1, |C(t)|]$ with $n_1 \leq n_2$, a coalition $(n_1, n_2)$ at time $t$ is a group of customers $\{\bar{\pi}_t(n_1), \ldots, \bar{\pi}_t(n_2)\}$ with

$$\frac{cost_t\left(\bar{\pi}_t(n)\right)}{\alpha_{\bar{\pi}_t(n)}} = \frac{cost_t\left(\bar{\pi}_t(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} \tag{21}$$

for all order indices $n_1 \leq n \leq n_2$ and

$$\frac{cost_t\left(\bar{\pi}_t(n)\right)}{\alpha_{\bar{\pi}_t(n)}} \neq \frac{cost_t\left(\bar{\pi}_t(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} \tag{22}$$

for both order indices with $n = n_1 - 1$ and $n = n_2 + 1$ and $1 \leq n \leq |C(t)|$.

Definition 14 suggests that the membership of a coalition is determined solely by the shared cost per alpha value of each customer. A sequence of customers who request service consecutively in time and have the same shared cost per alpha value are said to be in the same coalition. In terms of coalition formation, it is irrelevant whether a customer is an advance customer or a dynamic customer; a single coalition can consist of both advance and dynamic customers. Nor is it relevant whether the group of customers are assigned on the same vehicle or not.

The following statements are concerned with the way coalitions form and evolve over time under the special request order $\bar{\pi}_t$.

**Proposition 15.** *At any time $t \in [0, T_{max}]$, under the special request order $\bar{\pi}_t$, the coalition cost per alpha value of any coalition consisting solely of advance customers is a constant value. The value is fixed given the set of advance customers $\mathbb{AC}$ and is independent from the actual subset of advance customers in the coalition.*

*Proof.* For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any two integers $n_1, n_2 \in [1, |C(t)|]$ with $n_1 \leq n_2$, suppose that both $n_1$ and $n_2$ represent advance customers. That is, $n_1, n_2 \in [1, |\mathbb{AC}|]$. Then the coalition cost per alpha value of customers $\{\bar{\pi}_t(n_1), \ldots, \bar{\pi}_t(n_2)\}$ at time $t$ under submit order $\bar{\pi}_t$ is

$$ccpa_{\bar{\pi}_t(n_1,n_2)} = \frac{\sum_{n=n_1}^{n_2} mc(\bar{\pi}_t(n))}{\sum_{n=n_1}^{n_2} \alpha_{\bar{\pi}_t(n)}} \tag{23}$$

$$= \frac{\sum_{n=n_1}^{n_2} acpa \times \alpha_{\bar{\pi}_t(n)}}{\sum_{n=n_1}^{n_2} \alpha_{\bar{\pi}_t(n)}} \tag{24}$$

$$= \frac{acpa \sum_{n=n_1}^{n_2} \alpha_{\bar{\pi}_t(n)}}{\sum_{n=n_1}^{n_2} \alpha_{\bar{\pi}_t(n)}} \tag{25}$$

$$= acpa \tag{26}$$

The second equality follows from equation 17. Note that the coalition cost per alpha value equals to the advance cost per alpha value, which only depends on the set of advance customers $\mathbb{AC}$ and is independent of $n_1$, $n_2$, and even the request order $\bar{\pi}_t$. Equivalently speaking, given the set of advance customers, the coalition cost per alpha value of any coalition formed solely by advance customers is the same. Thus we have completed the proof. □

**Proposition 16.** *At time $t = 0$, under the special request order $\bar{\pi}_0$, all advance customers form a single coalition.*

*Proof.* At time $t = 0$, for any customer $i \in \mathbb{AC}$, let $n$ be the index order of the customer on the special request order $\bar{\pi}_0$. Equivalently, $\bar{\pi}_0(n) = i$ for some $1 \leq n \leq |\mathbb{AC}|$. By Definition 13, the shared cost of customer $i$ at time $t = 0$ under request order $\bar{\pi}_0$ is

$$cost_0\left(\bar{\pi}_0(n)\right) = \alpha_{\bar{\pi}_t(n)} \min_{n \le n' \le |\mathbb{AC}|} \max_{1 \le n'' \le n'} ccpa_{\bar{\pi}_t(n",n')} \tag{27}$$

$$= \alpha_{\bar{\pi}_t(n)} \min_{n \le n' \le |\mathbb{AC}|} \max_{1 \le n'' \le n'} acpa \tag{28}$$

$$= \alpha_{\bar{\pi}_t(n)} \times acpa \tag{29}$$

The second equality follows from the fact that both $\bar{\pi}_0(n')$ and $\bar{\pi}_0(n'')$ represent advance customers and that the coalition cost per alpha value of any coalition consisting solely of advance customers is always equal to *acpa* (Proposition 15). The third equality follows since the term inside the minimization and maximization operator is a constant and independent from both operators. Equation 29 shows that the shared costs among advance customers at time $t = 0$ under the special request order $\bar{\pi}_0$ obey the proportional cost sharing rule. It then follows that the shared cost per alpha values of any two advance customers $\bar{\pi}_0(n_1)$ and $\bar{\pi}_0(n_2)$ with $n_1, n_2 \in [1, |\mathbb{AC}|]$ must be the same.

$$\frac{cost_0\left(\bar{\pi}_0(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} = \frac{cost_t\left(\bar{\pi}_t(n_2)\right)}{\alpha_{\bar{\pi}_t(n_2)}} = acpa \tag{30}$$

which in turn proves that all advance customers form a single coalition at time $t = 0$ under the special request order $\bar{\pi}_0$. □

**Corollary 17.** *For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any customer $i \in C(t)$, let $n$ be the index order of the customer on request order $\bar{\pi}_t$. Equivalently, $\bar{\pi}_t(n) = i$ for some $1 \le n \le |C(t)|$. Then*

$$\frac{cost_t\left(\bar{\pi}_t(n)\right)}{\alpha_{\bar{\pi}_t(n)}} = \min_{n \le n' \le |C(t)|} \frac{cost_{u_{\bar{\pi}_t(n')}}\left(\bar{\pi}_t(n')\right)}{\alpha_{\bar{\pi}_t(n')}} \tag{31}$$

*where $u_{\bar{\pi}_t(n')}$ is the request time of customer $\bar{\pi}_t(n')$ and $cost_{u_{\bar{\pi}_t(n')}}\left(\bar{\pi}_t(n')\right)$ represents the initial quote this customer receives at the time of its request.*

*Proof.* Consider any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any customer $i \in C(t)$, let $n$ be the index order of the customer on request order $\bar{\pi}_t$. Equivalently, $\bar{\pi}_t(n) = i$ for some $1 \leq n \leq |C(t)|$. Then we have

$$\frac{cost_t\left(\bar{\pi}_t(n)\right)}{\alpha_{\bar{\pi}_t(n)}} = \min_{n \leq n' \leq |C(t)|} \max_{1 \leq n'' \leq n'} ccpa_{\bar{\pi}_t(n", n')} \tag{32}$$

$$= \min_{n \leq n' \leq |C(t)|} \min_{n' \leq m \leq n'} \max_{1 \leq n'' \leq m} ccpa_{\bar{\pi}_t(n", m)} \tag{33}$$

$$= \min_{n \leq n' \leq |C(t)|} \frac{cost_{u_{\bar{\pi}_t(n')}}\left(\bar{\pi}_t(n')\right)}{\alpha_{\bar{\pi}_t(n')}} \tag{34}$$

where the first and third equalities both follow from Definition 13. □

**Lemma 18.** *Under the special request order $\bar{\pi}_t$, once a group of customers forms a coalition at time $t$, they will remain in the same coalition until the end of the planning horizon. More customers may join the same coalition over time, but the original group of customers will never depart the coalition.*

*Proof.* For any time $t_1 \in [0, T_{max})$ and the corresponding set of customers who have requested service $C(t_1)$, let $(n_1, n_2)$ be a coalition at time $t_1$ under the special request order $\bar{\pi}_{t_1}$, where $1 \leq n_1 \leq n_2 \leq |C(t_1)|$. Let $t_2 \in (t_1, T_{max}]$ be any later point of time in the planning horizon. Now consider any customer with the order index $n_1 \leq n \leq n_2$ under the special request order $\bar{\pi}_{t_1}$. Then

$$\min_{n \leq n' \leq |C(t_1)|} \frac{cost_{u_{\bar{\pi}_{t_1}(n')}}\left(\bar{\pi}_{t_1}(n')\right)}{\alpha_{\bar{\pi}_{t_1}(n')}} = \frac{cost_{t_1}\left(\bar{\pi}_{t_1}(n)\right)}{\alpha_{\bar{\pi}_{t_1}(n)}} \tag{35}$$

$$= \frac{cost_{t_1}\left(\bar{\pi}_{t_1}(n_1)\right)}{\alpha_{\bar{\pi}_{t_1}(n_1)}} \tag{36}$$

$$= \min_{n_1 \leq n' \leq |C(t_1)|} \frac{cost_{u_{\bar{\pi}_{t_1}(n')}}\left(\bar{\pi}_{t_1}(n')\right)}{\alpha_{\bar{\pi}_{t_1}(n')}} \tag{37}$$

where the first and third equalities both follow from Corollary 17 and the second equality follows from Definition 14. In addition, since $t_1 \leq t_2 \leq T_{max}$, request order $\bar{\pi}_{t_2}$ is an extension of the order $\bar{\pi}_{t_1}$. Thus $\bar{\pi}_{t_2}(m) = \bar{\pi}_{t_1}(m)$ for all $1 \leq m \leq |C(t_1)|$ by definition. Equation 37 can be rewritten as follows

$$\min_{n \leq n' \leq |C(t_1)|} \frac{cost_{u_{\bar{\pi}_{t_2}(n')}}\left(\bar{\pi}_{t_2}(n')\right)}{\alpha_{\bar{\pi}_{t_2}(n')}} = \min_{n_1 \leq n' \leq |C(t_1)|} \frac{cost_{u_{\bar{\pi}_{t_2}(n')}}\left(\bar{\pi}_{t_2}(n')\right)}{\alpha_{\bar{\pi}_{t_2}(n')}} \tag{38}$$

Now consider adding the following set of terms to the minimization operators on both sides of equation 37.

$$\left\{\frac{cost_{u_{\bar{\pi}_{t_2}(j)}}\left(\bar{\pi}_{t_2}(j)\right)}{\alpha_{\bar{\pi}_{t_2}(j)}}\right\}_{|C(t_1)| < j \leq |C(t_2)|} \tag{39}$$

Since the same set of terms are added to both minimization operators, the equality is preserved.

Equation 37 can be rewritten as follows

$$\min_{n \le n' \le |C(t_2)|} \frac{cost_{u_{\bar{\pi}_{t_2}(n')}}(\bar{\pi}_{t_2}(n'))}{\alpha_{\bar{\pi}_{t_2}(n')}} = \min_{n_1 \le n' \le |C(t_2)|} \frac{cost_{u_{\bar{\pi}_{t_2}(n')}}(\bar{\pi}_{t_2}(n'))}{\alpha_{\bar{\pi}_{t_2}(n')}} \tag{40}$$

which by Corollary 17 is equivalent to

$$\frac{cost_{t_2}(\bar{\pi}_{t_2}(n))}{\alpha_{\bar{\pi}_{t_2}(n)}} = \frac{cost_{t_2}(\bar{\pi}_{t_2}(n_1))}{\alpha_{\bar{\pi}_{t_2}(n_1)}} \tag{41}$$

We have established that all of the customers in the original coalition at time $t_1$ have the same shared cost per alpha value at any future time $t_2$. By the definition of coalition, all of these customers must be in the same coalition at time $t_2$. Thus we have completed the proof. □

As a corollary to Proposition 16, we prove that under the special request order $\bar{\pi}_t$, the set of advance customers will remain in the same coalition throughout the planning horizon.

**Corollary 19.** *At any time $t \in [0, T_{max}]$, under the special request order $\bar{\pi}_t$, all advance customers are in the same coalition.*

*Proof.* At time $t = 0$, the corollary holds trivially based on Proposition 16. At any time $t > 0$, given that all of the advance customers are in the same coalition, by Lemma 18, they will remain in the same coalition until the end of the planning horizon. Thus we have completed the proof. □

We are now well equipped to present the HPOCS mechanism. For a realization of the dynamic vehicle routing problem, the shared costs are calculated as follows.

**Initialization.** $t = 0$.

1. Formulate a static vehicle routing problem corresponding to the set of customers $\mathbb{C} = \mathbb{AC} \cup \mathbb{DC}$ and construct the grand solution $\bar{S}$.

2. Construct the special request order $\bar{\pi}_0$ consisting of all advance customers. Any ordering among advance customers can be used.

**Quoting advance customers.** All advance customers receive their initial quotes at time $t = 0$.

1. Calculate the advance cost per alpha value *acpa* based on Definition 9.

2. Calculate the total cost, marginal cost, coalition cost per alpha, and the shared cost of each advance customer under the special request order $\bar{\pi}_0$ by Definition 10, equation 17, Definition 12, and equation 29.

3. For each advance customer $i \in \mathbb{AC}$, suppose that $n$ is its order index on request order $\bar{\pi}_0$. Provide $cost_0\left(\bar{\pi}_0(n)\right)$ as the initial quote for customer $i$.

**Quoting dynamic customers.** A dynamic customer $i$ receives its initial quote when it requests service at time $t = u_i$.

1. Append customer $i$ to the end of the special request order $\bar{\pi}_{u_i - 1}$ to form the new special request order $\bar{\pi}_{u_i}$. Recall that $|C(t)|$ represents the total number of customers who have requested service. By definition, $\bar{\pi}_{u_i}\left(|C(u_i)|\right) = i$.

2. Construct the partial schedule induced by $C(u_i)$ and the grand schedule $\bar{S}$.

3. Calculate and update the total costs, marginal costs, coalition cost per alpha values, and the shared costs of all existing customers on request order $\bar{\pi}_{u_i}$ by Definition 10, equation 18, Definition 12, and Definition 13.

4. Provide $cost_{u_i}\left(\bar{\pi}_{u_i}(|C(u_i)|)\right)$ as the initial quote for customer $i$.

**Final shared costs.** $t = T_{max}$.

1. At time $t = T_{max}$, all of the randomness in the system has been realized. The special request order $\bar{\pi}_{T_{max}}$ consists of all advance and realized dynamic customers, namely the set $C(T_{max})$.

2. For $1 \leq n \leq |C(T_{max})|$, the shared cost of customer $\bar{\pi}_{T_{max}}(n)$ at time $T_{max}$ under the special request order $\bar{\pi}_{T_{max}}$ is $cost_{T_{max}}\left(\bar{\pi}_{T_{max}}(n)\right)$. This is also the final cost of service for customer $\bar{\pi}_{T_{max}}(n)$.

