mathematics

MDPI

Article

Optimizing Fleet Size in Point-to-Point Shared Demand Responsive Transportation Service: A Network Decomposition Approach

Fudong Xie [1], Ce Wang [2] and Housheng Duan [3,*]

[1] Lingnan College, Sun Yat-sen University, Guangzhou 510275, China; xiefd@mail2.sysu.edu.cn
[2] Energy Development Research Institute, China Southern Gird, Guangzhou 510700, China; wangc53@mail2.sysu.edu.cn
[3] School of Management, Guangzhou University, Guangzhou 510006, China
* Correspondence: duanhsh@gzhu.edu.cn

Abstract: With increasing urbanization and the demand for efficient, flexible transportation solutions, demand-responsive transportation services (DTRS) has emerged as a viable alternative to traditional public transit. However, determining the optimal fleet size to balance the investment and operational revenue remains a significant challenge for service providers. In this article, we address the optimization of fleet size in point-to-point shared demand DRTS, which widely operates within many cities. To capture the uncertain passenger demands in the future when planning the fleet size currently, we model this problem with a framework of two-stage stochastic programming with recourse. Fleet sizing decisions are made in the first stage before the uncertain demands are revealed. After the uncertainty is revealed, the second stage involves making additional decisions to maximize operational revenue. The objective is to optimize the total revenue of the first-stage decisions and the expected revenue of the recourse actions. To solve this practical problem, we resort to the Model Predictive Control method (MPC) and propose a network decomposition approach that first converts the transportation network to a nodal tree structure and then develops a Nodal Tree Recourse with Dependent Arc Capacities (NTRDAC) algorithm to obtain the exact value of the expected recourse functions. In the experiments, NTRDAC is able to produce results within seconds for transportation networks with over 30 nodes. In contrast, a commercial solver is only capable of solving networks with up to five nodes. The stability tests show that NTRDAC remains robust as the problem size varies. Lastly, the value of the stochastic solution (VSS) was evaluated, and the results indicate that it consistently outperforms the expected value solutions. Numerical experiments show that the performance of the NTRDAC algorithm is quite encouraging and fit for large-scale practical problems.

Keywords: DRTS; fleet size; decomposition; algorithm; random arc capacities

MSC: 90-08

check for updates

Citation: Xie, F.; Wang, C.; Duan, H. Optimizing Fleet Size in Point-to-Point Shared Demand Responsive Transportation Service: A Network Decomposition Approach. *Mathematics* **2024**, *12*, 3048. https://doi.org/10.3390/math12193048

Academic Editor: Andrea Scozzari

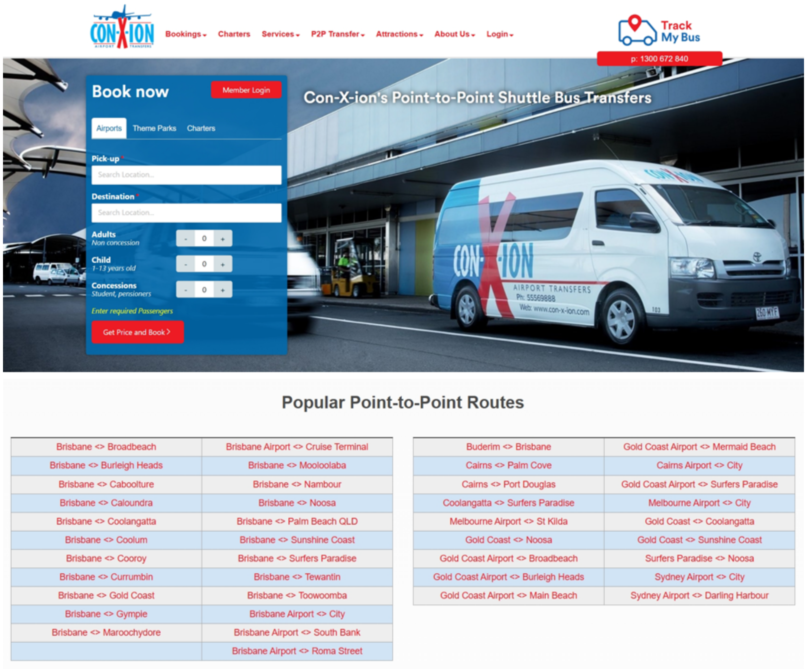
Received: 22 August 2024
Revised: 17 September 2024
Accepted: 26 September 2024
Published: 28 September 2024

1. Introduction

The rapid growth of urbanization and the increasing demand for flexible and efficient transportation solutions have highlighted the importance of demand-responsive transportation services (DRTS). Unlike traditional fixed-route transit systems, DRTS offers a dynamic and adaptive approach to public transportation, optimizing routes and schedules based on real-time passenger demands. This flexibility not only improves the user experience but also has the potential to enhance operational efficiency and reduce environmental impacts.

Demand responsive transportation, also known as flexible transportation, contains many patterns or forms of services to adapt to the demand of passengers, such as dial-a-ride [1,2], microtransit [3,4], point-to-point shuttle [5,6], feeder [7,8], ridesharing and carpooling [9,10], ride-hailing [11,12], etc. [13,14]. Among these service forms, point-to-point shuttle services have soared recently in practice. These services provide customers with direct routes between major transit points, benefiting service operators greatly by allowing them to select only feasible, well-defined locations where safe boarding and alighting are possible [15]. Furthermore, these services are designed for passengers to enhance connectivity within the transportation network, making it easier to transfer between transit points, such as airports, train stations, bus terminals, local attractions, central business district areas (CBDs), residential areas, etc. According to the research by Kjoerstad and Renolen [16], customers exhibit a high valuation of their time when utilizing transportation services. Their study results show that customers prefer using direct transit modes 1.8 to 5.0 times more than non-direct ones that require a 5 min wait at an interchange point. This preference increases to 2.5 to 9.2 times if the waiting time reaches 10 min. Before launching a point-to-point DRTS, service providers must make a critical decision on fleet size—determining how many vehicles to invest in—while considering current investment and potential future operational revenue.

Real-world examples regarding point-to-point shuttle services can be seen in many cities. For instance, in Zhuhai, China, a service provider, Zhuhai Airport Express (https://www.51sfbus.com/sflv-plus/#/rentcar/search?btntabModel=0 accessed on 20 July 2024 via WeChat browser), offers point-to-point on-demand transportation between several points within Zhuhai City, such as Chimelong Ocean Kingdom (local attraction), Gongbei Port (transportation terminal of railway and bus), Jida (Central Business District), and several resident areas. In Australia, Con-x-ion (https://www.con-x-ion.com/point-to-point accessed on 20 July 2024) provides many point-to-point routes, mostly from Brisbane Airport to many local attractions nearby. Figure 1 is a screenshot from Con-x-ion's website showing the mini-van used for this service and their popular service routes.



Popular Point-to-Point Routes

Brisbane <> Broadbeach	Brisbane Airport <> Cruise Terminal
Brisbane <> Burleigh Heads	Brisbane <> Mooloolaba
Brisbane <> Caboolture	Brisbane <> Nambour
Brisbane <> Caloundra	Brisbane <> Noosa
Brisbane <> Coolangatta	Brisbane <> Palm Beach QLD
Brisbane <> Coolum	Brisbane <> Sunshine Coast
Brisbane <> Cooroy	Brisbane <> Surfers Paradise
Brisbane <> Currumbin	Brisbane <> Tewantin
Brisbane <> Gold Coast	Brisbane <> Toowoomba
Brisbane <> Gympie	Brisbane Airport <> City
Brisbane <> Maroochydore	Brisbane Airport <> South Bank
	Brisbane Airport <> Roma Street

Buderim <> Brisbane	Gold Coast Airport <> Mermaid Beach
Cairns <> Palm Cove	Cairns Airport <> City
Cairns <> Port Douglas	Gold Coast Airport <> Surfers Paradise
Coolangatta <> Surfers Paradise	Melbourne Airport <> City
Melbourne Airport <> St Kilda	Gold Coast <> Coolangatta
Gold Coast <> Noosa	Gold Coast <> Sunshine Coast
Gold Coast Airport <> Broadbeach	Surfers Paradise <> Noosa
Gold Coast Airport <> Burleigh Heads	Sydney Airport <> City
Gold Coast Airport <> Main Beach	Sydney Airport <> Darling Harbour

Figure 1. Vehicles and popular point-to-point routes of Con-x-ion.

Con-x-ion's point-to-point routes create a DRTS network, allowing passengers to travel between various transit points. For example, consider five transit points: Brisbane, Noosa, Sunshine Coast, Caloundra, and Mooloolaba. In this DRTS network, passengers can travel between any two points, with each point serving as both a starting and an ending location. From each starting point, there are four direct routes connecting to the other four points. This DRTS network is depicted in Figure 2.

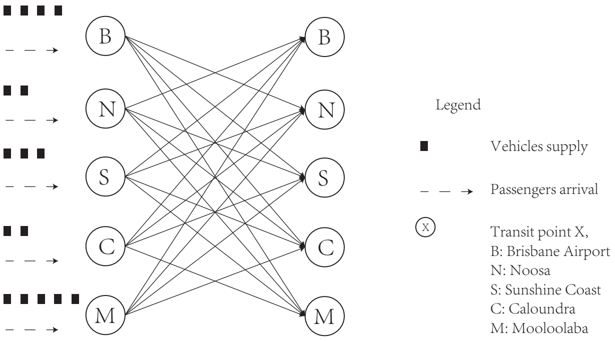


Figure 2. Network structure of Con-x-ion's point-to-point service.

