Acknowledgements  We would like to thank Puerto Rico’s Metropolitan Bus Authority and Mr. Mark Mindorff at Disabled and Aged Regional Transit Services (DARTS) in Hamilton, Ontario, Canada for sharing their insights and discussions with the research team.

Disclaimer  The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation’s University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.
Contents

1 Introduction and Problem Statement 3

2 Predicting Trip Cancellations and No-Shows in Paratransit Operations 4
  2.1 Data ......................................................................................... 5
  2.2 Prediction Problems and Methods .................................................. 7
    2.2.1 Applied Machine Learning Algorithms ................................. 8
    2.2.2 Clustering Approach to Class Probability Assignment .......... 9
  2.3 Accuracy Tests and Model Applications ....................................... 10
    2.3.1 Features, Feature Selection, and Feature Influence in the Likelihood of
          a Reservation Outcome .......................................................... 10
    2.3.2 Model Accuracy Tests .......................................................... 11
  2.4 Daily Demand Estimation Tests ................................................ 13
    2.4.1 Scenario Generation for Vehicle Scheduling ......................... 14
  2.5 Closing Remarks ....................................................................... 15

3 Paratransit Routing Models and Overbooking 17
  3.1 Monte Carlo Scenario-Based Feasibility Assessment ..................... 18
  3.2 Dynamic Dial-A-Ride Problem with Cancellations ....................... 19
    3.2.1 Dial-A-Ride Problem .......................................................... 20
    3.2.2 Dial-A-Ride Problem with Cancellations ............................. 23
    3.2.3 Illustrative Example .............................................................. 25
  3.3 Case study ............................................................................... 27
  3.4 Concluding Remarks .............................................................. 28
1 Introduction and Problem Statement

Publicly financed paratransit systems are a critically important social service for individuals with disabilities who do not have access or cannot afford alternative modes of transportation. Unfortunately, paratransit services are also one of the most expensive services provided by transit agencies. For example, in 2015 the average cost per passenger trip for paratransit systems operated by transit agencies was $24.90, while the cost per trip in standard fixed-route bus systems was $11.57 per trip (FTA, 2016). By federal law, the fare that a transit agency can charge for a paratransit trip cannot exceed twice the fare charged for a comparable trip on its fixed-route bus system, and as a consequence paratransit systems need to be heavily subsidized to cover operational costs. Naturally, transit agencies have attempted to implement various measures to improve the overall efficiency of their operations and reduce operational expenses. Among these measures are policies aimed at reducing no-shows and late cancellations (i.e., two situations in which a scheduled trip is not performed). No-shows and late cancellations are events that waste transit agencies’ resources, as well as degrade a system’s productivity by preventing other users from utilizing supplied service slots.

The impacts of no-shows and late cancellations are dramatically illustrated in the operations of Puerto Rico’s largest, publicly operated paratransit system: “Llame y Viaje” (Call and Travel; LV). A third of LV’s scheduled trips are either cancelled or no-shows. As a result, this system is among the least productive (0.76 trips per hour) and most expensive ($61 per trip) in the US (FTA, 2016). Transit agencies usually prevent the scale of LV’s problems by instituting and effectively enforcing policies that have been proven to reduce no-shows and late cancellations, including the suspension of services to individuals who have a pattern of missing scheduled trips. Despite these efforts, “[p]eriodic passenger no-shows and late cancellations are an expected cost of doing business for most paratransit systems” (Mathias, 2005).

Given that no-shows and late cancellations are part of the normal operations of paratransit systems, the analysis and prediction of their occurrence should be a formal component of operational and planning models for paratransit systems in order to improve productivity and decrease operating costs. However, little to no attention has been given to understanding and predicting no-shows or late cancellations for paratransit operations. In addition, the uncertainty introduced by no-shows and late cancellations has not been formally incorporated in the crucial operations processes of designing vehicle routes and booking trip requests in paratransit systems.