## 5.2 HPOCS Example

We now use an example to illustrate the HPOCS mechanism. Consider the dynamic vehicle routing problem based on the network shown in Figure 2 and the set of customers shown in Table 2. There are six customers and one depot located along a line segment. The distance between adjacent locations are as labeled. Customers $A$, $B$, and $C$ are advance customers, and the rest are dynamic customers. The demand of all customers are assumed to be 1, so that the alpha value of each customer equals to the distance from the depot to the customer. For convenience, we set the request deadline of each dynamic customer to be equal to the beginning of its service time window. The length of the service time of all customers are all equal 1 and $T_{max} = 50$. We consider the realization of the DVRP where customers $D$ and $F$ request service at time 9 and 20 respectively, and customer $E$ does not request service. It is assumed that all customers will accept any initial quote provided to them. Equivalently, the willingness-to-pay value of all customers are set to be equal to infinity. This assumption allows all of the customers that request service to stay in the system.

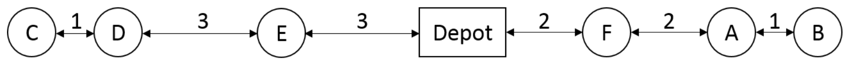


Figure 2: Example II: Network

| Customer | Demand | Alpha | Deadline | Time Window | Service Time | Request Time |
|---|---|---|---|---|---|---|
| A | 1 | 4 | – | $[0, 10]$ | 1 | 0 |
| B | 1 | 5 | – | $[5, 10]$ | 1 | 0 |
| C | 1 | 7 | – | $[25, 30]$ | 1 | 0 |
| D | 1 | 6 | 10 | $[10, 25]$ | 1 | 9 |
| E | 1 | 3 | 25 | $[25, 35]$ | 1 | – |
| F | 1 | 2 | 30 | $[30, 40]$ | 1 | 20 |

Table 2: Example V: Customer information

We solve the cost allocation problem associated with this realization of the DVRP in two cases. In the first case, we assume that only one vehicle is available, while in the second case, we assume that two vehicles are available.

**Single-vehicle case.**

Initialization.

All of the customers $A$ through $F$ are used to formulate a static VRP. Assuming that only one vehicle is available, the grand schedule $\bar{S}$ can take the form

$$r_1 = Depot[-, 0] - A[4, 5] - B[6, 10] - D[21, 24] - C[25, 26] - E[30, 31] - F[36, 37] - Depot[39, -] \quad (42)$$

where the two numbers in the brackets represent the arrival and departure times at each location. $C(0) = \mathbb{AC} = \{A, B, C\}$. Without loss of generality, we let the special request order $\bar{\pi}_0$ take the following form.

$$\bar{\pi}_0 = A - B - C \quad (43)$$

Quoting advance customers.

We first construct the partial schedule induced by the set $\mathbb{AC}$ and the grand schedule $\bar{S}$ as follows

$$\bar{S}(\mathbb{AC}) : r_1 = Depot[-,0] - A[4,5] - B[6,13] - C[25,26] - Depot[33,-] \quad (44)$$

Note that $\bar{S}(\mathbb{AC})$ is not unique and any feasible schedule can be used. Then we calculate the *acpa* value for advance customers.

$$acpa = \frac{totalcost\left(\bar{S}(\mathbb{AC})\right)}{\sum_{i \in \mathbb{AC}} \alpha_i} \quad (45)$$

$$= \frac{4+1+12+7}{4+5+7} \quad (46)$$

$$= 1.5 \quad (47)$$

The total cost, marginal cost, coalition cost per alpha, and the shared cost of each advance customer under the special request order $\bar{\pi}_0$ are calculated.

Table 3 shows the total and marginal cost values. For example, the total cost of serving customer $A$ is calculated as

$$totalcost\left(\bar{S}(\bar{\pi}_0(1))\right) = acpa \sum_{n=1}^{1} \alpha_{\bar{\pi}_t(n)} \quad (48)$$

$$= aspa \times \alpha_A \quad (49)$$

$$= 1.5 \times 4 \quad (50)$$

$$= 6 \quad (51)$$

and the corresponding marginal cost is calculated as

$$mc(\bar{\pi}_0(1)) = totalcost\left(\bar{S}(\bar{\pi}_0(1))\right) - totalcost\left(\bar{S}(\bar{\pi}_t(0))\right) \quad (52)$$

$$= totalcost\left(\bar{S}(\bar{\pi}_0(1))\right) - totalcost\left(\bar{S}(\emptyset)\right) \quad (53)$$

$$= 6 - 0 \quad (54)$$

$$= 6 \quad (55)$$

Since customer $A$ is an advance customer, the total cost after its request does not necessarily equal to the total travel cost of the partial solution induced by the set of customers that have requested service. Nevertheless, the total cost value defined on the entire set of advance customers $\mathbb{AC}$ is always equal to the total travel cost of the partial solution induced by the set $\mathbb{AC}$, as shown in

Definition 10. In this example, we have

$$totalcost\left(\bar{S}\left(\bar{\pi}_t\left(|\mathbb{AC}|\right)\right)\right) = acpa\sum_{n=1}^{|\mathbb{AC}|}\alpha_{\bar{\pi}_t(n)} = \frac{3}{2}\times(4+5+7) = 24 = totalcost\left(\bar{S}\left(\mathbb{AC}\right)\right) \qquad (56)$$

| Time | Request | Total Cost | Marginal Cost |
|---|---|---|---|
| 0 | A | 6.0 | 6.0 |
| 0 | B | 13.5 | 7.5 |
| 0 | C | 24.0 | 10.5 |
| 9 | D | 24.0 | 0.0 |
| 25 | F | 28.0 | 4.0 |

Table 3: Total and marginal costs with one vehicle

Table 4 summarizes the coalition cost per alpha values for each possible coalition formation. The row label represents the first customer in a coalition while the column label represents the last customer in the coalition. Based on the *ccpa* values, Table 5 shows the formation of coalitions among the customers. The shared costs per alpha *scpa* of each customer are calculated based on Definition 13 and customers who have the same *scpa* value are said to be in the same coalition. Last but not least, the HPOCS shared costs are calculated by multiplying the *scpa* value by the alpha value of each customer. The third row of Table 5 shows the initial quotes provided to advance customers at time $t = 0$.

| Time | Start of Coalition | End of Coalition | | | | |
|---|---|---|---|---|---|---|
| | | A | B | C | D | F |
| 0 | A | 1.5 | 1.5 | 1.5 | 1.1 | 1.2 |
| 0 | B | | 1.5 | 1.5 | 0.9 | 1.0 |
| 0 | C | | | 1.5 | 1.1 | 1.2 |
| 9 | D | | | | 0.0 | 0.5 |
| 25 | F | | | | | 2.0 |

Table 4: Coalition cost per alpha values with one vehicle

| Time | Coalitions | Shared Costs per Alpha | | | | | HPOCS Shared Costs | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A | B | C | D | F | A | B | C | D | F |
| 0 | (A) | 1.5 | | | | | 6.0 | | | | |
| 0 | (A, B) | 1.5 | 1.5 | | | | 6.0 | 7.5 | | | |
| 0 | (A, B, C) | 1.5 | 1.5 | 1.5 | | | 6.0 | 7.5 | 10.5 | | |
| 9 | (A, B, C, D) | 1.1 | 1.1 | 1.1 | 1.1 | | 4.4 | 5.5 | 7.6 | 6.6 | |
| 25 | (A, B, C, D) (F) | 1.1 | 1.1 | 1.1 | 1.1 | 2.0 | 4.4 | 5.5 | 7.6 | 6.6 | 4.0 |
| 50 | (A, B, C, D) (F) | 1.1 | 1.1 | 1.1 | 1.1 | 2.0 | 4.4 | 5.5 | 7.6 | 6.6 | 4.0 |

Table 5: Coalition formation, *scpa*, and HPOCS shared costs with one vehicle

Quoting dynamic customers.

A dynamic customer $i$ receives its initial quote at time $t = u_i$, when it requests service.

At time $t = 9$, customer D requests service and is appended to the end of the special request order. $\bar{\pi}_9 = A - B - C - D$. $C(9) = \{A, B, C, D\}$. The partial schedule induced by $C(9)$ and the grand schedule $\bar{S}$ is

$$\bar{S}\,(C(9)) : r_1 = Depot[-,0] - A[4,5] - B[6,10] - D[21,24] - C[25,26] - Depot[33,-] \tag{57}$$

The total cost, marginal cost, coalition cost per alpha values, and the shared costs of all customers at time $t = 9$ under request order $\bar{\pi}_9$ are calculated.

At time $t = 20$, customer F requests service and is appended to the end of the special request order. $\bar{\pi}_{20} = A - B - C - D - F$. $C(20) = \{A, B, C, D, F\}$. The partial schedule induced by $C(20)$ and the grand schedule $\bar{S}$ is

$$\bar{S}\,(C(20)) : r_1 = Depot[-,0]-A[4,5]-B[6,10]-D[21,24]-C[25,26]-F[35,36]-Depot[38,-] \tag{58}$$

The total cost, marginal cost, coalition cost per alpha values, and the shared costs of all customers at time $t = 20$ under request order $\bar{\pi}_{20}$ are calculated.

Final shared costs.

At time $t = T_{max} = 50$, all of the randomness in the problem has been realized. Since the shared costs only change when a dynamic customer requests service and time $t = 20$ was the last time a dynamic customer requested service, the final shared cost of each customer equals to its shared cost at time $t = 20$.

**Two-vehicle case.**

Initialization. All of the customers $A$ through $F$ are used to formulate a static VRP. Assuming that two vehicles are available, the grand schedule $\bar{S}^{'}$ can take the form

$$r_1^{'} = Depot[-,0] - A[4,5] - B[6,30] - F[33,34] - Depot[36,-] \tag{59}$$

$$r_2^{'} = Depot[-,10] - D[16,24] - C[25,26] - E[30,31] - Depot[34,-] \tag{60}$$

$C(0) = \mathbb{AC} = \{A, B, C\}$. Without loss of generality, we let the special request order $\bar{\pi}_0^{'}$ take the following form.

$$\bar{\pi}_0^{'} = C - B - A \tag{61}$$

Quoting advance customers.

We first construct the partial schedule induced by the set $\mathbb{AC}$ and the grand schedule $\bar{S}^{'}$. $\bar{S}^{'}(\mathbb{AC})$ can take the form

$$r_1^{'} = Depot[-,0] - A[4,5] - B[6,30] - Depot[35,-] \tag{62}$$

$$r_2^{'} = Depot[-,18] - C[25,26] - Depot[33,-] \tag{63}$$

Similarly as in the single vehicle case, $\bar{S}^{'}(\mathbb{AC})$ is not unique and any feasible schedule can be used. Now we calculate the advance cost per alpha value for advance customers.

$$acpa^{'} = \frac{totalcost\left(\bar{S}^{'}(\mathbb{AC})\right)}{\sum_{i \in \mathbb{AC}} \alpha_i} \tag{64}$$

$$= \frac{4+1+5+7+7}{4+5+7} \tag{65}$$

$$= 1.5 \tag{66}$$

The total cost, marginal cost, coalition cost per alpha, and the shared cost of each advance customer under the special request order $\bar{\pi}_0^{'}$ are calculated.

Table 6 shows the total and marginal costs. Similarly as in the single-vehicle case, the total cost value for a subset of advance customers are not necessarily equal to the total travel cost of the corresponding induced partial solution. Yet, the total cost value defined on the entire set of advance customers $\mathbb{AC}$ is always equal to the total travel cost of the partial solution induced by the set $\mathbb{AC}$.

| Time | Request | Total Cost | Marginal Cost |
|---|---|---|---|
| 0 | C | 10.5 | 10.5 |
| 0 | B | 18.0 | 7.5 |
| 0 | A | 24.0 | 6.0 |
| 9 | D | 24.0 | 0.0 |
| 25 | F | 24.0 | 0.0 |

Table 6: Total and marginal costs with two vehicles

Table 7 summarizes the coalition cost per alpha values for each possible coalition formation. The row label represents the first customer in a coalition while the column label represents the last customer in the coalition. Based on the coalition cost per alpha values, Table 8 shows the formation of coalitions among customers. The shared costs per alpha *scpa* of each customer are calculated based on Definition 13 and customers who have the same *scpa* value are said to be in the same coalition. Last but not least, the HPOCS shared costs are calculated by multiplying the *scpa* value by the alpha value of each customer. The third row of Table 8 shows the initial quotes provided to the advance customers at time $t = 0$.

| Time | Start of Coalition | End of Coalition | | | | |
|---|---|---|---|---|---|---|
| | | C | B | A | D | F |
| 0 | C | 1.5 | 1.5 | 1.5 | 1.1 | 1.0 |
| 0 | B | | 1.5 | 1.5 | 0.9 | 0.8 |
| 0 | A | | | 1.5 | 0.6 | 0.5 |
| 9 | D | | | | 0.0 | 0.0 |
| 25 | F | | | | | 0.0 |

Table 7: Coalition cost per alpha values with two vehicles

| Time | Coalitions | Shared Costs per Alpha | | | | | HPOCS Shared Costs | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | C | B | A | D | F | C | B | A | D | F |
| 0 | (C) | 1.5 | | | | | 10.5 | | | | |
| 0 | (C, B) | 1.5 | 1.5 | | | | 10.5 | 7.5 | | | |
| 0 | (C, B, A) | 1.5 | 1.5 | 1.5 | | | 10.5 | 7.5 | 6.0 | | |
| 9 | (C, B, A, D) | 1.1 | 1.1 | 1.1 | 1.1 | | 7.7 | 5.5 | 4.4 | 6.6 | |
| 25 | (C, B, A, D, F) | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 7.0 | 5.0 | 4.0 | 6.0 | 2.0 |
| 50 | (C, B, A, D, F) | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 7.0 | 5.0 | 4.0 | 6.0 | 2.0 |

Table 8: Coalition formation, *scpa*, and HPOCS shared costs with two vehicles

Quoting dynamic customers.

A dynamic customer $i$ receives its initial quote at time $t = u_i$, when it requests service.

At time $t = 9$, customer D requests service and is appended to the end of the special request order. $\bar{\pi}_9^{'} = C - B - A - D$. $C(9) = \{A, B, C, D\}$. The partial schedule induced by $C(9)$ and the grand schedule $\bar{S}^{'}$ is

$$r_1^{'} = Depot[-, 0] - A[4, 5] - B[6, 30] - Depot[35, -] \quad (67)$$

$$r_2^{'} = Depot[-, 10] - D[16, 24] - C[25, 26] - Depot[33, -] \quad (68)$$

The total cost, marginal cost, coalition cost per alpha values, and the shared costs of all customers at time $t = 9$ under request order $\bar{\pi}_9^{'}$ are calculated.

At time $t = 20$, customer F requests service and is appended to the end of the special request order. $\bar{\pi}_{20}^{'} = C - B - A - D - F$. $C(20) = \{A, B, C, D, F\}$. The partial schedule induced by $C(20)$

and the grand schedule $\bar{S}^{'}$ is

$$r_1^{'} = Depot[-,0] - A[4,5] - B[6,30] - F[33,34] - Depot[36,-] \quad (69)$$

$$r_2^{'} = Depot[-,10] - D[16,24] - C[25,26] - Depot[33,-] \quad (70)$$

The total cost, marginal cost, coalition cost per alpha values, and the shared costs of all customers at time $t = 20$ under request order $\bar{\pi}_{20}^{'}$ are calculated.