A critical challenge in implementing DRTS, especially for strategic planning, is determining the optimal fleet size [17]. An inadequately sized fleet can lead to service delays, increased operational costs, and unsatisfied customer demand, while an oversized fleet may result in underutilization of resources and unnecessary expenditures. Therefore, optimizing fleet size is essential to balance service quality with cost-effectiveness. In this study, we use the framework of two-stage stochastic programming with recourse to capture the feature of DRTS where future demands are uncertain when planning the fleet size in the first stage. The first-stage decision involves selecting an initial fleet size, while the second-stage decisions adjust operational strategies based on realized demand scenarios. To ensure that the system responds effectively to demand fluctuations, we integrate control mechanisms that manage fleet deployment dynamically. These control strategies are embedded in the second stage, where operational adjustments are made in response to actual demand, which is unknown during the initial fleet sizing stage. The stochastic nature of demand is modeled through a set of possible future demand scenarios, estimated using historical data and forecasting techniques. This ensures that the fleet size determined in the first stage is robust against various demand patterns, with the flexibility to adjust operations in the second stage based on the actual demand realization. Furthermore, for the aforementioned real-world problems , the large number of transit points where these shuttle services operate would make solving this large-scale practical stochastic programming problem very time-consuming or even infeasible to be optimal. To tackle this problem, we introduce a network decomposition method tailored to efficiently obtain the exact value of the expected recourse functions.

The contributions of this paper are threefold: (i) To the best of our knowledge, this is the first work to address the large-scale practical problem of optimizing fleet size in DRTS, considering vehicle investment and operational revenue resulting from uncertain passenger demands. (ii) The network decomposition method presented in this article relaxes the nodal tree recourse algorithm by allowing the arcs' capacities, which are random variables, to be dependent. (iii) This article presents a pragmatic practice bridging the gap between theory and real-world application.

The rest of this paper is organized as follows. Section 2 reviews the related literature. Section 3 introduces the notation and describes the two-stage stochastic formulation. Section 4 details the network decomposition method, and numerical experiments are provided in Section 5. Finally, Section 6 concludes this paper and discusses several extensions and future research directions.

2. Literature Review

Fleet size optimization problems have recently attracted much interest in the research domain, particularly within DRTS. These services, offering flexible routing and scheduling based on real-time uncertain demand, present unique challenges and opportunities for service providers in fleet management. In this section, we review the literature related to fleet sizing problems and classify the literature based on their objective and the literature regarding the solution technique to stochastic programming.

The first stream of papers investigates the correlation between fleet size and other parameters. Daganzo and Ouyang [18] present a general analysis framework for systems providing non-shared and shared services. With agent-based simulations, they investigate the relationship between travel time and fleet size at different demand density levels. Their research insights can be used to explore operation and pricing strategies for taxi companies and government agencies. Hörl et al. [19] study the performance of different fleet sizes and operational policies based on simulation scenarios by MATSim 0.10. Their results indicate that a fleet size of 7000–14,000 vehicles in Zurich, Switzerland, could serve 90% of requests. In these articles, fleet size is important but only serves as an exogenous variable or model parameter instead of a decision variable.

The second stream of research focuses on optimizing fleet size using different models and methods, where fleet size is one of the model decision variables. Shehadeh et al. [20] study the fleet size and allocation optimization problem for on-demand last-mile transportation services under passenger demand uncertainty. They propose and analyze a two-stage stochastic programming model and a robust optimization model where the two-stage stochastic programming model is solved by the sample average approximation (SAA) approach. They resort to the SAA approximation approach because obtaining an optimal solution to the stochastic programming model is complex. Beaujon and Turnquist [21] also investigate a fleet size and allocation problem under uncertainty where the uncertainties are related to demands of loaded vehicles and travel time between origin points to destination points. To solve the problem, they propose a network approximation method to compute the probability density functions of net vehicles at period t, where they argue that the normal probability density function can represent it. The pitfall of this model is that it ignores the demand uncertainty of passengers, which is the core feature of DRTS. Winter et al. [22] present an automated demand-responsive transport system (ADRTS) for a campus–train station shuttle service to evaluate the system's performance. They compute the required minimal fleet size while achieving expected results, such as cost savings and reduction in passenger waiting time. The model only considers a single pick-up and drop-off station, and the method they apply to obtain this result is simulation, so the result is not promising for practical DRTS problems. Diana et al. [23] study the problem of determining the number of vehicles needed for DRTS with predetermined quality for the customers in terms of waiting time at the stops and maximum allowed detours. They formulate the problem with a probability model and intend to analyze the impact of fleet size on the customers' waiting time. The objective is to provide an estimation of the required size of the fleet in different scenarios. The critical assumption of their paper is that there is no sharing among different customers for vehicle capacity, where the fleet has to be dispatched exclusively based on the list of requests, such as in the taxicab system, which is different from that of shared DRTS. Militão and Tirachini [24] introduce an optimal fleet size problem for a shared door-to-door DRTS to minimize the total costs for operators and customers. The decision variables are vehicle capacity and fleet size for human-driven and automated vehicle operations. They analyze the relevant variables using data from a large-scale agent-based simulation (ABS) applied to Munich. However, the outcomes of ABS can be susceptible to initial conditions and parameter settings, making it challenging to validate and generalize results. Furthermore, ABS can be computationally intensive, particularly when simulating large numbers of agents or when the model includes detailed interactions and stochastic elements. Even though fleet size is one of the decision variables in their models, these models do not consider the investment associated with the fleet.

Stochastic programming (SP) is a well-established mathematical framework for optimizing decision making under uncertainty. Model predictive control (MPC) is an advanced control strategy commonly used in dynamic systems where decisions are made based on the prediction of future system behavior over a finite horizon [25]. The key idea is to use a mathematical model of the system to predict its future states and solve an optimization problem at each decision step. In stochastic programming, uncertainty is modeled explicitly, typically through probabilistic scenarios or distributions. The optimization considers multiple possible future scenarios, incorporating uncertainty into the decision-making process. The complexity of these problems, mainly due to numerous scenarios (known as the curses of dimensionality [26]), has driven the development of various solution techniques. The techniques for solving stochastic programs can be classified into two groups.

The first group of methods aims to speed up the convergence or reduce the number of scenarios. These methods typically turn the original problem into huge deterministic problems by using the samples from the distribution of the random coefficients, such that different techniques have been developed to deal with convergence. For example, Benders decomposition, also known as the L-shaped method, first divides the original problem into a master problem and a subproblem, then feasibility and/or optimality cuts are imposed iteratively to obtain the optimal solution [27,28]. Iterations exist in the scenario aggregation method. To compute an implementable solution, this method typically starts with a relatively small subset of scenarios and includes additional constraints iteratively [29,30]. For the stochastic decomposition methods, their objective is to obtain the lower bounds of the expected recourse function using cutting planes [31,32]. The above-mentioned methods are usually used to accelerate convergence speed and are generally not practical for large-scale dynamic network problems. Readers are recommended to refer to Birge and Louveaux [33] for details. On the contrary, another technique, called scenario reduction, aims to reduce the problem size instead of solving large deterministic problems. Dupačová et al. [34] present the idea of scenario reduction, aiming to generate a natural probability metric as an approximation metric to obtain the closest subset from a larger superset of scenarios. However, ensuring the quality of solutions with the reduced scenarios is the most challenging, such that different methods related to scenario reduction on measuring the similarity between the original set and the reduced set have been suggested [35,36].

The second group of methods leverages the special structure of stochastic problems, especially those related to network recourse problems. The strategy of these methods is first to decompose the original problem into several tractable subproblems and then solve those subproblems instead of tackling the original one. For example, Wallace [37] introduces a method of using piecewise linear functions to approximate the expected recourse function for the network recourse problem. Powell and Cheung [38] introduce a TREC algorithm to obtain the exact value of the expected recourse function as a scalar supply of resources for a tree recourse problem. Their experiments show that the TREC algorithm fits large-scale practical problems. After that, they present a network recourse decomposition approach to applying the TREC algorithm for a two-stage dynamic network problem [39]. However, the TREC algorithm's condition is that each arc's random capacity must be mutually independent. In this article, we adopt the strategy from the TREC algorithm and propose our algorithm, which can handle dependent random arc capacities.

In summary, while some articles have explored fleet sizing optimization, many do not account for both the investment required for the fleet and the uncertainty of demand in their models. Moreover, the development of practical algorithms for obtaining the exact value of the expected recourse functions, particularly for large-scale, real-world DRTS problems, has not been previously addressed. The innovations of this study are summarized in Table 1.

Table 1. A comparison between our research and existing studies for optimizing fleet size.