The objective of this research project is threefold: (i) to develop a classification methodology to predict no-shows and cancellations of trips in paratransit systems, (ii) to incorporate the classification model predictions into trip booking and routing models for paratransit operations (including novel overbooking problems in the context of paratransit), and (iii) to evaluate the value of the classification models’ predictions in paratransit planning and operations. As part of this project, the researchers will collaborate with Puerto Rico’s Metropolitan Bus Authority to develop and apply the data analytic techniques proposed in the project using data from the LV paratransit system previously discussed.
2 Predicting Trip Cancellations and No-Shows in Paratransit Operations

Trip cancellations and no-shows are inescapable realities in paratransit system operations, despite the efforts of transit agencies to reduce their occurrence. Both events can generate considerable social and operational costs. The trip cancellation and no-show problem present in paratransit is similarly encountered in other systems that use reservations to manage demand. For instance, the operations of airlines, hotels, and medical offices are affected by reservation cancellations and no-shows. In the airline and hotel industries in particular, it is standard practice to not only expect no-shows and cancellations, but to also compute and incorporate the probability of their occurrence into operational decisions. The overbooking models in airline revenue management systems are an example of a decision tool that takes into account the probability of cancellations and no-shows (in this case, on whether to accept or reject a reservation once the service capacity has been exceeded) (Chatwin, 1999; Barnhart, 2003 applications).

Naturally, methods to quantify the probability of reservation cancellations and no-shows are required before any practical implementation of decision tools capable of accounting for the stochasticity caused by these events. To this end, machine learning has been shown to be an effective approach to estimate reservation outcome probabilities. For example, Lawrence et al. (2003) utilized predictive models, such as decision trees and naïve Bayes algorithms, to generate individual-level no-show probabilities for airline flights. In the context of hotel bookings, Morales and Wang (2010) developed support vector machine algorithms to dynamically forecast cancellations in a given booking horizon. Regression-based models, like logistic regression, have also been used to predict no-show probabilities for medical appointments (Daggy et al., 2010; Harris et al., 2016; Huang and Hanauer, 2014). Machine learning models have even been fitted, in the form of neural networks, to predict reservation cancellations in restaurant chains (Huang et al., 2013).

In this section we discuss research conducted on the application of standard machine learning algorithms to predict trip cancellations and no-shows in the context of paratransit systems. We examined the accuracy of the machine learning algorithms in the prediction of paratransit trip reservation outcomes and study the utility of these algorithms for demand prediction. This analysis should be of interest to transit operators given the practical applications that trip cancellation and no-show predictions can have on paratransit operations, particularly in systems that require users to make reservations with a day or more in advance. First, estimates of the uncertainty regarding the outcome of trip reservations can be used to generate robust vehicle schedules that minimize the expected cost of servicing pick-up and drop-off requests (Ghilas et al., 2016). In addition, the output of predictive analytics could be used as inputs to overbooking models of paratransit booking management tools (akin to the overbooking models in the airline and hospitality industries). This application of machine learning could help refine the overbooking strategies already in place in large-scale paratransit operations (Mathias, 2005).

Data from the paratransit service operated by the Metropolitan Bus Authority (MBA) of Puerto Rico was used in this study. The MBA data is described in the next subsection. In Section 2.2 the prediction problems of interests are discussed, and an overview of the
machine learning algorithms used is presented. A clustering-based approach to assign outcome probabilities to reservations is also proposed in Section 2.2. The results of illustrative applications of the machine learning methods used are reported in Section 2.3. In Section 2.5 closing remarks are offered.

2.1 Data

MBA operates a program called “Llame y Viaje” (Dial-and-Ride) that functions as the complementary paratransit service to its fixed-route bus system. MBA provided paratransit trip reservation data for the years 2010-2017. This data set consists of 978,411 records of trip reservations. Each reservation record contains information on: the day the reservation was created; the trip date and purpose; the type of mobility aids the user required; the reservation type (e.g., a subscription reservation, meaning that it is a reservation by a user that is subscribed to a repeating trip schedule); and the user’s disability type, sex, and birthdate. The categorical data (e.g., reservation type, disability type, sex) were converted into a total of 72 binary variables (or features, in machine learning terminology). Also, the data set contains the outcome of each reservation, which, for the purposes of this study, were grouped into four classes: performed (reservation resulted in the intended trip), cancellation, late cancellation (cancelled in the 24-hour period before the trip), and no-show (vehicle arrived but the person was not present at the pick-up location). The models fitted in this study were used to make predictions regarding the outcome of trip reservations, either in terms of assigning a reservation into an outcome class or quantifying the reservations’ class membership probabilities. Figure 1 presents the percentage frequency distribution of the reservations’ class memberships for the available years. On average, 67.2% of reservations resulted in a trip (performed class), 23.5% were cancelled, 6.8% were cancelled late, and 2.5% resulted in no-shows.
The average system user was 66 years old, female (64% of users) and more likely to be affected by a neuromuscular disease (30% of users). On average, users made reservation 10 days in advance. As shown in Figure 2, 34% percent of reservations were made for medical purposes (if reservations for medical appointments and dialysis treatment are combined). The most common trip purpose was work, which on average represented 23% of all reservations. Figure 3 reports the class distribution of the reservation outcomes by trip purpose. Reservations for the purpose of receiving dialysis treatment were the more likely to result in a trip, which was the case for 81% of reservations. Coincidentally, 81% of the reservations by people with kidney disease resulted in trips (the highest trip rate of any disability type). In contrast, only 55% of reservations made by people with schizophrenia resulted in trips; schizophrenics had the lowest trip performance rate.