Final shared costs.

At time $t = T_{max} = 50$, all of the randomness in the problem has been realized. Similarly as in the single-vehicle case, since that shared costs only change when a dynamic customer requests service and time $t = 20$ was the last time a dynamic customer requested service, the final shared cost of each customer equals to its shared cost at time $t = 20$.

In this example, customers are assigned across two vehicles but all of the customers end up forming a single coalition. This observation confirms the claim that coalitions are formed based on the shared cost per alpha values and the proximity in request times of the customers, rather than on their vehicle assignments.

## 5.3 Analysis of Properties

The HPOCS mechanism defines a way to allocate the total travel cost to each customer in the dynamic vehicle routing problem. By definition, this mechanism follows the same framework as the POCS mechanism, with the exception that the total cost function is defined differently. Given that the original POCS mechanism satisfies all of the desirable properties discussed in Section 4.2.3, it follows that the HPOCS mechanism also possess these properties, if it can be shown that the new total cost function satisfies the same assumptions as made by the POCS framework.

The POCS framework makes two assumptions of the total cost function. First, the total cost is non-decreasing over time. Second, the total cost at any time is independent of the request order among the group of customers that have requested service. These assumptions are, for example, satisfied for the minimal operating cost, which is the cost function used in the original POCS paper. However, optimality is not required in order for all of the desirable properties to be satisfied, as long as the cost function follows the two assumptions.

**Proposition 20.** *For any grand schedule $\bar{S}$, any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any integer $n \in [1, |C(t)|]$,* *the HPOCS total cost function* $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n\right)\right)\right)$ *is nondecreasing in* $n$ *and is independent of the request order of customers* $\{\bar{\pi}_t\left(1\right), \ldots, \bar{\pi}_t\left(n\right)\}$*. That is, for any request order* $\pi_t$ *satisfying* $\{\bar{\pi}_t\left(1\right), \ldots, \bar{\pi}_t\left(n\right)\} = \{\pi_t\left(1\right), \ldots, \pi_t\left(n\right)\}$*,* $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n\right)\right)\right) = totalcost\left(\bar{S}\left(\pi_t\left(n\right)\right)\right)$*.*

*Proof.* We first prove that $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n\right)\right)\right)$ is nondecreasing in $n$. Without loss of generality, let $n_1$ be any order index satisfying $1 \leq n_1 < |C(t)|$ and let $n_2 = n_1 + 1$. By definition, the partial schedule $\bar{S}\left(\bar{\pi}_t\left(n_1\right)\right)$ is constructed by removing customer $\bar{\pi}_t\left(n_2\right)$ from the schedule $\bar{S}\left(\bar{\pi}_t\left(n_2\right)\right)$. Let $i^-$ and $i^+$ represent the predecessor and successor locations of customer $\bar{\pi}_t\left(n_2\right)$ in the schedule $\bar{S}\left(\bar{\pi}_t\left(n_2\right)\right)$. Then we have

$$totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n_1\right)\right)\right) = totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n_2\right)\right)\right) - c_{i^-\bar{\pi}_t(n_2)} - c_{\bar{\pi}_t(n_2)i^+} + c_{i^-i^+} \tag{71}$$

Based on the triangle inequality property of pairwise distances, we have

$$c_{i^-\bar{\pi}_t(n_2)} + c_{\bar{\pi}_t(n_2)i^+} - c_{i^-i^+} \geq 0 \tag{72}$$

Thus equation 71 implies that

$$totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n_1\right)\right)\right) \leq totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n_2\right)\right)\right) \tag{73}$$

We have proved that $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n\right)\right)\right)$ is nondecreasing in $n$. We now prove the total cost is independent of the request order of customers $\{\bar{\pi}_t\left(1\right), \ldots, \bar{\pi}_t\left(n\right)\}$. Let $\pi_t$ be any request order satisfying $\{\bar{\pi}_t\left(1\right), \ldots, \bar{\pi}_t\left(n\right)\} = \{\pi_t\left(1\right), \ldots, \pi_t\left(n\right)\}$. That is to say, the first $n$ positions of $\pi_t$ and of $\bar{\pi}_t$ consist of the same group of customers. By Lemma 6, $\bar{S}\left(\bar{\pi}_t\left(n\right)\right)$ and $\bar{S}\left(\pi_t\left(n\right)\right)$ represent the same induced partial schedule. It then follows that $totalcost\left(\bar{S}\left(\bar{\pi}_t\left(n\right)\right)\right) = totalcost\left(\bar{S}\left(\pi_t\left(n\right)\right)\right)$ □

We have now proved that the total cost function used in the HPOCS mechanism satisfies the two assumptions required by the POCS framework. When implementing the HPOCS mechanism to solve the cost allocation problem associated with a DVRP, one must specify the way vehicles are routed in real time. In order to make the HPOCS mechanism satisfy all of the desirable properties, we need to define a dynamic vehicle routing strategy that can guarantee that the actual total travel cost incurred by the vehicles equals to the total cost calculated by the HPOCS mechanism. The following dynamic routing strategy satisfies this requirement.

1. Vehicles are routed based on the grand schedule $\bar{S}$.

2. No re-optimization is done during the planning horizon.

3. At the time when a vehicle is scheduled to depart from its current location and travel to a dynamic customer, if the customer has yet to request service, it is skipped and the vehicle travels directly from the predecessor location to the successor location of the dynamic customer.

Recall that by Definition 4, the grand schedule $\bar{S}$ requires that the time when a vehicle starts to travel to a dynamic customer is no earlier than the request deadline of the customer. If the customer has yet to request service by this time, it is certain that the customer will not request service at all. Thus if the customer is removed from the current schedule, it will not request service at a later time. Equivalently speaking, insertion of new customers is never needed under this routing strategy. The only diversion of vehicles happen when an unrealized dynamic customer is skipped, and no traveling is wasted due to the absence of dynamic customers. As a result, the total travel cost incurred by the vehicles is always equal to the total cost of the induced partial solution as calculated in Definition 10.

Thus, we can conclude that under the dynamic routing strategy defined above, the HPOCS mechanism satisfies all of the desirable properties discussed in Section 4.2.3, namely the online fairness, budget balance, immediate response, individual rationality, and ex-post incentive compatibility properties. The proofs follow directly from the proofs presented in the original POCS paper [21].

## 5.4 Experimental Analysis

We now analyze simulation results to study the effectiveness of the mechanism in terms of providing desirable quotes to both the advance and dynamic customers.

Simulations are performed on a modified Solomon RC201 instance for the vehicle routing problem with time windows (VRPTW) [51]. The instance specifies all of the deterministic information on customer locations, demands, service time windows, and fleet capacity. There are 100 customers, $\mathcal{N} = 100$. The length of the planning horizon is 960 time steps, $T_{max} = 960$. A dynamic vehicle routing instance is constructed by specifying two parameters, namely the percentage of advance customers - $ACPercent$, and the probability that a dynamic customer requests service - $RequestProb$. These two parameters jointly determine the mixture between the number of advance customers and the expected number of realized dynamic customers in the problem. We assume that all dynamic customers have the same probability of requesting service. $q_i = RequestProb, \forall i$. We use a triangular distribution function to model $f_i(t)$, the conditional probability density function of request time $u_i$. In particular, the minimum value of the distribution is set to 0, and the maximum value of the distribution is set to be equal to the request deadline, $v_i$. The mode of the distribution is set to $\frac{3}{4}v_i$. Within this time frame, the dynamic customers are more likely to make the request close to the time they need service. A realization of the problem specifies the actual set of advance customers, a group of dynamic customers who are to make requests, and the precise request times of these customers. For each dynamic instance, we simulate 50 realizations and report the average results. The grand schedule of each realization is calculated based on the assumption that all customers (both advance and dynamic) are known at the beginning of the planning horizon and must be served. $\bar{S}$ is the output of this deterministic VRP solved by construction and local search heuristics in [13].

When solving the corresponding cost allocation problem, the HPOCS mechanism calculates the initial quote of a customer at the time it requests service. It is assumed that all customers will accept any initial quote provided to them. Equivalently, the willingness-to-pay value of all customers are set to be equal to infinity. This assumption allows all of the customers that request service to stay in the system. The shared cost gets updated each time when a new dynamic customer requests service because existing customers can choose to form a coalition with the new customer if it can lower their shared costs. It is worth exploring how the sequence of the shared costs changes over time and how the overall pattern may be different for different customers.

Figure 3 illustrates a graph of a series of HPOCS shared costs of selected customers in the demand scenario, where $ACPercent = 0.25$ and $RequestProb = 0.75$. This setup reflects an operating environment with a relatively high proportion of dynamic customers. The number of advance customers is $100 * 0.25 = 25$ and the expected number of realized dynamic customers is $100 * (1 - 0.25) * 0.75 \approx 57$. The horizontal axis represents the request order. In this scenario, the first 25 positions of the request order correspond to advance customers. The vertical axis represents the shared cost per alpha value. Each data point on the graph represents the shared cost per alpha value of a selected customer at the time when the dynamic customer whose order index corresponds to the horizontal axis value requests service. Each trajectory on the graph represents the series of shared cost per alpha values of a selected customer. The first data point on each trajectory shows the initial quote per alpha value of the customer.

For example, the first trajectory shows the series of shared cost per alpha values of the first advance customer on the special request order. Since all advance customers have the same shared cost per alpha value at any time throughout the planning horizon, it is sufficient to use the first advance customer to represent the entire set. The following four series correspond to four dynamic customers. "Dynamic 1" corresponds to the first dynamic customer to request service. "Dynamic 2" represents the dynamic customer whose request position falls around the first 3-quantiles of the total expected number of realized dynamic customers. In this case, since $57 \times \frac{1}{3} \approx 19$, it corresponds roughly to the $19^{th}$ dynamic customer to request service, and equivalently the $44^{th}$ customer to request service when counting advance customers. Similarly, "Dynamic 3" represents the dynamic customer whose request position falls around the second 3-quantiles of the total expected number of realized dynamic customers. The last series represents a dynamic customer positioned near the end of the request order.

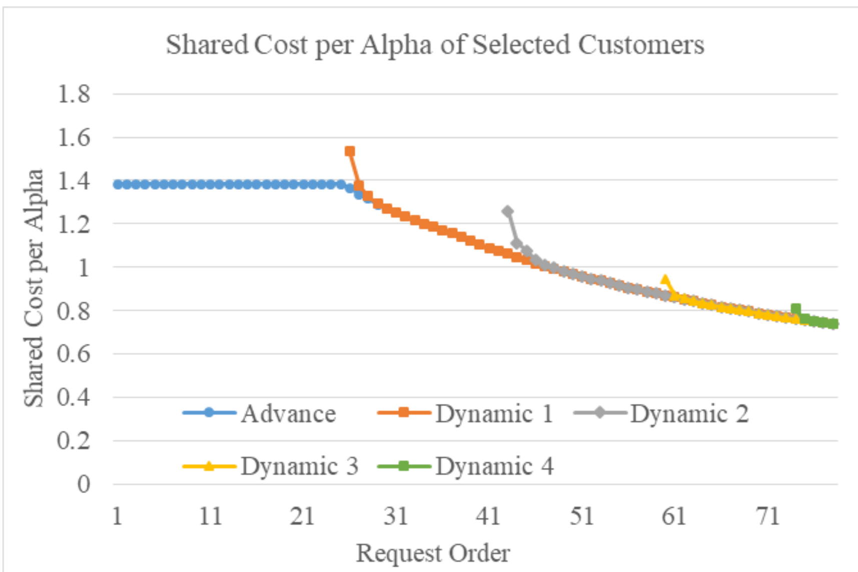


Figure 3: Trajectories of the HPOCS shared cost per alpha values in base case

It is worth pointing out that the request order shown by the horizontal axis is not equivalent to time. For instance, the first 25 units of the horizontal axis all correspond to time $t = 0$, because all of the advance customer requests are known at the same time. Each subsequent value on the horizontal axis corresponds to the actual request order.

It is evident from the graph that the shared cost of any customer is nondecreasing over the request order, which is a direct outcome of the way shared costs are calculated in the HPOCS mechanism. In particular, each time when a new customer requests service, existing customers will have the opportunity to form a coalition with the new customer. They will choose to form a new coalition if and only if their shared cost per alpha values can be lowered by doing so. Otherwise,

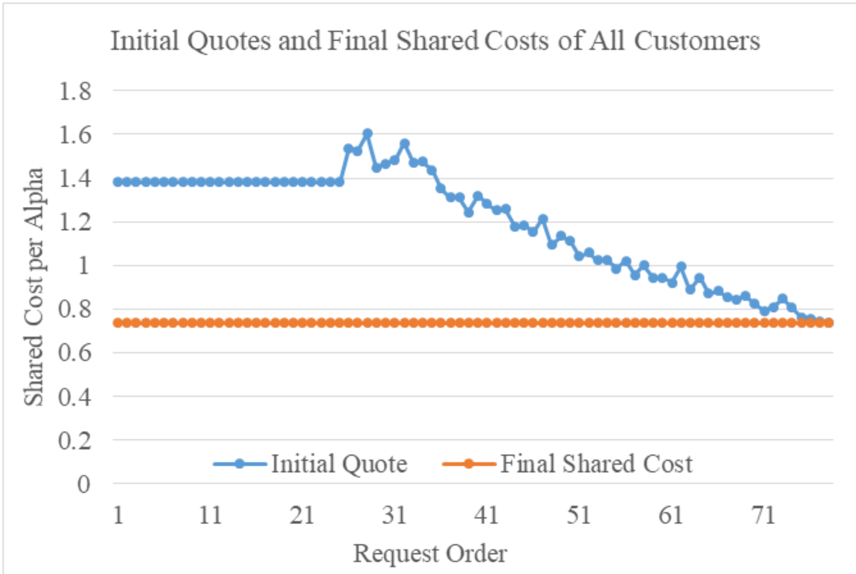


Figure 4: The HPOCS initial quotes and final shared cost values in base case

existing customers will choose to stay in their current coalitions.

Figure 4 illustrates a graph of the HPOCS initial quotes and the final shared costs of all customers in the base case demand scenario. Recall that the initial quote is the first shared cost value a customer receives and is the value that the customer has to use to make the decision of whether to accept the service or not. The final shared cost is the price that the customer actually pays for the service. These two values are the two most important shared cost values. All of the values shown on the graph are on the per-alpha basis. The horizontal axis represents the request order. In this scenario, the first 25 positions of the request order correspond to advance customers. The vertical axis represents the shared cost per alpha value. The upper series contains the initial quotes of all customers and the lower series contains the corresponding final shared costs. For each customer, its initial quote is always greater than or equal to its final shared cost, as guaranteed by the immediate response and individual rationality properties.