Author(s)	Investment Incorporated	Obtain Exact Value	Insensitive to Input Data	Stable to Problem Size
Militão and Tirachini [24]	✓	×	×	✓
Diana et al. [23]	×	✓	×	✓
Winter et al. [22]	×	×	✓	✓
Beaujon and Turnquist [21]	×	✓	×	✓
Shehadeh et al. [20]	✓	×	✓	×
Our study	✓	✓	✓	✓

3. Problem Formulation

Consider a transportation service provider who plans to operate a point-to-point shuttle service within a city. Assume the service provider has chosen $\mathcal{N}$ well-defined locations, such as airports, train or bus stations, and local attractions, as transit points. We also assume that the fare from point $i \in \mathcal{N}$ to point $j \in \mathcal{N}$ for each customer, denoted as f_{ij}, has been set. Passengers place orders on their mobile phone, and their arrival at each transit point $i \in \mathcal{N}$ destined to point $j \in \mathcal{N}$, denoted as d_{ij}, is the demand for shuttle service, which is a random variable. Its distribution is known to the service provider. To fulfill these random demands, the provider plans to purchase s identical vehicles, each of which has H available seats, which we call vehicle capacity. The challenge to this service provider is determining the optimal s to maximize the profit, which is the revenue from serving customers minus the investment in these s vehicles.

This fleet size optimization problem can be modeled as two-stage stochastic programming with recourse. The decision to determine s, the fleet size, is called the first-stage investment decision, which is to be taken without full information on the random customer demands $d_{ij}, i \in \mathcal{N}, j \in \mathcal{N}$. Later, full information is received on the realization of random demands. Then, second-stage or dispatching actions $x_{ij}, i \in \mathcal{N}, j \in \mathcal{N}$ are taken, which generates revenue from these s vehicles. We first state our assumptions before presenting the mathematical model.

Assumption 1. *Order rejections are allowed for service providers.*

Assumption 2. *The distribution of random demands and the depreciation of each vehicle are considered within the same period, such as an hour.*

Assumption 3. *Decisions are made only once within a time period.*

The first assumption says that the service provider is allowed to reject orders. This mechanism is because passengers have many alternatives in transportation, and the private sector operates the shuttle service, so maximizing revenue is one of the critical objectives. Recently, many studies have investigated this mechanism [24,40,41]. The second assumption is made to simplify our presentation. Since the distribution of random demands has to be estimated for a certain period, having the depreciation of each vehicle representing the investment for the same period will significantly facilitate our model and computation. The third assumption is to keep the network size manageable and avoid the details of making decisions multiple times within a time period [42].

Let $\omega \in \Omega$ be an outcome of a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ for the passenger demands in the second stage. The parameters and variables for the two-stage stochastic programming model are summarized in Table 2.

Table 2. Notations.

		Description
Parameters	$\mathcal{N}$	Set of transit points, $\mathcal{N} = \{1, 2, ..., N\}$.
	H	Available seats per vehicle, also known as vehicle capacity.
	g	The depreciation of each vehicle for a certain period.
	c_{ij}	Cost of each trip from point i to point j for each vehicle.
	f_{ij}	Fare of each passenger from point i to point j.
	$d_{ij}(\omega)$	Passenger demands from point i to point j under realization ω for a certain period.
Intermediate variable	$v_i(\omega)$	Available vehicle at point i under realization ω.
Decision variables	s	Fleet size in first stage.
	$x_{ij}(\omega)$	Number of vehicles dispatched from point i to point j in second stage under realization ω.

To ease our presentation, we define the available vehicle vector $\boldsymbol{V} = (v_i : i \in \mathcal{N})$ and the passenger demands vector $\boldsymbol{D}(\omega) = (d_{ij}(\omega) : i \in \mathcal{N}, j \in \mathcal{N})$. The two-stage fleet size optimization problem can be formulated as

$$max \quad -g \cdot s + \mathbb{E}_{\omega}[Q(\boldsymbol{V}, \boldsymbol{D}(\omega))], \tag{1}$$

subject to

$$\sum_{i=1}^{N} v_i = s, \tag{2}$$

$$s \geq 0, s \in \mathbb{Z} \tag{3}$$

where for a given available vehicle vector $\boldsymbol{V}$ and a particular realization ω, $Q(\boldsymbol{V}, \boldsymbol{D}(\omega))$ is the value of a maximization problem which is defined as follows:

$$Q(\boldsymbol{V}, \boldsymbol{D}(\omega)) = max \quad \sum_{i=1}^{N}\sum_{j=1}^{N}\left(-c_{ij} \cdot x_{ij}(\omega) + f_{ij} \cdot min\{H \cdot x_{ij}(\omega), d_{ij}(\omega)\}\right), \tag{4}$$

subject to

$$\sum_{j=1}^{N} x_{ij}(\omega) = v_i, \quad \forall i \in \mathcal{N}, \tag{5}$$

$$x_{ij}(\omega) \geq 0, x_{ij}(\omega) \in \mathbb{Z}, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{N}. \tag{6}$$

The problem defined by Equations (1)–(6) is a typical two-stage stochastic program with network recourse, where (1)–(3) are called the first-stage problem and (4)–(6) are called the second-stage problem. The expectation functional $\mathbb{E}_{\omega}[Q(\boldsymbol{V}, \boldsymbol{D}(\omega))]$ is called the expected recourse function. The objective function (1) is profit maximization, which consists of the depreciation of vehicles and the expected recourse revenue. Constraint (2) is to force the total available vehicles to be equal to the fleet size, and (3) is the non-negativity constraint. The state vector $\boldsymbol{V}$ stores the available vehicles for stage two, and it is the only variable that communicates information to the second stage. The expected recourse function (4) is determined by solving the second-stage problem for each scenario ω according to the available vehicle supplies $\boldsymbol{V}$. Constraint (5) enforces that the total number of vehicles dispatched from point $i \in \mathcal{N}$ must be those vehicles available at point $i \in \mathcal{N}$, and Constraint (6) ensures that the number of dispatched vehicles must be a positive integer. It is worth noting that to deal with the expression $min\{H \cdot x_{ij}(\omega), d_{ij}(\omega)\}$, we just need to replace this expression with intermediate variables $q_{ij} \geq 0$ and add two constraints, $q_{ij} \leq H \cdot x_{ij}(\omega)$ and $q_{ij} \leq d_{ij}(\omega)$, where $\forall i \in \mathcal{N}, \forall j \in \mathcal{N}$. In this way, problems (4)–(6) become common recourse problems.

Finding exact optimal solutions to stochastic programs is generally intractable due to the uncertain demands that increase the number of scenarios exponentially. In this point-to-point shuttle service case, if the service provider operates at $N = 10$ transit points, which generates 9 shuttle service routes in each point $i \in \mathcal{N}$ (from point i to other points $j \in \mathcal{N}$ except point i itself), the total number of shuttle service routes is 90. Assume that each random passenger demand, $d_{ij}, i \in \mathcal{N}, j \in \mathcal{N}$, is a discrete random variable and takes four possible values on average. The total number of scenarios is $4^{90} = 2^{180}$, which is intractable. In this article, we resort to the model predictive control method (MPC), where a decision is made at time t by solving a typically approximate model over a horizon $(t, t + L)$. Specifically, in our model, to determine the fleet size s in the first stage, we have to evaluate the value function $\mathbb{E}_{\omega}[Q(\boldsymbol{V}, \boldsymbol{D}(\omega))]$ parameterized in s in the second stage. In the following Section 4, we present a network decomposition method tailored for this type of network recourse problem to efficiently obtain the exact value of the expected recourse functions. The workflow of the network decomposition method, consisting of four steps, is illustrated in Figure 3. Step 2, solving a single-root-node problem, is to compute $\mathbb{E}_{\omega}[Q_i(v_i, \boldsymbol{D}(\omega))]$,

$\forall i \in \mathcal{N}$. Repeating step 2 for all nodes $i \in \mathcal{N}$ can obtain $\mathbb{E}_\omega[Q(\boldsymbol{V}, \boldsymbol{D}(\omega))]$, which is the main purpose of step 3.

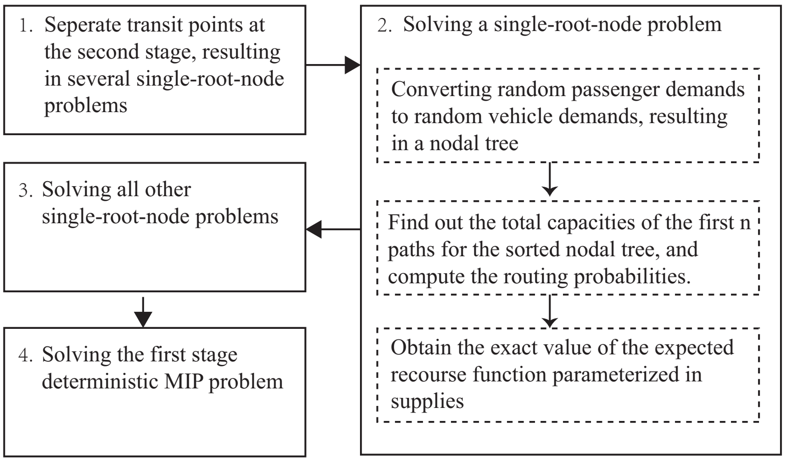

1. Seperate transit points at the second stage, resulting in several single-root-node problems
2. Solving a single-root-node problem
Converting random passenger demands to random vehicle demands, resulting in a nodal tree
Find out the total capacities of the first n paths for the sorted nodal tree, and compute the routing probabilities.
Obtain the exact value of the expected recourse function parameterized in supplies
3. Solving all other single-root-node problems
4. Solving the first stage deterministic MIP problem

Figure 3. The workflow of the network decomposition method.

4. Recourse Problem Decomposition

This section elaborates on the network decomposition method tailored for network recourse problems. First, in Section 4.1, we analyze the separability of the network, which enables us to deal with each transit point separately. Second, in Section 4.2, for completeness, we review an idea of the nodal tree algorithm which we leverage to obtain the exact value of the expected recourse function for a nodal tree structure recourse problem. Third, in Section 4.3, we elaborate on our algorithm extended from the nodal tree algorithm. Finally, we present the solution technique to the first-stage problem in Section 4.4.

4.1. Separability of Transit Points

It is reasonable to assume that passengers' arrival to transit points (transit points and nodes are interchangeable in this article) are independent, and the random demands, $d_{ij}, i \in \mathcal{N}, j \in \mathcal{N}$, are independent discrete random variables too. The rationale behind this assumption is that in a real-world application, a common service provider often serves a large number of independent passengers [42]. For the second-stage recourse problem, this independence enables us to decompose the network recourse problem by transit points, meaning we can treat each transit point separately. With this separability property, the expectation expression in Equation (1) can be rewritten as

$$\mathbb{E}_\omega[Q(\boldsymbol{V}, \boldsymbol{D}(\omega))] = \sum_{i=1}^{N} \mathbb{E}_\omega[Q_i(v_i, \boldsymbol{D}(\omega))]. \tag{7}$$

Equation (7) provides a favorable method to obtain the recourse function's expected value, enabling us to handle individual transit points separately. Since each transit point $i \in \mathcal{N}$ is identical, we focus on one of the transit points $i \in \mathcal{N}$ only, which means we can turn to solving the single-root-node recourse problem rooted at node $i \in \mathcal{N}$. The structure of this single-root-node recourse problem is illustrated in Figure 4.

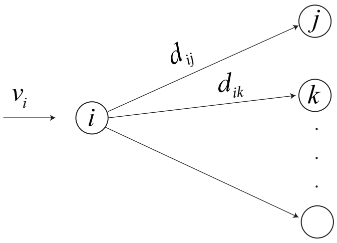


Figure 4. The structure of a single-root-node recourse problem.

4.2. Solution Technique to Nodal Tree Recourse Problems

This subsection briefly reviews the solution technique to nodal tree recourse problems introduced by Powell and Cheung [38], which we leverage to solve our single-root-node recourse problem.

A nodal tree recourse problem is defined as follows. Consider a directed one-level tree (we say a tree has n levels if n is the depth of the tree) rooted at node r with independent random arc capacities ξ_i such as the one depicted in Figure 5. The total supply to this tree, entering the root node, is S. For a given realization ω, a unit of flow traversing a path (link connecting the root node to a leaf node) will result in a certain cost c_i. Assume all m paths have been ranked by their costs in ascending order, meaning $c_1 < c_2 < ... < c_m$. We wish to find the expected total cost of the tree problem as a function of the scalar supply.

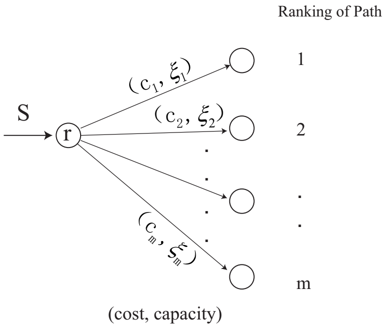


Figure 5. A nodal tree problem with random arc capacities.

Let us define

$$f_n = \text{flow on path } n.$$

For a given realization ω, the nodal tree recourse problem is

$$Q(S, \xi(\omega)) = \min_{f_n(\omega)} \quad c^T f(\omega), \tag{8}$$

subject to

$$\sum_{n=1}^{m} f_n(\omega) = S, \tag{9}$$

$$0 \le f_n(\omega) \le \xi_n(\omega), \quad n = 1, 2, ..., m. \tag{10}$$

To solve the problems (8)–(10), assigning flow to the paths in their ranking greedily can obtain the optimal solution. This optimal policy enables us to solve the problem efficiently. Define the following:

$\phi(k, n) = P$ {the kth unit of flow entering node r takes the nth path}, which we call a routing probability;
$G(k)$ = expected marginal cost for the kth unit of flow entering node r;
Z^n = the total capacity of the first n ranked paths.

Therefore, Z^n can be computed by

$$Z^n = \sum_{i=1}^{n} \xi_i, \tag{11}$$

and $\phi(k, n)$ can be obtained by

$$\phi(k, n) = P\{(Z^n \ge k\} - P\{Z^{n-1} \ge k\}. \tag{12}$$

With the routing probability $\phi(k,n)$, $G(k)$ can be calculated by

$$G(k) = \sum_{n=1}^{m} c_i \cdot \phi(k,n). \tag{13}$$

Finally, the expected value of the recourse problem, $\hat{Q}(S)$, can be obtained easily by

$$\hat{Q}(S) = \sum_{k=1}^{S} G(k). \tag{14}$$

Equation (11) holds under the condition that random arc capacities are mutually independent. The theory behind Equation (12) is that the greed algorithm holds for this nodal tree problem, meaning that the probability of the kth unit taking the nth path is the one that the total capacity of the first n paths must be greater or equal to k, and the total capacity of the first $n-1$ paths must be less than k. Mathematically, it is $\phi(k,n) = P\{Z^n \geq k \quad \text{and} \quad Z^{n-1} < k\}$. Since the random arc capacities are mutually independent, such that the probability of $Z^n \geq k$ and $Z^{n-1} < k$ is independent, this means that $\phi(k,n) = P\{Z^n \geq k\} \cdot P\{Z^{n-1} < k\} = P\{Z^n \geq k\} \cdot (1 - P\{Z^{n-1} \geq k\})$. Finally, suppose the total capacity of the first $n-1$ paths is greater than or equal to k, which indicates that the total capacity of the first n paths must be greater than or equal to k (with probability of 1), such that $P\{Z^n \geq k\} \cdot P\{Z^{n-1} \geq k\} = P\{Z^{n-1} \geq k\}$. For proof details, we recommend readers refer to [38].

The solution to the nodal tree recourse problem provides an efficient method for calculating the expected value of the recourse function. In Section 4.3, we propose a process to convert the single-root-node recourse problem to a nodal tree structure.

4.3. Conversion of Single-Root-Node Recourse Problems

This subsection presents the procedure to convert the single-root-node recourse problem to the nodal tree structure. Three factors need to be considered for a nodal tree problem. The first is the supply to the root node, the second is the revenue of each path, and the third is the random capacity of the corresponding path. Since the vehicles v_i are scalar supplies entering into the root node, we focus on how to compute the revenue and capacity for each path.

To begin with, we present the converting processes with a single leaf node $j \in \mathcal{N}$, which means we first focus on the path from root node i to leaf node j. The passenger demand on this path is d_{ij}, a discrete random variable. The fare for each passenger is f_{ij}, and the cost for each vehicle traversing this path is c_{ij}.

We start with the computation of the path revenue. Vehicles carrying $h = 1, 2, ..., H$ passengers departed from node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$ generate different revenue, denoted by $r_{ij}(h)$, which can be calculated by $r_{ij}(h) = h \cdot f_{ij} - c_{ij}$, where $h = 1, 2, ..., H$ and H is the total available seats of each vehicle. It is natural to consider creating H paths rooted at node $i \in \mathcal{N}$ pointing to node j, such that the revenue of each path can be determined by $r_{ij}(h), h = 1, 2, ..., H$, respectively. Duplicating the leaf node $j \in \mathcal{N}$ $H-1$ times creates H leaf nodes $j_h, h = 1, 2, ..., H$ and generates a nodal tree. This nodal tree structure, rooted at node $i \in \mathcal{N}$ and ended in leaf nodes $j_h, h = 1, ..., H$ with total H paths and corresponding revenue $r_{ij}(h)$, is depicted in Figure 6.

Now, we turn to compute each path's capacity, denoted by $\xi_{ij}(h)$, which is a random variable capturing the number of vehicles carrying $h = 1, 2, ..., H$ passengers given the passenger random demand d_{ij}. The policy of assigning passengers to vehicles determines the number of vehicles used and the number of passengers carried in each vehicle. Since our objective is to maximize operational revenue, a passenger assignment policy exists to achieve this, as stated in Proposition 1.

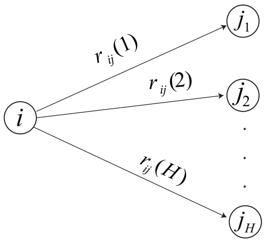


Figure 6. A nodal tree structure with H paths with corresponding path revenue $r_{ij}(h), h = 1, 2, ..., H$.