1. We first study the initial quotes provided to all customers. By Proposition 15, the initial quote per alpha value at time $t = 0$ of all advance customers are the same, and are equal to the advance cost per alpha *acpa* value. This is reflected by the level segment on the initial quote curve. For the realized dynamic customers, their initial quotes start higher than that of the advance customers, but drop very quickly as more dynamic customers become realized. Recall that the HPOCS mechanism calculates the total costs based on the total travel costs of induced partial solutions. All of these partial solutions are induced by a single grand solution that is

constructed at time $t = 0$ and is fixed throughout the planning horizon. As more customers request service, the grand schedule is gradually recovered and the synergy among the group of customers who have requested service increases. The marginal cost decreases, which makes it more attractive and likely for existing customers to form a new coalition with the customer who just requested. This in turn causes the initial quote offered to the dynamic customer that just became realized to decrease over time. This phenomenon can be undesirable since higher initial quotes offered to early request dynamic customers may turn them away if a finite willingness-to-pay threshold is implemented. If those early request dynamic customers decline service, the similar high level initial quotes will be offered to subsequent dynamic customers who request service, and the same problem remains. For the same reason, it is also undesirable that the initial quotes offer to many realized dynamic customers drop below the initial quote of advance customers.

2. We then study the final shared costs of all customers. It can be clearly seen from the graph that the final shared cost curve nearly represents a flat line. The final shared cost per alpha values across all advance and realized dynamic customers tend to be the same, which suggests that all of the customers tend to form a single coalition. The synergy among customers becomes so high that existing customers almost always can lower their shared costs by forming a new coalition with the dynamic customer that just became realized. This may be undesirable since customers that request early do not have any advantage over customers that request late. The lack of differentiation in the final shared costs fails to encourage customers to request service early. Given the discussion that the initial quotes offered to late request customers also tend to be lower than those offered to early request customers, customers may even be persuaded to request later.

A good mechanism should be able to demonstrate that it is more advantageous for each customer to make its service request known early as an advance customer than to request late as a dynamic customer. We seek to improve the performance of the HPOCS mechanism by providing an extra incentive for customers to declare their requests early. Thus a modification in HPOCS mechanism favoring early customers is introduced in Section 5. Next, we propose to incorporate a re-optimization method in generating total costs for partial schedules in purpose of boosting the performance of our HPOCS mechanism by reducing the overall shared cost, whether it is the initial quote or final price. The modified HPOCS with the re-optimization method is presented in Section 6.

# 6 Hybrid Proportional Online Cost Sharing with Discount (HPOCSD)

In this section, we introduce a modification of the HPOCS mechanism that aims to incentivize customers to request service early. Generally speaking, this can be achieved by offering discounts for advance customers to make it less costly to request early, and applying overcharge for dynamic customers to make it more costly to request late. The actual charge to each customer equals to its HPOCS shared cost times a cost modifier, which can either reflect a discount or an overcharge, depending on whether the customer is an advance customer or an dynamic customer. The same discount factor should be used for all advance customers in order to maintain the online fairness property. However, the overcharge factor can be different for different dynamic customers, and may be dependent on their actual request times. We design and study multiple heuristic methods for calculating the suitable overcharge factor for realized dynamic customers, based on their request orders and the discount factor for advance customers. In the following sections, we formally define the Hybrid Proportional Online Cost Sharing with Discount (HPOCSD) mechanism, study its properties, and analyze experimental results under various demand scenarios.

## 6.1 Mechanism Design

In the HPOCS mechanism, the shared costs are the actual price values provided to the customers. The idea behind HPOCSD is to use the modified charges to substitute for the HPOCS shared costs and offer the modified charges to the customers. All of the calculations of the total costs, marginal costs, coalition cost per alpha values, shared costs, and the definition of coalitions remain the same as defined by the HPOCS mechanism. Additional notations and definitions are as follows.

| | |
|---|---|
| $\delta$ | the discount factor |
| $\lambda_i$ | the cost modifier of customer $i$ |
| $g(n, \delta)$ | the overcharge function |

We require that $0 < \delta \leq 1$ and that $g(n, \delta) \geq 1, \forall n, \delta$.

**Definition 21.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any customer $i \in C(t)$, let $n$ be the index order of the customer on request order $\bar{\pi}_t$. Equivalently, $\bar{\pi}_t(n) = i$ for some $1 \leq n \leq |C(t)|$. Then the cost modifier of customer $i$ under request order $\bar{\pi}_t$ is defined as

$$\lambda_{\bar{\pi}_t(n)} = \begin{cases} (1-\delta) & \text{for } 1 \leq n \leq |\mathbb{AC}| \\ (1+g(n,\delta)) & \text{for } |\mathbb{AC}| < n \leq |C(t)| \end{cases} \tag{74}$$

The cost modifier for all advance customers is the same, and is equal to $1-\delta$. The cost modifier for a dynamic customer depends on the value of the function $g(n, \delta)$, which returns the overcharge factor based on the request index of the customer and the discount factor used for advance customers. Note that for each customer, its cost modifier is fixed and independent of time, meaning that the same factor will be applied to the sequence of shared costs of the customer.

**Definition 22.** For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any customer $i \in C(t)$, let $n$ be the index order of the customer on request order $\bar{\pi}_t$. Equivalently, $\bar{\pi}_t(n) = i$ for some $1 \leq n \leq |C(t)|$. Then the charge of customer $i$ at time $t$ under request order $\bar{\pi}_t$ is defined as

$$charge_t\left(\bar{\pi}_t(n)\right) = cost_t\left(\bar{\pi}_t(n)\right)\lambda_{\bar{\pi}_t(n)} \tag{75}$$

where $cost_t\left(\bar{\pi}_t(n)\right)$ denotes the HPOCS shared cost as defined in Definition 13. $charge_t\left(\bar{\pi}_t(n)\right)$ is the value that is provided to the customer.

We define the HPOCSD mechanism by using the same structure as the HPOCS mechanism presented in Section 5.1, except that all $cost_t\left(\bar{\pi}_t(n)\right)$ values are replaced with $charge_t\left(\bar{\pi}_t(n)\right)$ values. The same dynamic routing strategy presented in Section 5.3 is used for scheduling and routing vehicles.

## 6.2 Analysis of Properties

We now discuss the properties of the HPOCSD mechanism.

**Proposition 23.** *The HPOCSD mechanism satisfies the online fairness, immediate response, individual rationality, and ex-post incentive compatibility properties, provided that the* overcharge function $g(n, \delta)$ is nondecreasing in $n$.

*Proof.* We first prove the online fairness property. For any time $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any customer

$i \in C(t)$, let $n_1$ and $n_2$ be two indices representing advance customers, $1 \leq n_1 \leq n_2 \leq |\mathbb{AC}|$. Since the HPOCS mechanism satisfies the online fairness property, we have

$$\frac{cost_t\left(\bar{\pi}_t(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} = \frac{cost_t\left(\bar{\pi}_t(n_2)\right)}{\alpha_{\bar{\pi}_t(n_2)}} \tag{76}$$

Since both $n_1$ and $n_2$ are advance customers, their cost modifiers are the same and are equal to $\delta$. The equation above then implies that

$$\frac{charge_t\left(\bar{\pi}_t(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} = \frac{cost_t\left(\bar{\pi}_t(n_1)\right)(1-\delta)}{\alpha_{\bar{\pi}_t(n_1)}} = \frac{cost_t\left(\bar{\pi}_t(n_2)\right)(1-\delta)}{\alpha_{\bar{\pi}_t(n_2)}} = \frac{charge_t\left(\bar{\pi}_t(n_2)\right)}{\alpha_{\bar{\pi}_t(n_2)}} \tag{77}$$

which proves the online fairness property for advance customers. Now suppose $n_1$ and $n_2$ be two indices representing dynamic customers, $|\mathbb{AC}| < n_1 \leq n_2 \leq |C(t)|$. Since the HPOCS mechanism satisfies the online fairness property, we have

$$\frac{cost_t\left(\bar{\pi}_t(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} \leq \frac{cost_t\left(\bar{\pi}_t(n_2)\right)}{\alpha_{\bar{\pi}_t(n_2)}} \tag{78}$$

Given that both $n_1$ and $n_2$ are dynamic customers and that function $g(n, \delta)$ is nondecreasing in $n$, we have $1 \leq g(n_1, \delta) \leq g(n_2, \delta)$. It then follows that

$$\frac{charge_t\left(\bar{\pi}_t(n_1)\right)}{\alpha_{\bar{\pi}_t(n_1)}} = \frac{cost_t\left(\bar{\pi}_t(n_1)\right)(1+g(n_1,\delta))}{\alpha_{\bar{\pi}_t(n_1)}} \leq \frac{cost_t\left(\bar{\pi}_t(n)\right)1+g(n_2,\delta))}{\alpha_{\bar{\pi}_t(n_2)}} = \frac{charge_t\left(\bar{\pi}_t(n_2)\right)}{\alpha_{\bar{\pi}_t(n_2)}} \tag{79}$$

We have not proved that the online fairness property is satisfied for both advance and realized dynamic customers.

Similarly, given that for each customer $i$, the cost modifier $\lambda_i$ is fixed and independent of time, and that the overcharge function $g(n, \delta)$ is nondecreasing in $n$, it can be proved that the HPOCSD mechanism inherits the immediate response, individual rationality, and ex-post incentive compatibility properties from the HPOCS mechanism. □

**Proposition 24.** *The HPOCSD mechanism is $\delta-$budget balanced. That is to say, at any time during the planning horizon, the sum of the charges for all customers that have become realized recovers at least $100 \times (1-\delta)$ percent of the total travel cost of the corresponding induced partial schedule.*

*Proof.* For any grand schedule $\bar{S}$, at time $t = 0$, $C(0) = \mathbb{AC}$. We have

$$\sum_{n=1}^{|\mathbb{AC}|} charge_0\left(\bar{\pi}_0(n)\right) = \sum_{n=1}^{|\mathbb{AC}|} cost_0\left(\bar{\pi}_0(n)\right)(1-\delta) \tag{80}$$

$$= (1-\delta) \times totalcost\left(\bar{S}\left(\bar{\pi}_0\left(|\mathbb{AC}|\right)\right)\right) \tag{81}$$

$$= (1-\delta) \times totalcost\left(\bar{S}\left(C(0)\right)\right) \tag{82}$$

Which means that at time $t = 0$, the sum of the charges for advance customers using the HPOCSD mechanism recovers exactly $100 \times (1-\delta)$ percent of the total travel cost of the partial solution induced by $\bar{S}$ and the set $\mathbb{AC}$. Now consider any time during the planning horizon, $1 < t \leq T_{max}$. We have

$$\sum_{n=1}^{|C(t)|} charge_t\left(\bar{\pi}_t(n)\right) = \sum_{n=1}^{|\mathbb{AC}|} cost_t\left(\bar{\pi}_t(n)\right)(1-\delta) + \sum_{n=|\mathbb{AC}|+1}^{|C(t)|} cost_t\left(\bar{\pi}_t(n)\right)(1+g(n,\delta)) \tag{83}$$

$$\geq \sum_{n=1}^{|\mathbb{AC}|} cost_t\left(\bar{\pi}_t(n)\right)(1-\delta) + \sum_{n=|\mathbb{AC}|+1}^{|C(t)|} cost_t\left(\bar{\pi}_t(n)\right)(1-\delta) \tag{84}$$

$$= (1-\delta) \times \sum_{n=1}^{|C(t)|} cost_t\left(\bar{\pi}_t(n)\right) \tag{85}$$

$$= (1-\delta) \times totalcost\left(\bar{S}\left(\bar{\pi}_t\left(|C(t)|\right)\right)\right) \tag{86}$$

$$= (1-\delta) \times totalcost\left(\bar{S}\left(C(t)\right)\right) \tag{87}$$

where the inequality follows from the fact that the cost modifier of any dynamic customer is always greater than or equal to the cost modifier of any advance customer, $g(n,\delta) \geq (1-\delta), \forall n, \delta$. Equation 87 implies that the sum of the charges for all customers who have requested service recovers at least $100 \times (1-\delta)$ percent of the total travel cost of the corresponding induced partial solution. Thus we can conclude that the HPOCSD mechanism is $\delta-$budget balanced. □

In addition, we note that the equality in equation 84 is achieved if and only if $1 + g(n,\delta) = 1-\delta, \forall n, \delta$. This can only be true if $g(n,\delta) = \delta = 0$, which implies that practically no discount or overcharge is applied at all. Without the discounts and overcharges, the HPOCSD mechanism reduces to the HPOCS mechanism. In the HPOCSD setup with strictly positive discounts and overcharges, equation 84 will always imply an inequality relationship. It then follows that

$$\sum_{n=1}^{|C(t)|} charge_t\left(\bar{\pi}_t(n)\right) > (1-\delta) \times totalcost\left(\bar{S}\left(C(t)\right)\right) \tag{88}$$

at any time $1 < t \leq T_{max}$. This means that the worst-case budget deficit scenario always happens at time $t = 0$, when there is no realized dynamic customer and the sum of the HPOCSD charges recover exactly $100 \times (1 - \delta)$ percent of the total travel cost.

We have shown that the HPOCSD mechanism is approximately budget balanced. The loss of the budget balance property is the sacrifice that has to be made in order to encourage customers to request early. Proposition 24 provides an upper bound on the worst-case budget deficit, which is dependent on the discount factor provided to the advance customers. Intuitively speaking, the larger the discount, the more incentivize it provides to encourage customers to request early, and the bigger the risk of not being able to recover the total operating cost. On the other hand, Proposition 24 does not state that the HPOCSD mechanism will always incur a budget deficit. It could happen that the overcharge on dynamic customers recovers fully the discounts provided to advance customers and a budget balance is achieved. It could also happen that the overcharge over compensates for the discounts, such that a budget surplus is generated.

## 6.3 Experimental Analysis

We use the same experimental setup as introduced in Section 5.4. In our experiments, we use the parameters $ACPercent$ and $RequestProb$ to adjust the mixture between advance and realized dynamic customers. Recall that $ACPercent$ is the percentage of advance customers, and that $RequestProb$ is the probability that a dynamic customer requests service. Given the total number of potential customers $\mathcal{N}$, the number of advance customers is $\mathcal{N}_{AC} = \mathcal{N} \times ACPercent$, and the expected number of realized dynamic customers is $\mathcal{N}_{ERDC} = \mathcal{N} \times (1 - ACPercent) \times RequestProb$. For each realization of the dynamic vehicle routing problem, we solve the corresponding cost allocation problem using the HPOCSD mechanism paired with one of the following three heuristic methods for calculating the overcharge factors.