Proposition 1. *For any realization ω, given passenger demands, the optimal policy of passenger assignment to vehicles is to assign more passengers to a departing vehicle until its seats are all occupied.*

Proof. Assume there are $d_{ij}(\omega), i \in \mathcal{N}, j \in \mathcal{N}$ passengers waiting at transit point $i \in \mathcal{N}$ under realization ω, and assume there are $h = 0, 1, 2, ..., H-1$ passengers having been assigned into vehicle k. The revenue of this transportation is $r_{ij}(h) = f_{ij} \cdot h - c_{ij}$. If we assign additional $\Delta h > 0$ passengers into vehicle k, the marginal revenue $MR(P_{ij}) = \frac{r_{ij}(h+\Delta h) - r_{ij}(h)}{\Delta h} = f_{ij} \geq 0$. It means assigning more passengers into this vehicle k can achieve positive marginal revenue, such that the optimal policy is to assign more passengers into a vehicle until it is fully loaded. □

Proposition 1 indicates, given passenger demands $d_{ij}(\omega), i \in \mathcal{N}, j \in \mathcal{N}$ under realization ω, if $k \geq 1$ vehicles have been dispatched from point i to point j, there is at the most one of these k vehicles which is not fully loaded. So, the number of fully loaded vehicles under realization ω can be obtained by $\xi_{ij}(H) = \lfloor \frac{d_{ij}(\omega)}{H} \rfloor$, where $\lfloor X \rfloor$ means to round down X, and H is the available seats of a vehicle. The number of the not fully loaded vehicles is 1, which carries the remainder of passengers $\xi_{ij}(h) = d_{ij}(\omega) \ mod \ H$, where operator "*mod*" is the modulo operation, and $h \in \{1, ..., H-1\}$. In this way, passenger demands $d_{ij}(\omega)$ can be converted to vehicle demands under any realization ω, which can be expressed mathematically as

$$\xi_{ij}(h) = \begin{cases} \lfloor \frac{d_{ij}(\omega)}{H} \rfloor, & \text{fully loaded vehicles carrying } h = H \text{ passengers,} \\ 1, & \text{vehicle carrying the remainder } h = d_{ij}(\omega) \ mod \ H \text{ passengers ,} \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

With Equation (15), passenger demand $d_{ij}(\omega)$ can be converted to vehicle demand $\xi_{ij}(h), h = 1, 2, ..., H$ for any realization ω. In a nodal tree structure, these vehicle demands are represented by paths rooted at node $i \in \mathcal{N}$ pointing to leaf nodes $j_h \in \mathcal{N}, h = 1, 2, ..., H$. The number of passengers carried determines the ranking of these paths. Corollary 1 states that the ranking of these paths is from i -> j_H down to i -> j_1. The symbol "x -> y" represents the path from root node x to leaf node y.

Corollary 1. *Assume a nodal tree structure rooted at node $i \in \mathcal{N}$ with H leaf nodes $j_h \in \mathcal{N}, h = 1, 2, ..., H$. Path i -> j_h represents carrying $h = 1, 2, ..., H$ passengers. The ranking of these paths according to path revenue in descending order starts from the path i -> j_H down to the one i -> j_1.*

Proof. Since the path revenue is calculated by $r_{ij}(h) = f_{ij} \cdot h - c_{ij}$, if $h' > h$, we must have $r_{ij}(h') > r_{ij}(h)$. Because $H > H-1 > ... > 1$, the ranking of paths according to path revenue in descending order is from path i -> j_H down to the one i -> j_1. □

Proposition 1 and Corollary 1 provide a method to convert the passenger demands to vehicle demands where multiple paths represent the vehicle demands. The single-root-node recourse problem rooted in node $i \in \mathcal{N}$ pointing to node $j \in \mathcal{N}$ with random passenger demands d_{ij} can be represented by a nodal tree structure with root node i

and leaf nodes $j_h, h = 1, 2, ..., H$. The path revenue is $r_{ij}(h)$ and the corresponding path capacity is $\xi_{ij}(h)$ where $h = 1, 2, ..., H$. The idea of this conversion is illustrated in Figure 7. For a better understanding of this converting process, Table 3 shows an example with the discrete random variable d_{ij} taking four possible values and the capacity of vehicle $H = 4$. The ranking of these four paths is $P_1 > P_2 > P_3 > P_4$, corresponding to vehicles carrying $h = 4, 3, 2, 1$ passengers, respectively.

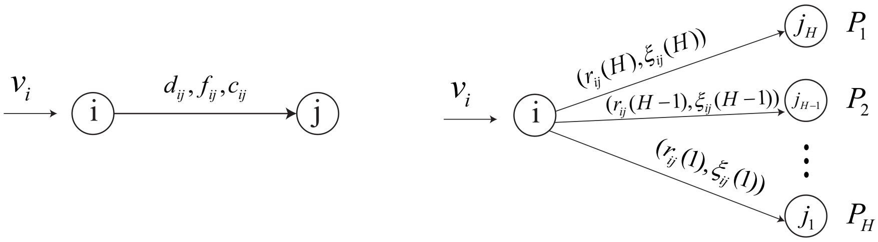

v_i
i
d_{ij}, f_{ij}, c_{ij}
j
v_i
i
$(r_{ij}(H), \xi_{ij}(H))$
j_H
P_1
$(r_{ij}(H-1), \xi_{ij}(H-1))$
j_{H-1}
P_2
$(r_{ij}(1), \xi_{ij}(1))$
j_1
P_H

Figure 7. Convert a single leaf node recourse problem to a nodal tree structure with H paths.

Table 3. An example of the converting process.

Possible Value of d_{ij}	**Corresponding Probability**	**Path Capacities $\xi_{ij}(h)$**			
		P_1	P_2	P_3	P_4
		$\xi_{ij}(4)$	$\xi_{ij}(3)$	$\xi_{ij}(2)$	$\xi_{ij}(1)$
3	0.1	0	1	0	0
5	0.3	1	0	0	1
6	0.4	1	0	1	0
9	0.2	2	0	0	1

The example in Table 3 shows that the numbers of vehicles $\xi_{ij}(h)$ dispatched from node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$ carrying $h = \{1, 2, 3, 4\}$ are also discrete random variables. Let us take $\xi_{ij}(4)$ and $\xi_{ij}(3)$, for example: the result indicates that the distribution of $\xi_{ij}(4)$ is $\{(0, 0.1), (1, 0.7), (2, 0.2)\}$ and $\xi_{ij}(3)$ is $\{(0, 0.9), (1, 0.1)\}$. However, we cannot obtain the total capacity of the first two paths, P_1 and P_2, by summing $\xi_{ij}(4)$ and $\xi_{ij}(3)$ directly because they are not independent. Proposition 2 provides an efficient approach to compute the total capacity of the first n paths.

Proposition 2. *For a sorted nodal tree rooted at the node $i \in \mathcal{N}$ pointing to H leaf nodes $j_h, h = 1, 2, ..., H$, assume the random variable d_{ij} takes $T < \infty$ possible values. Given all paths' capacities $\xi_{ij}(h)$, where $h = 1, 2, ..., H$, if the first n paths include those paths i -> j_H, i -> j_{H-1},..., i -> j_h, then the distribution of the total capacity of the first n paths Z^n_{ij} can be computed by $P\{Z^n_{ij} = k\} = \sum_T P\{\sum_{x=h}^{H} \xi_{ij}(x) = k\}$.*

Proof. Assume all paths rooted at point $i \in \mathcal{N}$ pointing to leaf nodes $j_h, h = 1, 2, ..., H$ have been ranked in descending order. According to Corollary 1, the path to point j_H must be ranked first and followed by paths to node $j_{H-1}, j_{H-2}, ..., j_1$. According to Proposition 1, each outcome of d_{ij} can be converted to $\xi ij(h), h = 1, 2, ..., H$. the value of the first n paths Z^n is $\sum_{x=h}^{H} \xi_{ij}(x)$ because the first n paths contain those paths i -> j_H, i -> j_{H-1},..., i -> j_h. Since $\xi_{ij}(x), x = h, h+1, ..., H$ are discrete variables taking finite possible values, the probability of $P\{Z^n = k\}$ can be computed by summing up those corresponding probabilities over all outcomes T on $\sum_{x=h}^{H} \xi_{ij}(x) = k$, which is $P\{Z^n_{ij} = k\} = \sum_T P\{\sum_{x=h}^{H} \xi_{ij}(x) = k\}$. □

Proposition 2 provides an efficient approach to obtain the total capacity of the first n paths, Z^n_{ij}. Let us demonstrate the calculation using Z^2_{ij}. Since the possible values of Z^2_{ij} are taken from $\xi_{ij}(4) + \xi_{ij}(3)$, it means Z^2_{ij} can take the values of 1 or 2, and the probability of Z^2_{ij} taking the value of 1 is the sum of all corresponding probabilities where $\xi_{ij}(4) + \xi_{ij}(3) = 1$, which are 0.1, 0.3, and 0.4. The calculation is the same for Z_{ij}^2, taking the value of 2, which is 0.2. Table 4 shows the results of Z_{ij}^n in the example shown in Table 3.

Table 4. Results of the total capacity of the first n paths.

n	Paths Included	Distribution of Z_{ij}^n
1	P_1	$\{(0, 0.1), (1, 0.7), (2, 0.2)\}$
2	P_1, P_2	$\{(1, 0.8), (2, 0.2)\}$
3	P_1, P_2, P_3	$\{(1, 0.4), (2, 0.6)\}$
4	P_1, P_2, P_3, P_4	$\{(1, 0.1), (2, 0.7), (3, 0.2)\}$

With Proposition 2, we can obtain the first n paths' total capacity Z_{ij}^n efficiently for a nodal tree rooted at node $i \in \mathcal{N}$ with leaf nodes $j_h \in \mathcal{N}, h = 1, 2, ..., H$, even though these paths' random capacities are not independent. It extends the one introduced by Powell and Cheung [38], where their algorithm is restricted by the condition that all random arc capacities are mutually independent. However, the approach in Proposition 2 can only apply to computing the total capacities of the first n paths for a single leaf node. Since a single-root-node recourse problem generally contains multiple leaf nodes, we state an approach in Proposition 3 to compute a nodal tree rooted at node $i \in \mathcal{N}$ with multiple leaf nodes $j \in \mathcal{N}$. To ease our presentation, we present this recourse problem with two leaf nodes $j \in \mathcal{N}$ and $k \in \mathcal{N}$, which relates to passenger demands d_{ij} and d_{ik}. It is worth noting that this is not a limitation of our approach.