**Level overcharge.** The same overcharge factor is applied to all dynamic customers, no matter they request early or late.

$$g_{level}(n, \delta) = \frac{\delta \mathcal{N}_{AC}}{\mathcal{N}_{ERDC}} \times \gamma_{level} \tag{89}$$

where $\delta$ is the discount factor used for advance customers and $\gamma_{level}$ is a model parameter than can by tuned via experiments. Equation 89 states that the overcharge factor for dynamic customers is calculated based on and in proportion to the discount factor for advance customers. Since $\mathcal{N}_{AC}$

and $\mathcal{N}_{ERDC}$ represent the (expected) number of customers and do not take into account the alpha values, the parameter $\gamma_{level}$ is needed to adjust the actual overcharge level in order to avoid any systematic bias towards budget deficit or budget surplus under different demand scenarios.

**Linear overcharge.** The overcharge factor is designed to be linearly increasing over the request order. Dynamic customers that request late will be assigned a higher overcharge factor than those who request early.

$$g_{linear}(n, \delta) = \frac{\delta \mathcal{N}_{AC}}{\mathcal{N}_{ERDC}} \times \frac{(n + 1 - \mathcal{N}_{AC})}{\mathcal{N}_{ERDC}} \times \gamma_{linear} \tag{90}$$

Similarly as in the level overcharge heuristic, the above definition states that the linear overcharge factor is calculated based on and in proportion to the discount factor, and is linearly increasing over the request index $n$. Again, the parameter $\gamma_{linear}$ is needed to adjust the actual overcharge level to avoid bias.

**Exponential overcharge.** The overcharge factor is designed to be exponentially increasing over the request order, which provides smaller penalties for early request dynamic customers and larger penalties for late request dynamic customers as compared to the linear overcharge heuristic.

$$g_{exp}(n, \delta) = \frac{\delta \mathcal{N}_{AC}}{\mathcal{N}_{ERDC}} \times \frac{(\exp(\gamma_{exp}(n + 1 - \mathcal{N}_{AC})) - 1)}{(\exp(\gamma_{exp} \mathcal{N}_{ERDC}) - 1)} \times \gamma_{exp}^{'} \tag{91}$$

Similarly as in the level overcharge heuristic, the above definition states that the exponential overcharge factor is calculated based on and in proportion to the discount factor, and is exponentially increasing over the request index $n$. Two parameters $\gamma_{exp}$ and $\gamma_{exp}^{'}$ are needed to adjust the actual overcharge level to avoid bias.

Intuitively speaking, the larger the discount, the more significant the effect of incentivizing customers to request early. At the same time, the mechanism may be subject to bigger risks of not being able to recover the total operating cost. Thus, it is worth examining the performance of different overcharge heuristics using different discount factor levels. For each one of the overcharge heuristics presented above, we perform simulations using four discount factors, namely $\delta = 0.1, 0.2, 0.3$ and 0.4. We use the same base case demand scenario as used in Section 5.4, where $ACPercent = 0.25$ and $RequestProb = 0.75$.

Figures 5, 6, and 7 illustrate trajectories of the charge per alpha values of selected customers when using the HPOCSD mechanism, when paired with the level, linear, and exponential overcharge heuristics respectively. Each figure contains four panels, and each panel contains the graph of the

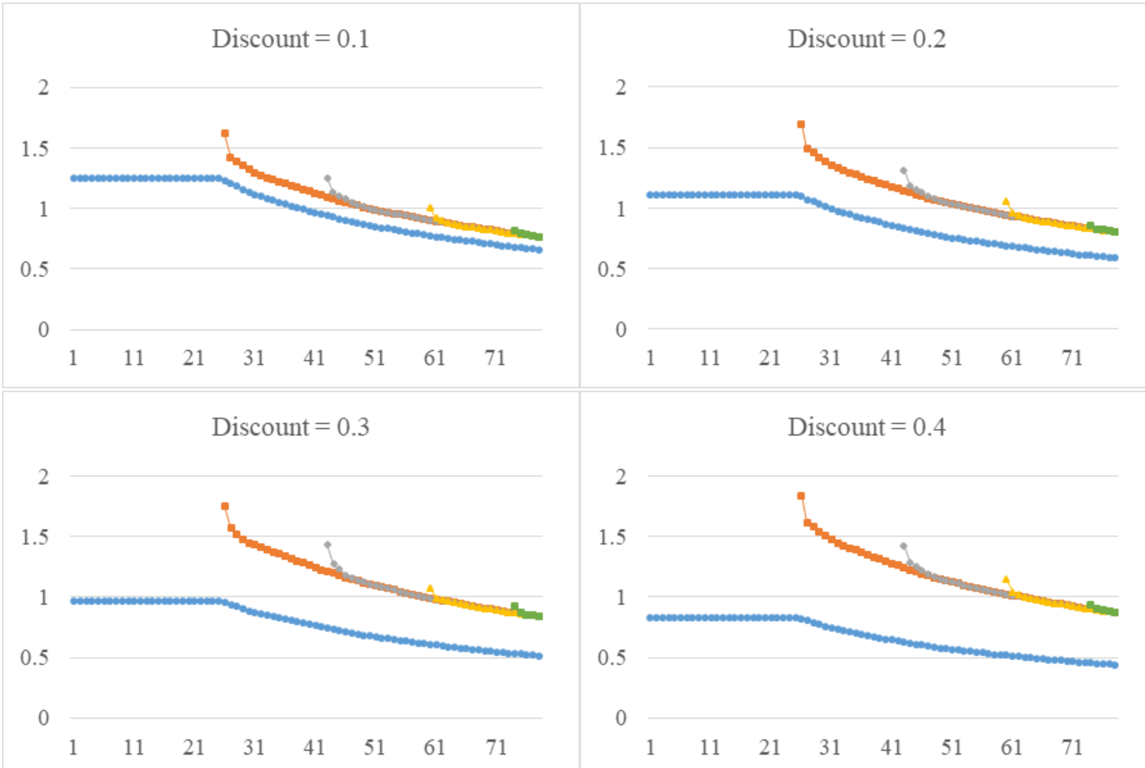


Figure 5: Trajectories of the charge per alpha values under HPOCSD with level overcharge

trajectories corresponding to one of the four discount factors that we have tested. On each graph, the horizontal axis represents the request order. In this scenario, the first 25 positions of the request order correspond to advance customers. The vertical axis represents the charge per alpha value. Each trajectory on the graph represents a series of charge per alpha values of a selected customer. The first data point on each trajectory shows the initial quote per alpha value of the customer.

There are five trajectories on each graph, and they correspond to the same customers across different graphs in different figures. For example, the first trajectory on any graph in any figure always correspond to the first advance customer on the special request order. The following four series correspond to four dynamic customers. The particular customer they each represent follows the same matching as shown in Figure 3.

1. It can be clearly observed from the graphs that for any overcharge heuristic using any of the discount factors we tested, the trajectories corresponding to realized dynamic customers always lie above the trajectory corresponding to the advance customer. It suggests that the HPOCSD mechanism is effective in terms of separating advance and realized dynamic customers into different coalitions. The gap between the two groups of trajectories increases

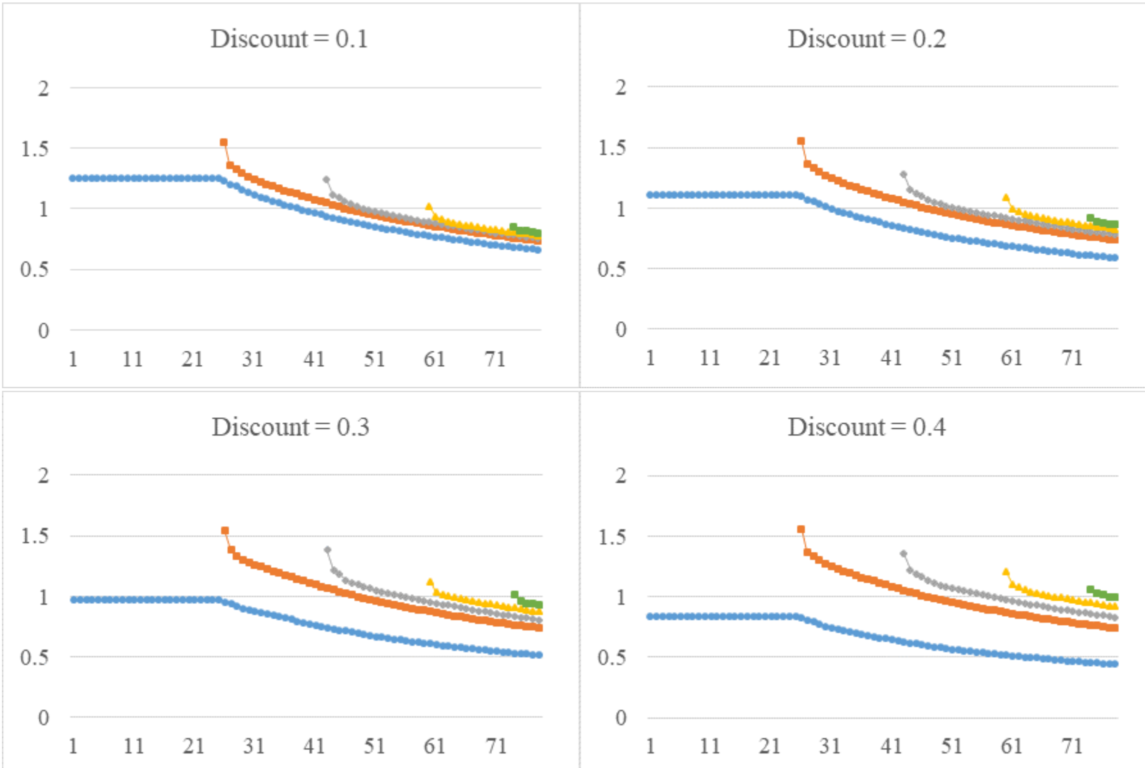


Figure 6: Trajectories of the charge per alpha values under HPOCSD with linear overcharge

as the discount factor increases, which confirms the intuition that a larger discount factor leads to a more significant effect.

2. When using the level overcharge heuristic, as shown in Figure 5, the trajectories corresponding to realized dynamic customers tend to overlap each other, following the same pattern as observed for the baseline HPOCS mechanism. This is because the level overcharge heuristic assigns the same overcharge factor to all realized dynamic customers, such that the charge per alpha values among all dynamic customers tend to remain the same as for the HPOCS mechanism. On the contrary, as shown in Figures 6 and 7, the linear and exponential heuristics can differentiate realized dynamic customers among themselves, since both heuristics penalize customers based on their request orders. Late request dynamic customers are penalized more than early request dynamic customers. The phenomenon is more significant when a larger discount factor is used.

Figures 8, 9, and 10 show graphs of the initial quote per alpha and the final charge per alpha values of all customers under the HPOCSD mechanism, when paired with the level, linear, and exponential overcharge heuristics respectively. Each figure contains four panels, and each

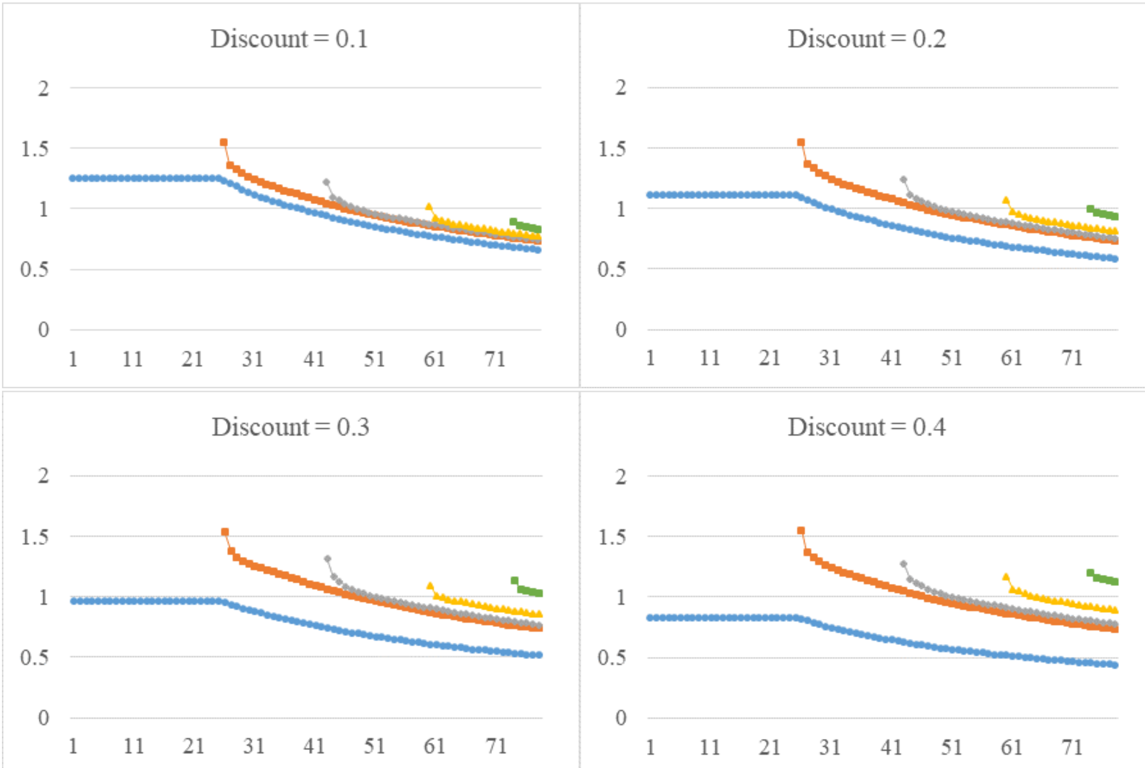


Figure 7: Trajectories of the charge per alpha values under HPOCSD with exponential overcharge

panel contains the graph of the initial quotes and the final charges corresponding to one of the four discount factors that we have tested. Recall that the initial quote is the first charge value a customer receives and is the value that the customer has to use to make the decision of whether to accept the service or not. The final charge is the price that the customer actually pays for the service. These two values are the two most important charge values. All of the values shown on the graph are on the per-alpha basis. On each graph, the horizontal axis represents the request order. The vertical axis represents the charge per alpha value. The upper series contains the initial quotes of all customers and the lower series contains the corresponding final charges. For each customer, its initial quote is always greater than or equal to its final charge, as guaranteed by the immediate response and individual rationality properties.