Proposition 3. *Assume a nodal tree T_i rooted at node $i \in \mathcal{N}$ contains two leaf nodes $j \in \mathcal{N}$ and $k \in \mathcal{N}$. Duplicating leaf nodes j and k $H-1$ times, respectively, creates two new nodal subtrees T_{ij} and T_{ij} rooted at node i with $2H$ paths and leaf nodes $j_h, h = 1, 2, ..., H$ and $k_h, h = 1, 2, ..., H$, respectively. Assume all paths in T_i have been sorted in descending order by their revenue $r_{ij}(h)$ and $r_{ik}(h)$, where $h = 1, 2, ..., H$, and the first n paths contain paths i -> j_H,i -> j_{H-1},...,i -> j_{h1} and i -> k_H,i -> k_{H-1},...,i -> k_{h2}, such that the total capacities of the first n paths Z_i^n can be obtained by $Z_i^n = Z_{ij}^{h1} + Z_{ik}^{h2}$.*

Proof. For the nodal subtree T_{ij} rooted in node $i \in \mathcal{N}$ destined to leaf nodes $j_h, h = 1, 2, ..., H$ where all paths have been ranked by their revenue in descending order, if the first n paths contain i -> j_H,i -> j_{H-1},...,i -> j_{h1}, the total capacity of these paths is Z_{ij}^{h1}. It is the same for the nodal subtree T_{ik} rooted at node $i \in \mathcal{N}$ destined to leaf nodes $k_h, h = 1, 2, ..., H$. if the first n paths contain i -> k_H,i -> k_{H-1},...,i -> k_{h2}, then the total capacity of these paths is Z_{ik}^{h2}. Since passengers' demands d_{ij} and d_{ik} are independent, it means Z_{ij}^{h1} and Z_{ik}^{h2} are also independent. As a result, $Z_i^n = Z_{ij}^{h1} + Z_{ik}^{h2}$. □

A general single-root-node, multiple-leaf-node network recourse problem can be converted to a general nodal tree structure, and Proposition 3 provides an efficient way to compute the total capacity of the first n paths Z_i^n for a general nodal tree recourse problem. To ensure feasibility, a common technique is adding a virtual uncapacitated path with zero revenue into the nodal tree, representing that the service provider can hold as many vehicles at transit points $i \in \mathcal{N}$ but generate no revenue. Since this virtual path is uncapacitated, we need to calculate the total capacities of all sorted paths from the first path until reaching the virtual one. Furthermore, in our general single-root-node recourse problem, since the fleet size is a decision variable, the terminate criteria in computing $G_i(k)$ can be set to $G_i(k) \leq 0$. The rationale behind this setting is that the $v_i > k$ vehicles at transit point $i \in \mathcal{N}$ can only generate zero revenue.

A complete but simple example to demonstrate the whole computation process is presented in the following. Suppose a single-root-node recourse problem with a root node i and two leaf nodes j and k. Passenger demands d_{ij} and d_{ij} are discrete random variables that take on two possible values each, which are $\{(5, 0.4), (7, 0.6)\}$ and $\{(7, 0.3), (8, 0.7)\}$, respectively. Fares f_{ij} and f_{ik} are 4 and 5. Vehicle traversing costs c_{ij} and c_{ik} are 7 and 8. Assume the vehicle capacity $H = 4$. We first convert the passenger demands to vehicle demands, shown in Tables 5 and 6.

Next, we compute the path revenue $r_{ij}(h)$, $r_{ik}(h)$, and the first n paths' total capacities Z^n_{ij}, Z^n_{ik}, where $h = 1, 2, 3, 4$. Results are shown in Tables 7 and 8.

Table 5. Convert passenger demand d_{ij} to vehicle demand ξ_{ij}.

Possible Value of d_{ij}	Corresponding Probability	Path Capacities $\xi_{ij}(h)$			
		P^j_1	P^j_2	P^j_3	P^j_4
		$\xi_{ij}(4)$	$\xi_{ij}(3)$	$\xi_{ij}(2)$	$\xi_{ij}(1)$
5	0.4	1	0	0	1
7	0.6	1	1	0	0

Table 6. Convert passenger demand d_{ik} to vehicle demand ξ_{ik}.

Possible Value of d_{ij}	Corresponding Probability	Path Capacities $\xi_{ik}(h)$			
		P^k_1	P^k_2	P^k_3	P^k_4
		$\xi_{ik}(4)$	$\xi_{ik}(3)$	$\xi_{ik}(2)$	$\xi_{ik}(1)$
7	0.3	1	1	0	0
8	0.7	2	0	0	0

Table 7. Path revenue $r_{ij}(h)$ and the first n paths' total capacity Z^n_{ij} for nodal subtree T_{ij}.

n	Path Revenue	Distribution of Z^n_{ij}
1	9	$\{(1, 1.0)\}$
2	5	$\{(1, 0.4), (2, 0.6)\}$
3	1	$\{(1, 0.4), (2, 0.6)\}$
4	−3	$\{(2, 1.0)\}$

Table 8. Path revenue $r_{ik}(h)$ and the first n paths' total capacity Z^n_{ik} for nodal subtree T_{ik}.

n	Path Revenue	Distribution of Z^n_{ik}
1	12	$\{(1, 0.3), (2, 0.7)\}$
2	7	$\{(2, 1.0)\}$
3	2	$\{(2, 1.0)\}$
4	−3	$\{(2, 1.0)\}$

Finally, we add a virtual path to this nodal tree, where the path capacity is ∞ and the path revenue is 0. We sort all paths according to their revenue, and the resulting nodal tree is depicted in Figure 8.

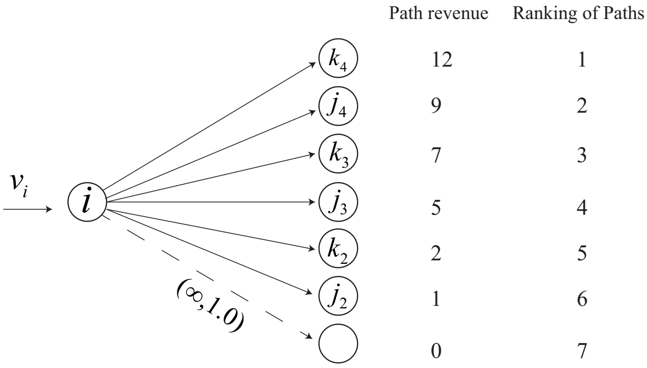


Figure 8. The resulting nodal tree with a virtual path.

With the sorted nodal tree, we can calculate the first n paths' total capacity Z^n according to Proposition 3. The results are shown in Table 9.

Table 9. Path revenue $r_i(n)$ and the total capacity of the first n paths for nodal tree T_i.

n	Path Revenue r_n	Computation of Z^n	Distribution of Z^n
1	12	$= Z^1_{ik}$	$\{(1, 0.3), (2, 0.7)\}$
2	9	$= Z^1_{ik} + Z^1_{ij}$	$\{(2, 0.3), (3, 0.7)\}$
3	7	$= Z^1_{ij} + Z^2_{ik}$	$\{(3, 1.0)\}$
4	5	$= Z^2_{ik} + Z^2_{ij}$	$\{(3, 0.4), (4, 0.6)\}$
5	2	$= Z^3_{ik} + Z^2_{ij}$	$\{(3, 0.4), (4, 0.6)\}$
6	1	$= Z^3_{ik} + Z^3_{ij}$	$\{(3, 0.4), (4, 0.6)\}$
7	0	Terminate	$\{(\infty, 1.0)\}$

Based on the total capacity of the first n paths, we can compute the routing probability $\phi(k, n)$ by equation $\phi(k, n) = P\{(Z^n \geq k\} - P\{Z^{n-1} \geq k\}$. $Z^0 = 0$ since it means no path in the nodal tree. The results of $\phi(k, n)$ are shown in Table 10, and we terminate these computations when $G(k) \leq 0$.

Table 10. Results of $\phi_i(k, n)$ and $G_i(k)$ for nodal tree T_i.

	$\boldsymbol{\phi_i(k,n)}$							**Value of**
k	**n = 1**	**2**	**3**	**4**	**5**	**6**	**7**	$\boldsymbol{G_i(k) = r_n \cdot \phi_i(k,n)}$
1	1.0	-	-	-	-	-	-	12.0
2	0.7	0.3	-	-	-	-	-	11.1
3	-	0.7	0.3	-	-	-	-	8.4
4	-	-	-	0.6	-	-	0.4	3.0
5	-	-	-	-	-	-	1.0	0.0

The pseudo-code for obtaining $\hat{Q}(v_i)$, which we call the Nodal Tree Recourse with Dependent Arc Capacities (NTRDAC) Algorithm 1, is outlined as follows.