1. We start our analysis by focusing on the initial quote curve. When comparing the shape of the initial quote curve to that of the baseline HPOCS model, it is evident that the flat segment corresponding to advance customers is lowered and the part corresponding to realized dynamic customers is raised. As a result, the probability that an advance customer accepts its initial quote is increased if a finite willingness-to-pay value is implemented. Meanwhile,

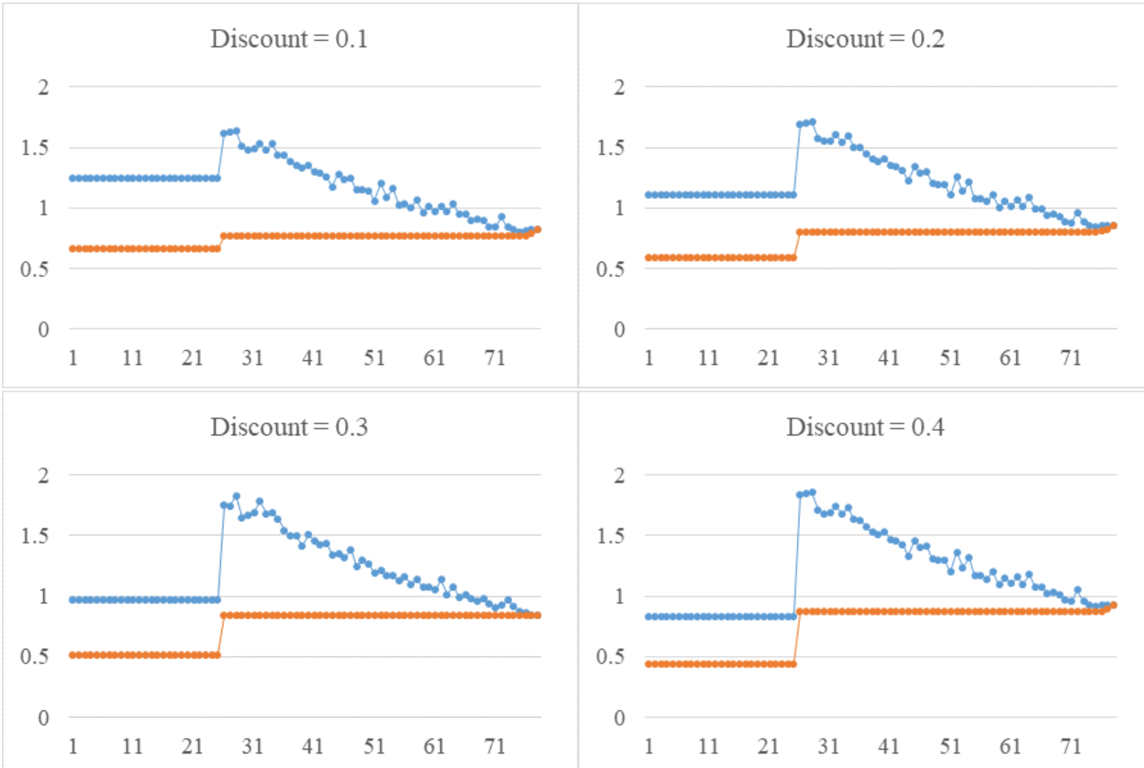


Figure 8: Initial quotes and final charges under HPOCSD with level overcharge

dynamic customers are effectively penalized and the probability that they accept their initial quotes may decrease. This phenomenon can be observed for all overcharge heuristics using any of the discount factors we have tested. In addition, all of the overcharge heuristics are shown to be more effective when a larger discount factor is used.

2. As discussed in Section 4.2.3, the online fairness property is only concerned with the final charges of customers, rather than the initial quotes. Thus it is possible for a mechanism that satisfies the online fairness property to have undesirable behavior associated with the initial quotes. Indeed, we have discovered such an issue for the baseline HPOCS mechanism based on Figure 4, namely that the initial quotes offered to late request dynamic customers may drop below that offered to advance customers. In order to correct this issue, an effective overcharge heuristic should raise the initial quotes for dynamic customers high enough such that all of them are at least as high as that offered to advance customers. Based on Figure 8, a discount of 40% is needed under the level overcharge heuristic. Figures 9 and 10 suggest that a discount of 30% is sufficient for the linear and exponential heuristics to be effective.

3. We now focus on the segment of the initial quote curve that corresponds to realized dynamic

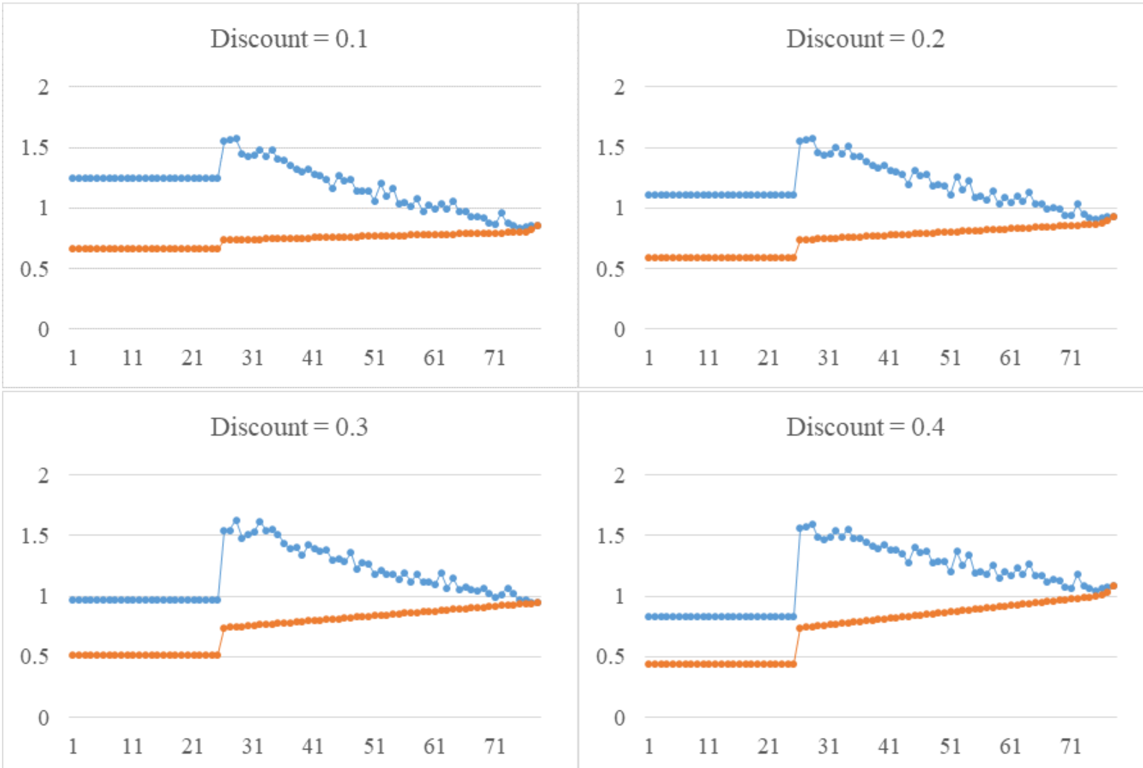


Figure 9: Initial quotes and final charges under HPOCSD with linear overcharge

customers. When using the level heuristic, this segment tends to keep its original shape as shown in Figure 4. A spike in the initial quote is caused for dynamic customers who request early, and the initial quotes for subsequent dynamic customers decrease over time. This effect can be significant when a big discount factor is used, and is clearly undesirable. On the other hand, the linear and the exponential heuristics tend to flatten the segment of the initial quote curve corresponding to realized dynamic customers, since both heuristics assign increasingly larger overcharge factors to customers who request late. When using the exponential heuristic, in particular, it can be observed that the decreasing trend can even be reversed at the tail of the initial quote curve when using a discount factor that is large enough.

4. We then analyze the effect of discounts and overcharges on the final charges. Recall that under the HPOCS mechanism, the final shared costs of all customers tend to be the same as many dynamic customers become realized, as the synergy among customers becomes too high. Figure 8, 9, and 10 show that all of the overcharge heuristics tested can prevent the advance and realized dynamic customers to have the same final charge per alpha value, even when a small discount factor is used. In particular, a jump in the final charge value can be observed

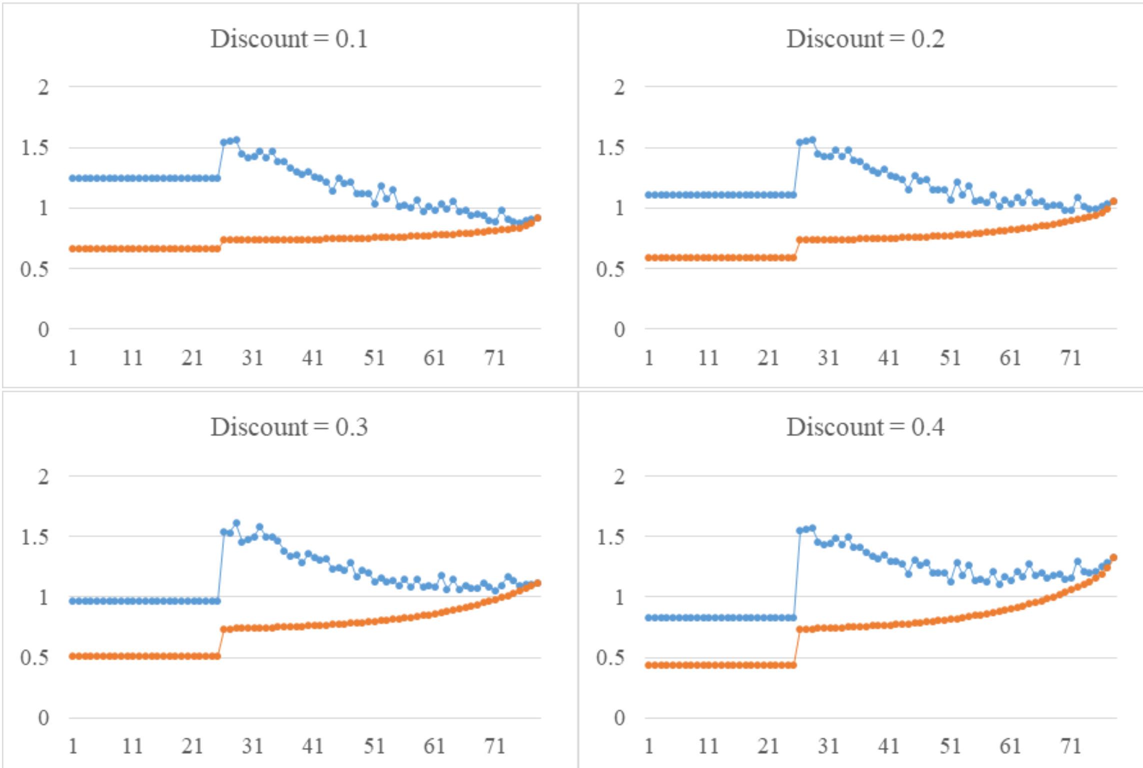


Figure 10: Initial quotes and final charges under HPOCSD with exponential overcharge

for the first dynamic customer that becomes realized. In addition, the linear and exponential heuristics cause the final charges for dynamic customers to resemble a linear and exponential pattern respectively. Both effects are more significant when a larger discount factor is used.

The simulation results discussed above suggest that larger discount factors are generally more effective in terms of promoting customers to request early. Meanwhile, based on Proposition 24, a larger discount factor could also lead to a bigger budget deficit in the worst case. Thus it is worth examining the performance of the HPOCSD mechanism on budget balance when using different overcharge heuristics and discount factors. We use the percentage of the cost recovered as the performance measure. For each realization of the problem, each overcharge heuristic, and each discount factor, we calculate the percentage of the total travel cost that can be recovered by the sum of the final HPOCSD charges for all customers that become realized. In particular, the percentage of the cost recovered $pcr$ is calculated as

$$pcr = \frac{\sum_{n=1}^{|C(T_{max})|} charge_{T_{max}} \left(\bar{\pi}_{T_{max}}(n)\right)}{totalcost(C\left(T_{max}\right))} \tag{92}$$

By definition, $pcr$ is a random variable that changes its value as the realization of the DVRP changes. $pcr = 100\%$ suggests that perfect budget balance is achieved. $pcr < 100\%$ implies a budget deficit; the total charges collected from the customers cannot recover the total operating cost. $pcr > 100\%$ means that the total charges collected exceeds the total operating cost, and a surplus is generated. Due to the nature of the dynamic vehicle routing problem, each realization of the problem may be different in terms of the actual group of customers that become realized and the request order among these customers. When positive discounts and overcharges are used, no overcharge heuristic can guarantee consistently budget balance performance. Instead, the goal is to eliminate systematic bias towards budget deficit or surplus. This can be achieved by fine tuning the parameters of each overcharge heuristic such that the budget is balanced in the average sense.

Table 9 summarizes the percentage of the cost recovered values under the level, linear, and exponential overcharge heuristics using different discount factors ranging from 0.1 to 0.4. The results shown are based on the demand scenario, where $ACPercent = 0.25$ and $RequestProb = 0.75$. We simulate each heuristic paired with each discount level on 50 realizations of the DVRP. The same set of realizations are used for all of the heuristic and discount level combinations. For each heuristic at each discount level, we report the average percentage of the cost recovered, the minimum percentage of the cost recovered among all realizations, and the maximum percentage of the cost recovered among all realizations.

| | Percentage of the Cost Recovered | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Min. | Max. | Avg. | Min. | Max. | Avg. | Min. | Max. |
| Discount | Level | | | Linear | | | Exponential | | |
| 0.1 | 100.0% | 99.3% | 100.7% | 100.1% | 99.5% | 100.7% | 100.0% | 99.4% | 100.8% |
| 0.2 | 100.0% | 98.7% | 101.4% | 100.2% | 98.9% | 101.4% | 100.1% | 98.8% | 101.5% |
| 0.3 | 100.0% | 98.0% | 102.0% | 100.2% | 98.4% | 102.2% | 100.1% | 98.2% | 102.3% |
| 0.4 | 100.0% | 97.3% | 102.7% | 100.3% | 97.8% | 102.9% | 100.2% | 97.6% | 103.0% |

Table 9: Budget balance analysis of HPOCSD for the base case

1. The average percentage of the cost recovered across all heuristics at all discount levels are close to 100%, which is the target value we use when fine tuning the model parameters. Besides, for each heuristic and discount level, the minimum and maximum percentage values of the cost recovered are generally positioned symmetrically around the corresponding mean value. Equivalently speaking, the maximum deficit and the maximum surplus incurred among all realizations are generally the same. For example, when using the level overcharge heuristic with 10% discount, the maximum deficit and surplus incurred both equal to 0.7%. This implies that the parameter settings that we use are not biased towards budget balance or

surplus, and lead to budget balanced cost allocations in general.

2. For all overcharge heuristics, there is bigger variation in the performance measure when a larger discount factor is used. For example, when using the level overcharge heuristic with 40% discount, even though the HPOCSD mechanism is generally budget balanced on average, it could incur either a 2.7% budget deficit or a 2.7% budget surplus in the worst case. If a 10% discount is used in the same heuristic, the worst-case deviations are both less than 1%. This observation confirms the discussion in Section 6.2, which argues that using larger discount factors makes the heuristic more risky and vulnerable to worst-case performance.

3. When fixing the discount level, the minimum and maximum percentage of the cost recovered do not vary significantly across different overcharge heuristics. This observation suggests that the level, linear, and exponential heuristics display similar worst-case performance.

Based on the above analysis, it can be concluded that the HPOCSD mechanism can indeed resolve the problems observed for the HPOCS mechanism, at the cost of losing the budget balance property of the original formulation. Nevertheless, the HPOCSD mechanism remains approximately budget balanced. Among the heuristics we have tested, the exponential overcharge heuristic is more effective in providing more of an incentive to make an advance or early dynamic request than the other two options.

# 7 Hybrid Proportional Online Cost Sharing with Re-optimization (HPOCSrO)

In this section, we propose to incorporate re-optimization to tackle the problem in HPOCS that the grand solution used to calculate total cost may perform poorly when the request probability is low and the number of realized customers is small since the operation cost of the grand schedule is less representative of the actual total cost. This problem will not only cause advance customers to have higher initial quotes and lose the advantage of requesting in advance, but also drive the final total cost far away from optimal, making all the customers' final cost less than ideal. In general, we address the above problem by replacing the grand solution in HPOCS with repeated re-optimization to compute the schedule that can reduce the total cost and therefore boost the overall performance of the HPOCS mechanism. However, this modification itself has a major issue of violating one of the desired properties of a well-designed cost sharing scheme, the ex-post incentive compatibility property. We first introduce the mechanism design for HPOCSrO and then we analyze the properties of this mechanism. Finally we present some experimental results showing the advantage of HPOCSrO and investigate the impact it has after losing the ex-post incentive compatibility property.

## 7.1 Mechanism Design

The HPOCSrO mechanism shares the same framework as the HPOCS mechanism, with the exception that the total cost function is defined differently. Recall in Section 5.4 that the grand schedule $\bar{S}$ is calculated based on solving a deterministic VRP problem. Differently, the HPOCSrO mechanism calculates a partial schedule initially as well as throughout the whole time horizon. The general framework of the proposed mechanism can be summarized as follows.

**Initialization.** $t = 0$. Formulate a static vehicle routing problem corresponding to the set of customers $\mathbb{AC}$ and construct the partial solution $\bar{S}(\mathbb{AC})$ using the same heuristics as the grand solution $\bar{S}$.

**Quoting advance customers.** All advance customers receive their initial quotes at time $t = 0$. This step calculates the advance cost per alpha value *acpa* and the shared cost of each advance customer using the same method as in HPOCS (see Section 5.1).

**Quoting dynamic customers.** A dynamic customer $i$ receives its initial quote when it requests service at time $t = u_i$. Customer $i$ is added into the existing partial schedule using the cheapest insertion method [13]. Then the mechanism updates the total cost and calculates the shared cost accordingly.

**Re-optimizing and updating the costs.** At each decision epoch, the same heuristics in [13] are used to optimize the current partial schedule resulting in a reduction in the total cost and the shared cost of all customers who have requested service by this decision epoch are updated.

**Final shared costs.** At time $t = T_{max}$, all of the randomness in system has been realized. The solution schedule consisting of all advance and realized dynamic customers is produced and the shared cost of these customers at time $T_{max}$ is outputted as the final cost of service for them.

## 7.2 Analysis of Property

Given that the HPOCS mechanism is proven to possess all the desirable properties discussed in Section 5.3, it follows that the HPOCSrO mechanism also possesses these properties except for the ex-post incentive compatibility property.

Recall in Section 5.3, we explain that in order for a proportional online cost sharing mechanism to satisfy all five desirable properties, its total cost function should be non-decreasing over time and be independent of the request order at any time. It is trivial to show that the HPOCSrO mechanism does not satisfy the first assumption. The total cost function over time is not an optimal solution to the current customer group but rather a good solution obtained by local search heuristics. In other words, adding a customer into the dynamic vehicle route after a re-optimization is executed may have less total cost than before. Removing this assumption will lead to the loss of ex-post incentive compatibility property which implies that if we can prove that the total cost function in the HPOCSrO mechanism satisfies the independence assumption, all the other four desirable properties are maintained [21].

**Proposition 25.** *For any partial solution $S_t$, $t \in [0, T_{max}]$ and the corresponding set of customers who have requested service $C(t)$, the special request order $\bar{\pi}_t$, and any integer $n \in [1, |C(t)|]$, the HPOCSrO total cost function totalcost$(S_t(\bar{\pi}_t(n)))$ is independent of the request order of customers $\{\pi_t(1), \ldots, \pi_t(n)\}$.*

*Proof.* The partial solution $S_t(\pi_t(n))$ is constructed by inserting a new dynamic customer using the cheapest insertion method. As a result, $S_t(\pi_t(n))$ is only concerned with the set of customers that have requested service, but not about the ordering of the requests. Therefore, for any two different orderings $\pi_t$ and $\pi_t^{'}$ containing the same $n$ customers, we have $S_t(\pi_t(n)) = S_t(\pi_t^{'}(n))$. □

Given Proposition 25, and following the same framework as in HPOCS, we can conclude that the HPOCSrO mechanism satisfies the online fairness, immediate response, individual rationality and budget balance properties.

## 7.3 Experimental Analysis

We now present simulation results to show the effectiveness of the HPOCSrO mechanism in improving the overall performance.

In order to compare the result with HPOCS, we use the same experimental setup as in Section 4.4. For HPOCSrO, the number of decision epochs which we use to re-optimize the partial solution is set at 20 which is shown to be a nice balance between identifying improvements in the solution quality and computation time [13].

The HPOCS mechanism holds all the desirable properties of a cost-sharing mechanism but could perform poorly in terms of the final shared cost when the number of dynamic customers is small, and this effect is magnified when the number of customers requesting service is small. We use the scenarios as shown in Table 10 to compare the differences of the two above routing strategies where *RequestProb* represents the probability of requesting service among dynamic customers and # of Advance Customers represents the number of advance customers in those scenarios.

| *RequestProb* | 0.25 | | 0.5 | |
|---|---|---|---|---|
| # of Advance Customers | 10 | 25 | 10 | 25 |

Table 10: Scenarios of the simulation instances

Figures 11 - 14 show graphs of the initial quote per alpha value (with legend "Initial quote") and the final shared cost per alpha value (with legend "Final price") of all customers under the two strategies in each scenario. Each graph represents a scenario and the 2 panels within the graph are the routing performances corresponding to the two strategies: HPOCS and HPOCSrO.

Based on the simulation results, we can make the following observations:

1. We first examine the initial quotes. We find that the HPOCS results exhibit a downward trend with customers who call in later having a lower initial quote than the earlier customers, favoring those who request later than advance customers as described in Section 5.4. The HPOCSrO results have a smaller slope which implies dynamic customers benefit less by delaying.

2. We then examine the final shared cost. We find that HPOCSrO has a smaller final shared

Scenario 1: # Advance Customers = 10; $RequestProb = 0.25$

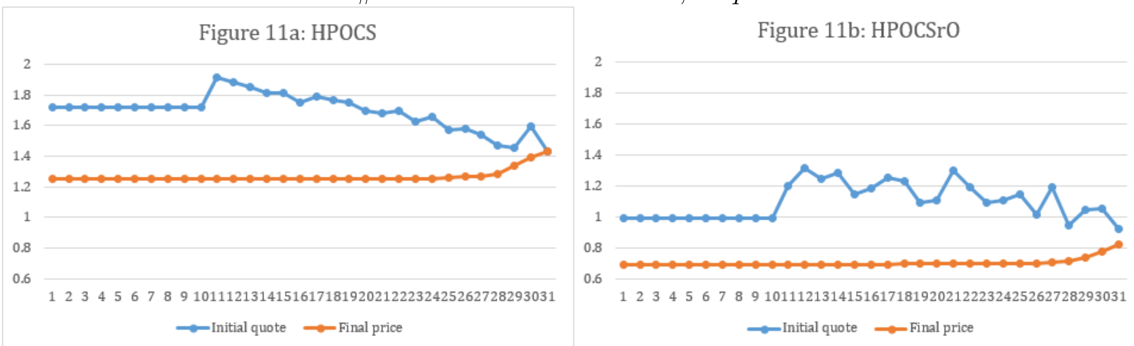


Figure 11: Initial quote and final shared cost of the two methods in scenario 1

Scenario 2: # Advance Customers = 25; $RequestProb = 0.25$

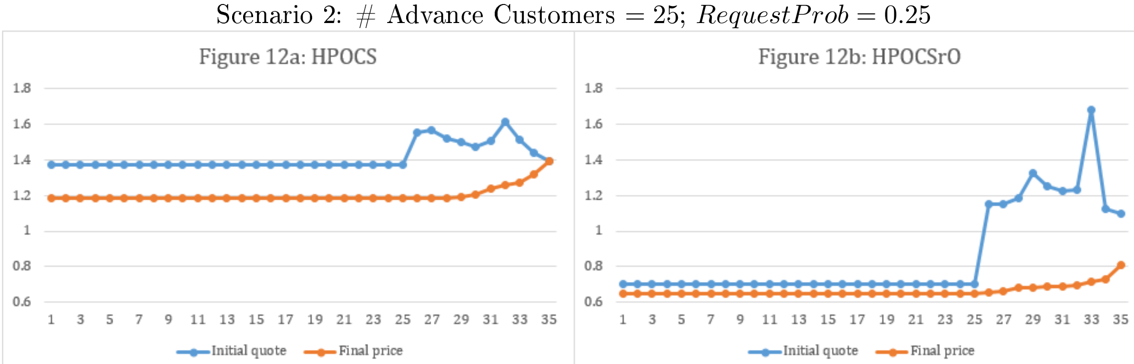


Figure 12: Initial quote and final shared cost of the two methods in scenario 2

cost indicating the efficiency of the re-optimization approach in reducing the final shared cost of each customer.

3. Next, when we fix the number of advance customers, as the probability of a dynamic customer calling in ($RequestProb$) gets higher, both methods encounter a lower final price. When we fix the probability of dynamic customers calling in ($RequestProb$), as the number of advance customers gets larger, the initial quote performance for both methods gets better in that less dynamic customers benefit from having a lower initial quote than advance customers and dynamic customers who request later will be less likely to have a lower initial quote than its predecessors. The final shared cost performance depends on the number of total customers who actually request service.

Given the above analysis, we can conclude that the HPOCSrO mechanism does help improve the overall performance of the proportional cost sharing design. However, we need to keep in mind

Scenario 3: # Advance Customers = 10; $RequestProb = 0.5$

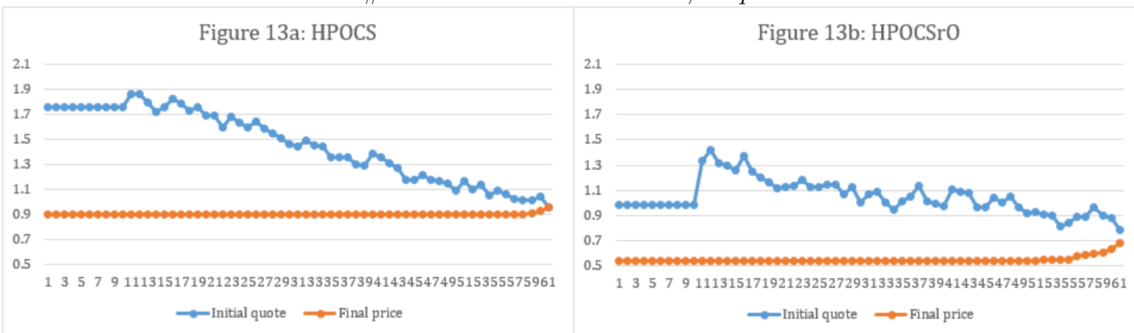


Figure 13: Initial quote and final shared cost of the two methods in scenario 3

Scenario 4: # Advance Customers = 25; $RequestProb = 0.5$

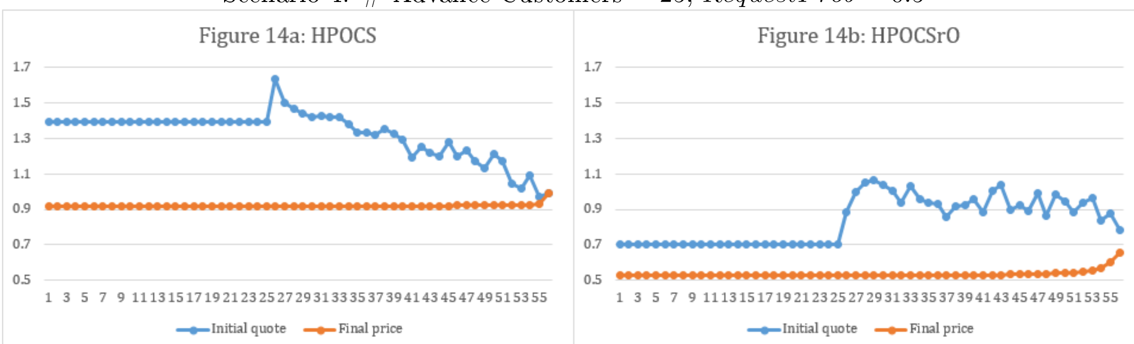


Figure 14: Initial quote and final shared cost of the two methods in scenario 4

that it suffers from the consequences of losing the ex-post incentive compatibility property which we will investigate next.

To test the impact of losing the ex-post incentive compatibility property, we look into scenarios where there are 21 dynamic customers and the number of advance customers is 0, 10, and 20 respectively. Notice that each scenario has 100 instances that shares the same generating method as the previous simulations. We then introduce the concept of **Delay Slot** which is a slot where the first dynamic customer is delayed to. For example, delay slot 6 means the previous $1^{st}$ dynamic customer is now the $6^{th}$ dynamic customer in the ex-post instance. For each scenario, all 100 instances are evaluated, and for each instance, we select 5 slots that are evenly distributed, namely the $2^{nd}$, $6^{th}$, $11^{th}$, $16^{th}$ and $21^{st}$ slots. This results in altogether $100 \times 5 = 500$ samples for each scenario. And if we aggregate all scenarios into one, the 1500 samples with 300 instances for each delay slot give us the general impact of losing ex-post incentive compatibility property regardless of scenario settings.