Algorithm 1: NTRDAC algorithm for a nodal tree T_i rooted at node $i \in \mathcal{N}$.

```
Step 1: Set $n = 0$; $h_N = 0$; $Z^n = (0, 1.0)$ ;
Step 2: Convert $d_{ij}$ into $\xi_{ij}(h), j \in \mathcal{N}, h = 1, 2, ..., H$ by Equation (15);
Step 3: Compute $Z^h_{ij}, j \in \mathcal{N}, h = 1, 2, ..., H$ by Proposition 2;
Step 4: Set $n = n + 1$;
Step 5:     Set $h_j$ = index of the last path in nodal subtree $T_{ij}$ included
                  by the first $n$ paths of nodal tree $T_n$, where $j \in \mathcal{N}$.
Step 6:     Compute $Z^n$ by Proposition 3 where $Z^n = \sum_{x \in h_j, j \in \mathcal{N}} Z^x_{ij}$;
Step 7: Repeat step 4 to step 6 until reaching the uncapacitated path;
Step 8: Compute $\phi(s, n)$ using Equation (12);
Step 9: Compute $G_i(k)$ by Equation (13) until $G_i(k) \leq 0$.
```

4.4. First-Stage Problem

The NTRDAC algorithm enables us to obtain the exact value of $E_\omega Q_i(v_i, \boldsymbol{D}(\omega))$ efficiently by a function $\hat{Q}(v_i)$, where v_i is the scalar supplies (vehicles) to the transit point $i \in \mathcal{N}$. Equation (14) states that $\hat{Q}(v_i) = \sum_{k=1}^{v_i} G(k)$, such that the first-stage problem can be turned into a deterministic mixed integer programming (MIP) optimization problem by replacing $E_\omega Q_i(v_i, \boldsymbol{D}(\omega))$ with $\hat{Q}(v_i)$.

Let us define y^k_i = the number of vehicles dispatched at transit point $i \in \mathcal{N}$ taking path k (decision variables).

The first-stage problem can be reformulated as follows:

$$max \quad -g \cdot s + \sum_{i=1}^{N} \sum_{k=1}^{v_i} y_i^k \cdot G_i(k), \tag{16}$$

subject to

$$\sum_{i=1}^{N} v_i = s, \tag{17}$$

$$0 \leq y_i^k \leq 1, \quad \forall i \in \mathcal{N}, k = 1, 2, ..., v_i, \tag{18}$$

$$s \geq 0, s \in \mathbb{Z}. \tag{19}$$

The first-stage problem (16)–(19) is a common deterministic MIP problem that can be solved efficiently by commercial solvers, such as Cplex 12.6 or Gurobi 10.0.

5. Numerical Experiments

In this section, we conduct numerical experiments to evaluate the efficiency of the NTRDACA algorithm by reporting its CPU time for several test problems of different sizes. Afterward, to assess the value of the stochastic solution (VSS), we compare the results obtained by the decomposition method with those achieved by solving expected value models that replace all random variables with their expected values. All experiments are conducted on a DELL personal computer with an Intel Core i3-6100 CPU and 12 GB RAM. The program is coded in Java 8, and the heap space limitation is set to 700 MB.

5.1. The Efficiency of NTRDAC Algorithm

We evaluated the speed of the NTRDAC algorithm for six different sizes of problem. Three parameters determine the scale of these problems: (i) total number of transit points M, (ii) number of possible values that random arc capacities take on K, and (iii) total number of available seats per vehicle H. The total number of paths, calculated by $M \cdot (M-1)$, is the critical factor affecting the computation time because the number of scenarios is calculated by $K^{M \cdot (M-1)}$, which will increase exponentially as M increases. For the random capacity of each arc, we truncate a Poisson variable ξ at k if k is the largest integer such that $P\{\xi \geq k\} < 0.0001$. The Poisson variables' means range from 3 to 20, with each arc taking a random value. We simulate each test case 10 times and report the average CPU time. Table 11 summarizes the CPU times and problem settings.

Table 11. Efficiency of NTRDAC algorithm for recourse problems.

Problems				CPU Time (s)	
Test Case	**No. of Nodes**	**No. of Possible Values**	**Total Scenarios**	**NTRDAC Algorithm**	**Gurobi (DEP)**
1	3	3	729	0.001	13.62
2	4	2	4096	0.001	345
3	5	2	2^{20}	0.004	–
4	10	5	5^{190}	0.018	–
5	15	10	10^{210}	0.054	–
6	30	15	15^{870}	0.915	–

The symbol "–" indicates an error due to a Java heap size overflow.

Table 11 shows that our proposed NTRDAC algorithm performs quite well, especially for real-world large-scale problems. First, it can obtain the exact value of the recourse problems as the numeration process does. Second, it can achieve the results quickly and be stable even though the scenarios increase exponentially. Finally, in the last test case, with 30 nodes and 15 possible values of each arc capacity, resulting in 15^{870} scenarios, our NTRDAC algorithm can still achieve the expected value in less than one second. In real-world applications, a shuttle service transportation network with 30 transit points typically covers most areas within a city. The ability to efficiently solve such a large-scale problem demonstrates that the NTRDAC algorithm is suitable for practical use. In contrast, when using a commercial solver like Gurobi to tackle the same problem, solutions can only be obtained for much smaller instances—those with fewer than five nodes and each random variable taking three possible values. This highlights the curse of dimensionality, where the problem size grows exponentially with the square of the number of nodes in this model.

5.2. The Value of Stochastic Solutions (VSSs)

In this subsection, we aim to evaluate the value of the stochastic solution (VSS). All arc capacities are generated with truncated Poisson random variables as we did in Section 5.1. We vary the number of nodes and cost matrix to generate different test cases. For each test case, we first use the NTRDAC algorithm to obtain the exact value of the recourse functions for the second-stage problem, then we solve the first-stage problem and obtain the solution S^1_{Decom} for the first stage. After that, we replace the random arc capacities with their expected values in the second stage, then we solve the deterministic equivalent problem and obtain the expected value solution S^1_{EV} for the first stage. Taking these two different solutions for the first-stage problem, which is the fleet size S^1_{Decom} and S^1_{EV}, we apply the Monte Carlo simulation method for the second stage. That is, we randomly pick one of the values of arc capacities for all arcs in the second stage, which creates a scenario. For each scenario, we solve this deterministic network flow problem and obtain the expected total revenue R^2_{Decom} and R^2_{EV} corresponding to having S^1_{Decom} and S^1_{EV} vehicles, respectively. Comparing $-g \cdot S^1_{Decom} + R^2_{Decom}$ with $-g \cdot S^1_{EV} + R^2_{EV}$, we achieve the value of the stochastic solution. We take 1000 scenarios in each simulation. Table 12 shows the test cases and the value of the stochastic solution, where the positive values of VSS indicate that the stochastic solutions are better than the expected value solutions. The reason is stochastic solutions account for the uncertainty in the data and decision-making process, leading to more robust and flexible outcomes.

Table 12. Results of the value of stochastic solution.

Problems		EV Solution		Stochastic Solution		VSS
Test Case	No. of Nodes	Fleet Size	Total Revenue	Fleet Size	Total Revenue	
1	5	16	455.37	20	495.32	39.95
2	10	88	2791.89	97	2885.35	93.46
3	30	887	29,989.94	912	30,230.29	240.35

5.3. Managerial Application

In this section, we focus on the managerial application of the NTRDAC algorithm, which is crucial for practitioners. Specifically, we examine the total expected profit by varying the vehicle capacity H, while keeping the random passenger demands consistent across all test cases. We use synthetic data to simulate the decision-making process for selecting vehicle types and fleet size prior to launching the transportation service. The random passenger demands are generated in the same manner as described in Section 5.1. The results are presented in Table 13.

Table 13. A comparison of expected profits generated by different types of vehicles for 30 transit points.

Vehicle Capacity (H)	Fleet Size (s)	Expected Total Profit
4	912	30,388.63
6	816	59,698.29
11	628	85,149.58
17	509	86,647.13
33	430	79,292.92
50	313	61,991.54

According to the results in Table 13, investments in vehicles with a capacity of 17 yields the highest profit, with a capacity of 11 being the second-best option. This may be due to the fact that the random passenger demands follow a Poisson distribution with means that are close to these vehicle capacities. In practice, the distribution of estimated passenger demands may vary and be more general. The NTRDAC algorithm can easily adapt to any general distribution by approximating them with discrete random variables, taking K possible values. As K increases, the accuracy of the approximation improves.

6. Conclusions

This paper addresses a fleet size optimization problem with uncertain passenger demands over point-to-point shared demand responsive service. We propose a stochastic programming framework to model this problem. The key idea is to obtain the exact value of the expected recourse functions in the second stage. We achieve this through a tailored Nodal Tree Recourse with Dependent Arc Capacities (NTRDAC) algorithm. The NTRDAC algorithm extends the TREC algorithm but relaxes the restrictions that all random arc capacities must be independent. Before applying the efficient NTRDAC algorithm, we introduce a conversion process that converts the random passenger demands to random vehicle demands represented by random arc capacities. This conversion is a critical process in generating the nodal tree. The NTRDAC algorithm simplifies the first-stage problem to a deterministic integer programming problem, which can be solved efficiently. Numerical experiments demonstrate that the NTRDAC algorithm can efficiently obtain the exact value of the recourse functions, indicating that our algorithm is quite suitable for large-scale practical problems. For practitioners, implementing the NTRDAC algorithm for real-world fleet size optimization problems is straightforward. By using random demand estimates from forecasts or historical data, the optimal results can be computed by selecting different vehicle capacities and their corresponding parameters, such as depreciation and trip costs.