All scenarios are compared based on final shared cost per alpha value. The average results of the 500 samples for each scenario is displayed in Table 11. The 2nd, 3rd and 4th columns of the table display the percentage of getting a lower or higher or the same final shared cost when a dynamic customer delays its request submission. The 5th column depicts how the final shared cost for a delayed customer changes overall in each scenario. Specifically, a positive percentage indicates the final shared cost of the customer generally gets worse by delaying its submission while a negative percentage indicates the other way around. Table 11 shows that as the number of advance customers increases, both the chances of resulting in a higher final shared cost and a lower final shared cost increase. And in total, 32.1% of the time, a customer who delays its request submission shall end up with lower final shared cost while 55% of the time the cost ends up higher.

| Scenarios | % Better off | % Worse | % Same | AVG Price Change |
|---|---|---|---|---|
| 0AC_21DC | 27.2% | 50.6% | 22.2% | 8.871% |
| 10AC_21DC | 32.4% | 56.8% | 10.8% | 8.623% |
| 20AC_21DC | 36.6% | 57.6% | 5.8% | 5.921% |
| Total | 32.1% | 55.0% | 12.9% | 7.805% |

Table 11: Average gap results of 500 samples in each scenario

We next take a closer look at the detailed results for each slot. Notice that each slot has 100 instances as samples. Percentage-wise, we observe in Table 12 a downward trend after slot 11 in each scenario and an upward trend showing positive correlation between the number of advance customers and the percentage of customers getting better off if delayed until slot 16. Similarly in Table 13, we observe an upward trend of the percentage of customers getting worse off across all scenarios.

| Scenarios | Slot 2 | Slot 6 | Slot 11 | Slot 16 | Slot 21 |
|---|---|---|---|---|---|
| 0AC_21DC | 20% | 37% | 28% | 33% | 18% |
| 10AC_21DC | 30% | 39% | 39% | 31% | 23% |
| 20AC_21DC | 44% | 40% | 44% | 41% | 14% |
| Total | 31.3% | 38.7% | 37.0% | 35.0% | 18.3% |

Table 12: Percentage of customers better off in each slot across different scenarios

| Scenarios | Slot 2 | Slot 6 | Slot 11 | Slot 16 | Slot 21 |
|---|---|---|---|---|---|
| 0AC_21DC | 12% | 37% | 63% | 59% | 82% |
| 10AC_21DC | 35% | 50% | 55% | 67% | 77% |
| 20AC_21DC | 40% | 53% | 52% | 57% | 86% |
| Total | 29.0% | 46.7% | 56.7% | 61.0% | 81.7% |

Table 13: Percentage of customers worse off in each slot across different scenarios

How price change performs in the different scenarios are illustrated in Table 14. From the results, we can conclude that between slot 2 and slot 16, on average, there is an upward trend for the level of getting worse off. In other words, a customer is likely to be charged more as it delays its submission time to slot 6 or later. And when we look at the results in slot 21, the surge in final shared cost is so high that no customer is likely to delay its request to the last one.

| Scenarios | Slot 2 | Slot 6 | Slot 11 | Slot 16 | Slot 21 |
|---|---|---|---|---|---|
| 0AC_21DC | −0.433% | 0.837% | 1.803% | 2.905% | 51.782% |
| 10AC_21DC | 0.267% | 0.036% | 1.183% | 2.767% | 63.275% |
| 20AC_21DC | 0.276% | 0.706% | 1.258% | 3.147% | 47.849% |
| Total | −0.007% | 0.541% | 1.449% | 2.928% | 54.391% |

Table 14: Average price change of delayed customers in each slot across different scenarios

In general, the HPOCSrO mechanism can reduce the final shared cost of all the customers but at the expense of loss of the ex-post incentive compatibility property. For the tested scenarios, in most cases, the customers are worse off by delaying.

# 8 Implementation

This project addresses a real-time online cost sharing problem with uncertainties in customer requests. A typical application of this project is in the trucking delivery industry, i.e., companies consolidating and delivering shipments from multiple suppliers to their multiple destinations that are rather short. Trucking is the major form of transportation used in moving goods within the Los Angeles region, and has the highest level of interaction with other social functions. Methods that can encourage consolidated delivery will improve the efficiency of truck usage and possibly improve the overall logistics network and reduce traffic congestion in urban areas. In particular, this research project tackles the problem by proposing a cost-sharing mechanism that has the desirable properties to stimulate owners with small loads of goods to collaborate with others and delivering companies to provide sharing delivery services. This is inclined to mitigate truck traffic by reducing total vehicle miles and number of vehicles used through better planning when faced with uncertainties.

The implementation of our proposed mechanism in detail will require suitable programming software tools such as C++, Java, etc. It also requires access to real-world customer request data such as distance and/or travel time between facilities, and historical information on the earliest and the latest time shipments are allowed to be picked up, and the service time of processing the shipments. The entire solution framework including DVRP heuristics is implemented in C++. The same code is used to generate the experimental results presented in the report.

# 9 Conclusion and Future Directions

In this report, we study the problem of building a real-time cost sharing transportation system, which results from horizontal cooperation among multiple suppliers. In this problem, part of the customer requests are known at the beginning of the planning horizon, while the rest of the requests become realized dynamically over time. There are two major research issues closely related to the problem we study, namely the dynamic vehicle routing problem (DVRP) and cost-sharing mechanism design.

Based on the study of the DVRP, we study how the total operating cost should be allocated to each customer for a dynamic routing environment. To this end, we first define the online cost allocation problem associated with the DVRP, and discuss a list of desirable properties for online cost-sharing mechanisms. We develop the Hybrid Proportional Online Cost Sharing (HPOCS) mechanism as an online cost-sharing mechanism that combines proportional cost sharing for calculating the initial quotes for advance customers and the Proportional Online Cost Sharing (POCS) mechanism [21] for handling dynamic customer requests. The idea behind HPOCS is that customers can choose to form coalitions with customers who request service directly after them to decrease their shared costs. It is proved that the HPOCS mechanism satisfies all of the desirable properties we propose, including online fairness, budget balance, immediate response, individual rationality, and ex-post incentive compatibility.

The baseline HPOCS model is extended to two directions. One extension is to incorporate discounts for advance customers and overcharges for dynamic customers, which both help to incentivize customers to request early. The new HPOCSD mechanism is proved to be approximately budget balanced. All of the other properties of HPOCS are preserved. We compare and contrast multiple heuristics methods for calculating the overcharge factors. Simulation results show that among the three heuristics we have tested, the exponential overcharge method appears to be the most effective.The other extension is to incorporate periodic re-optimization to improve the performance on the final shared cost for the customers. In experiments across multiple scenarios, though losing the ex-post incentive compatibility property, HPOCSrO is shown to be a good mechanism design alternative to HPOCS when the $RequestProb$ is low and the number of all realized customers is small since the grand schedule in HPOCS assumes all customers request service before operating service and is therefore less representative of the actual total cost.

More work can be done along the lines of improving the HPOCS mechanism. For example, while incorporating the dynamic vehicle framework to calculate the shared costs, we can add customer forecasting to see if it can further reduce the final shared costs. Additionally, there may exist other approaches to improve the HPOCS mechanism, possibly at the cost of sacrificing one or more of the desirable properties.

## References


[1] Robert C. Anderson and Armin Claus. Cost allocation in transportation systems. *Southern Economic Journal*, 43(1):793–803, 1976.

[2] Aaron Archer, Joan Feigenbaum, Arvind Krishnamurthy, Rahul Sami, and Scott Shenker. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47(1):36–71, 2004.

[3] Sindhura Balireddi and Nelson A. Uhan. Cost-sharing mechanisms for scheduling under general demand settings. *European Journal of Operational Research*, 217(2):270–277, 2012.

[4] Yvonne Bleischwitz and Burkhard Monien. Fair cost-sharing methods for scheduling jobs on parallel machines. *Journal of Discrete Algorithms*, 7(3):280–290, 2009.

[5] Janina Brenner and Guido Sch. Cost sharing methods for makespan and completion time scheduling. *Time*, 4393(i):670–681, 2007.

[6] Janina Brenner and Guido Schäfer. Singleton acyclic mechanisms and their applications to scheduling problems. In *Algorithmic Game Theory*, volume 4997, pages 315–326. 2008.

[7] John Gunnar Carlsson and Mehdi Behroozi. Worst-case demand distributions in vehicle routing. *European Journal of Operational Research*, 256(2):462–472, 2017.

[8] Frans Cruijssen, Wout Dullaert, and Hein Fleuren. Horizontal cooperation in transport and logistics: A literature review. *Transportation Journal*, 46(3):22–39, 2007.

[9] Frans Cruijssen, Wout Dullaert, and Tarja Joro. Logistics efficiency through horizontal cooperation: The case of Flemish road transportation companies. Technical report, 2006.

[10] C. Daganzo. *Logistics Systems Analysis*. Springer, 2005.

[11] C. F. Daganzo. The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. *Transportation Science*, 18(4):331–350, 1984.

[12] Carlos F Daganzo. The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2):135–145, 1984.

[13] M.M. Dessouky and Han Zou. Efficiencies in Freight and Passenger Routing and Scheduling (METRANS Project 2-1a). Technical report, METRANS Transportation Center, 2015.

[14] Nikhil R. Devanur, Milena Mihail, and Vijay V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems*, 39(1):11–22, 2005.

[15] Julia Drechsel and Alf Kimms. Computing core allocations in cooperative games with an application to cooperative procurement. *International Journal of Production Economics*, 128(1):310–321, 2010.

[16] Julia Drechsel and Alf Kimms. Cooperative lot sizing with transshipments and scarce capacities: Solutions and fair cost allocations. *International Journal of Production Research*, 49(9):2643–2668, 2011.

[17] Miguel Andres Figliozzi, Hani Mahmassani, and Patrick Jaillet. Framework for study of carrier strategies in auction-based transportation marketplace. *Transportation Research Record*, (1854):162 – 170, 2003.

[18] Miguel Andres Figliozzi, Hani Mahmassani, and Patrick Jaillet. Competitive performance assessment of dynamic vehicle routing technologies using sequential auctions. *Transportation Research Record*, (1882):10 – 18, 2004.

[19] Miguel Andres Figliozzi, Hani S. Mahmassani, and Patrick Jaillet. Pricing in dynamic vehicle routing problems. *Transportation Science*, 41(3):302–318, 2007.

[20] M. Frisk, M. Göthe-Lundgren, K. Jörnsten, and Mikael Rönnqvist. Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2):448–458, 2010.

[21] Masabumi Furuhata, Kenny Daniel, Sven Koenig, Fernando Ordóñez, Maged Dessouky, Marc Etienne Brunet, Liron Cohen, and Xiaoqing Wang. Online cost-sharing mechanism design for demand-responsive transport systems. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):692–707, 2015.

[22] Donald B Gillies. Solutions to general non-zero-sum games. In *Contributions to the Theory of Games (AM-40)*, volume IV, pages 47–85. 1959.

[23] Donald Bruce Gillies. *Some theorems on n-person games*. PhD thesis, 1953.

[24] Jerry Green, Elon Kohlberg, and Jean Jacques Laffont. Partial equilibrium approach to the free-rider problem. *Journal of Public Economics*, 6(4):375–394, 1976.

[25] Anupam Gupta, Jochen Könemann, Stefano Leonardi, R Ravi, and Guido Schäfer. An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem. In *SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1153–1162, 2007.

[26] Anupam Gupta, Aravind Srinivasan, and Éva Tardos. Cost-sharing mechanisms for network design. *Algorithmica*, 50(1):98–119, 2008.

[27] Nicole Immorlica, Mohammad Mahdian, and Vahab S. Mirrokni. Limitations of cross-monotonic cost-sharing schemes. *ACM Transactions on Algorithms*, 4(2):1–25, 2008.

[28] Jochen Könemann, Stefano Leonardi, and Guido Schäfer. A group-strategyproof mechanism for Steiner forests. *Soda*, pages 612–619, 2005.

[29] M. A. Krajewska, Herbert Kopfer, Gilbert Laporte, Sarah Röpke, and Georges Zaccour. Horizontal cooperation among freight carriers: Request allocation and profit sharing. *Journal of the Operational Research Society*, 59(11):1483–1491, 2008.

[30] Stefano Leonardi and Guido Schäfer. Cross-monotonic cost sharing methods for connected facility location games. *Theoretical Computer Science*, 326(1-3):431–442, 2004.

[31] Konrad Lewczuk and Mariusz Wasiak. Transportation services costs allocation for the delivery system. *Proceedings - ICSEng 2011: International Conference on Systems Engineering*, pages 429–433, 2011.

[32] Gai Di Li, Dong Lei Du, Da Chuan Xu, and Ru Yao Zhang. A cost-sharing method for the multi-level economic lot-sizing game. *Science China Information Sciences*, 57(1):1–9, 2014.

[33] Peng Liu, Yaohua Wu, and Na Xu. Allocating collaborative profit in less-than-truckload carrier alliance. *Journal of Service Science and Management*, 03(01):143–149, 2010.

[34] Salvador Lozano, Pilar Moreno, B. Adenso-Díaz, and Encarnacion Algaba. Cooperative game theory approach to allocating benefits of horizontal cooperation. *European Journal of Operational Research*, 229(2):444–452, 2013.

[35] Aranyak Mehta, Tim Roughgarden, and Mukund Sundararajan. Beyond Moulin mechanisms. *Games and Economic Behavior*, 67(1):125–155, 2009.

[36] Daniel Merchan. *Effects of Urban Road Network Efficiency on Strategic Decisions in Last-Mile Logistics*. PhD thesis, MIT, 2017.

[37] Benoit Montreuil. Toward a physical internet: Meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2-3):71–87, 2011.

[38] Benoit Montreuil, Russell D. Meller, and Eric Ballot. Physical internet foundations. In Theodor Borangiu, Andre Thomas, and Damien Trentesaux, editors, *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, volume 472 of *Studies in Computational Intelligence*, pages 151–166. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[39] William H. Moore. National transportation statistics. Technical report, U.S. Department of Transportation, 2013.

[40] Hervé Moulin. Incremental cost sharing: Characterization by group strategyproofness. *Social Choice and Welfare*, 16(2):279–320, 1999.

[41] Hervé Moulin and Scott Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[42] Okan Örsan Özener and Martin Savelsbergh. Allocating cost of service to customers in inventory routing. *Operations Research*, 61(1):112–125, 2013.

[43] Martin Pál and Éva Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings*, pages 584–593, 2003.

[44] Kevin Roberts. The characterization of implementable choice rules. In *Aggregation and revelation of preferences*, volume 12, pages 321–349. 1979.

[45] Tim Roughgarden and Mukund Sundararajan. New tradeo-offs in cost-sharing mechanisms. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 79–88, 2006.

[46] Tim Roughgarden and Mukund Sundararajan. Optimal efficiency guarantees for network design mechanisms. In *Integer Programming and Combinatorial Optimization*, pages 469–483. 2007.

[47] Tim Roughgarden and Mukund Sundararajan. Quantifying inefficiency in cost-sharing mechanisms. *Journal of the ACM*, 56(4):1–33, 2009.

[48] David Schmeidler. The nucleolus of a characteristic function game. In *Game and economic theory: selected contributions in honor of Robert J. Aumann*, chapter The Nucleo, pages 231 – 238. 1995.

[49] Lloyd S. Shapley. A value for n-person games. *Annals of Mathematics Studies*, 28(2):307–317, 1953.

[50] Tage Skjoett-Larsen. European logistics beyond 2000. *International Journal of Physical Distribution & Logistics Management*, 30(5):377–387, 2000.

[51] Marius M. Solomon. Algorithms for the vehicle-routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[52] Yves Sprumont. Ordinal cost sharing. *Journal of Economic Theory*, 81(1):126–162, 1998.

[53] S. H. Tijs and T. S. H. Driessen. Game theory and cost allocation problems. *Management Science*, 32(8):1015–1028, 1986.

[54] Yun-Tong Wang and Daxin Zhu. Ordinal proportional cost sharing. *Journal of Mathematical Economics*, 37(3):215–230, 2002.