Considering homogeneous resources is one limitation of this research, and extending the model to accommodate more complex network structures for realistic problems, such as incorporating multiple types of vehicles and allowing backlogged demands, is a promising direction. A potential research direction could explore how to decompose heterogeneous resources and efficiently manage unfulfilled demands from previous stages.

Author Contributions: Conceptualization, F.X.; methodology, F.X. and H.D.; programming and testing, F.X. and C.W.; writing—original draft preparation, F.X. and C.W.; writing—review and editing, C.W. and H.D.; supervision, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Author Ce Wang was employed by China Southern Gird. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Ho, S.C.; Szeto, W.; Kuo, Y.H.; Leung, J.M.; Petering, M.; Tou, T.W. A survey of dial-a-ride problems: Literature review and recent developments. *Transp. Res. Part B Methodol.* **2018**, *111*, 395–421. [CrossRef]
2. Rist, Y.; Forbes, M.A. A new formulation for the dial-a-ride problem. *Transp. Sci.* **2021**, *55*, 1113–1135. [CrossRef]
3. Miah, M.M.; Naz, F.; Hyun, K.K.; Mattingly, S.P.; Cronley, C.; Fields, N. Barriers and opportunities for paratransit users to adopt on-demand micro transit. *Res. Transp. Econ.* **2020**, *84*, 101001. [CrossRef]
4. Macfarlane, G.S.; Hunter, C.; Martinez, A.; Smith, E. Rider Perceptions of an On-Demand Microtransit Service in Salt Lake County, Utah. *Smart Cities* **2021**, *4*, 717–727. [CrossRef]
5. Ziakopoulos, A.; Oikonomou, M.G.; Vlahogianni, E.I.; Yannis, G. Quantifying the implementation impacts of a point to point automated urban shuttle service in a large-scale network. *Transp. Policy* **2021**, *114*, 233–244. [CrossRef]

6. Oikonomou, M.G.; Orfanou, F.P.; Vlahogianni, E.I.; Yannis, G. Impacts of Autonomous Shuttle Services on Traffic, Safety and Environment for Future Mobility Scenarios. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020. [CrossRef]
7. Jiang, S.; Guan, W.; Yang, L.; Zhang, W. Feeder Bus Accessibility Modeling and Evaluation. *Sustainability* **2020**, *12*, 8942. [CrossRef]
8. Gao, X.; Liu, S.; Jiang, S.; Yu, D.; Peng, Y.; Ma, X.; Lin, W. Optimizing the Three-Dimensional Multi-Objective of Feeder Bus Routes Considering the Timetable. *Mathematics* **2024**, *12*, 930. [CrossRef]
9. Si, H.; Shi, J.; Hua, W.; Cheng, L.; De Vos, J.; Li, W. What influences people to choose ridesharing? An overview of the literature. *Transp. Rev.* **2023**, *43*, 1211–1236. [CrossRef]
10. Aguiléra, A.; Pigalle, E. The Future and Sustainability of Carpooling Practices. An Identification of Research Challenges. *Sustainability* **2021**, *13*, 11824. [CrossRef]
11. Tirachini, A. Ride-hailing, travel behaviour and sustainable mobility: An international review. *Transportation* **2020**, *47*, 2011–2047. [CrossRef]
12. Chalermpong, S.; Kato, H.; Thaithatkul, P.; Ratanawaraha, A.; Fillone, A.; Hoang-Tung, N.; Jittrapirom, P. Ride-hailing applications in Southeast Asia: A literature review. *Int. J. Sustain. Transp.* **2023**, *17*, 298–318. [CrossRef]
13. Martí, P.; Jordán, J.; González Arrieta, M.A.; Julian, V. A Survey on Demand-Responsive Transportation for Rural and Interurban Mobility. *Int. J. Interact. Multimed. Artif. Intell.* **2023**, *8*, 43–54. [CrossRef]
14. Wang, H.; Li, J.; Wang, P.; Teng, J.; Loo, B.P. Adaptability analysis methods of demand responsive transit: A review and future directions. *Transp. Rev.* **2023**, *43*, 676–697. [CrossRef]
15. Czioska, P.; Kutadinata, R.; Trifunović, A.; Winter, S.; Sester, M.; Friedrich, B. Real-world meeting points for shared demand-responsive transportation systems. *Public Transp.* **2019**, *11*, 341–377. [CrossRef]
16. Kjoerstad, K.; Renolen, H. Better Public Transport: Passengers' Valuation of Time and Service Improvements. In Proceedings of the Public Transport Planning and Operations. Proceedings of Seminar F Held at the PTRC European Transport Forum, London, UK, 2–6 September 1996; Volume P405.
17. Mylonas, C.; Stavara, M.; Tzanis, D.; Mitsakis, E. Fleet optimization in shared mobility services: Theoretical findings and future steps. *Transp. Res. Procedia* **2023**, *72*, 2872–2879. [CrossRef]
18. Daganzo, C.F.; Ouyang, Y. A general model of demand-responsive transportation services: From taxi to ridesharing to dial-a-ride. *Transp. Res. Part B Methodol.* **2019**, *126*, 213–224. [CrossRef]
19. Hörl, S.; Ruch, C.; Becker, F.; Frazzoli, E.; Axhausen, K. Fleet operational policies for automated mobility: A simulation assessment for Zurich. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 20–31. [CrossRef]
20. Shehadeh, K.S.; Wang, H.; Zhang, P. Fleet sizing and allocation for on-demand last-mile transportation systems. *Transp. Res. Part C Emerg. Technol.* **2021**, *132*, 103387. [CrossRef]
21. Beaujon, G.J.; Turnquist, M.A. A model for fleet sizing and vehicle allocation. *Transp. Sci.* **1991**, *25*, 19–45. [CrossRef]
22. Winter, K.; Cats, O.; Correia, G.H.d.A.; Van Arem, B. Designing an automated demand-responsive transport system: Fleet size and performance analysis for a campus–train station service. *Transp. Res. Rec.* **2016**, *2542*, 75–83. [CrossRef]
23. Diana, M.; Dessouky, M.M.; Xia, N. A model for the fleet sizing of demand responsive transportation services with time windows. *Transp. Res. Part B Methodol.* **2006**, *40*, 651–666. [CrossRef]
24. Militão, A.M.; Tirachini, A. Optimal fleet size for a shared demand-responsive transport system with human-driven vs automated vehicles: A total cost minimization approach. *Transp. Res. Part A Policy Pract.* **2021**, *151*, 52–80. [CrossRef]
25. Powell, W.B. A unified framework for stochastic optimization. *Eur. J. Oper. Res.* **2019**, *275*, 795–821. [CrossRef]
26. Powell, W.B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*; John Wiley & Sons: New York, NY, USA, 2007; Volume 703.
27. Laporte, G.; Louveaux, F.V. The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* **1993**, *13*, 133–142. [CrossRef]
28. Sherali, H.D.; Fraticelli, B.M. A modification of benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *J. Glob. Optim.* **2002**, *22*, 319–342. [CrossRef]
29. Rockafellar, R.; Wets, R.J.B. Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Math. Oper. Res.* **1991**, *16*, 119–147. [CrossRef]
30. Mulvey, J.M.; Ruszczyński, A. A New Scenario Decomposition Method for Large-Scale Stochastic Optimization. *Oper. Res.* **1995**, *43*, 477–490. [CrossRef]
31. Higle, J.L.; Sen, S. Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse. *Math. Oper. Res.* **1991**, *16*, 650–669. [CrossRef]
32. Girardeau, P.; Leclere, V.; Philpott, A.B. On the Convergence of Decomposition Methods for Multistage Stochastic Convex Programs. *Math. Oper. Res.* **2015**, *40*, 130–145. [CrossRef]
33. Birge, J.R.; Louveaux, F. *Introduction to Stochastic Programming*; Springer Science & Business Media: New York, NY, USA, 2011.
34. Dupačová, J.; Gröwe-Kuska, N.; Römisch, W. Scenario reduction in stochastic programming. *Math. Program.* **2003**, *95*, 493–511. [CrossRef]
35. Zhang, W.; Wang, K.; Jacquillat, A.; Wang, S. Optimized scenario reduction: Solving large-scale stochastic programs with quality guarantees. *INFORMS J. Comput.* **2023**, *35*, 886–908. [CrossRef]

36. Kammammettu, S.; Li, Z. Scenario reduction and scenario tree generation for stochastic programming using Sinkhorn distance. *Comput. Chem. Eng.* **2023**, *170*, 108122. [CrossRef]
37. Wallace, S.W. A Piecewise linear upper bound on the network recourse function. *Math. Program.* **1987**, *38*, 133–146. [CrossRef]
38. Powell, W.B.; Cheung, R.K. Stochastic programs over trees with random arc capacities. *Networks* **1994**, *24*, 161–175. [CrossRef]
39. Powell, W.B.; Cheung, R.K. A network recourse decomposition method for dynamic networks with random arc capacities. *Networks* **1994**, *24*, 369–384. [CrossRef]
40. Schuller, P.; Fielbaum, A.; Alonso-Mora, J. Towards a geographically even level of service in on-demand ridepooling. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 2429–2434. [CrossRef]
41. Lu, C.; Tiwari, S.; Nassir, N.; Nagel, K. Efficient operation of demand-responsive transport (DRT) systems:Active requests rejection. *Procedia Comput. Sci.* **2024**, *238*, 81–90. [CrossRef]
42. Cheung, R.K.; Powell, W.B. An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management. *Oper. Res.* **1996**, *44*, 951–963. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.