Missing from the data are the dates and time in which reservations were cancelled, as well as detailed information on trip origins and destinations. Had the data on the cancellation timings been available, models could have been developed that dynamically updated the cancellation probabilities of reservations as their trip dates approached. In addition, more detailed origin and destination data could allow the detection of patterns (e.g., associated with trip purposes beyond the generic purpose information available) that could have led to more accurate predictions. In any case, agencies such as MBA can easily collect these missing data. Although, even without these additional data, machine learning models can be applied to produce useful results, as this section will show.
2.2 Prediction Problems and Methods

Machine learning algorithms were used for two purposes: to quantify the probability of reservation outcomes and to classify reservations into outcome classes. A binary prediction problem was considered under the assumption that the transit operator is only interested in predicting whether a reservation will result in a trip or not. In addition, a multiclass prediction problem was studied with four possible reservation outcome classes: performed, cancellation, late cancellation, and no-show classes. Next, the machine learning methods
used in this study are briefly reviewed, and a clustering-based procedure is proposed. The machine learning methods discussed in this section were implemented using the scikit-learn Python library (Pedregosa et al., 2011).

2.2.1 Applied Machine Learning Algorithms

Logistic regression, naïve Bayes algorithms, decision tree learning, and artificial neural networks were used as solutions to the aforementioned prediction problems. Logistic regression is a widely used linear statistical model that computes the probability of an outcome using a logistic function (Peng and So, 2002). The model features are linearly combined in this model. Naïve Bayes algorithms apply Bayes’ theorem to predict the probability that a set of features (or feature vector) belongs to a particular class. This type of algorithm is called naïve because it assumes that, conditioned on the class, the features are probabilistically independent of each other (Bishop, 2006). In decision tree learning a tree-like structure (i.e., a decision tree) partitions the feature space using a set of decision rules that are learned, or inferred, from the data. The decision rules partition the data with the goal of creating groups of data samples that have the same classification; these rules can then be used to assign a class label (e.g., cancellation outcome) to any new data sample (e.g., a new reservation) (Myles et al., 2004). Lastly, artificial neural network models offer a black box approach to making predictions. In these models, a series of linear combinations of features are transformed via nonlinear functions (e.g., sigmoid functions), which are themselves linearly combined (Bishop, 2006). As their name suggests, neural networks were originally inspired by the information processing mechanisms in biological systems, although the algorithms used in practice have only tenuous connections to their biological counterparts.

In addition, ensemble methods were explored. An ensemble model is a model that combines the classification information (e.g., an outcome class assignment) provided by multiple prediction models. These methods are considered as they have been shown to often produce better results than individual prediction models (Dietterich, 2000). The ensemble methods implemented in this study were: random forest, majority voting, bagging, AdaBoost, and gradient boosting. To overcome some of the limitations of decision trees, random forest algorithms randomly generate multiple decision tree classifiers and then combine the classification information provided by each tree to output a single class prediction (i.e., random forests are an ensemble of decision trees) (Ho, 1995). In a majority voting classifier, a sample is assigned the class predicted by the majority of the individual models (e.g., a set of the five algorithms previously discussed), which are trained on the same data set. Bagging methods train predictive models on randomly generated subsets of data drawn from the original data set, and then majority voting is used based on the set of model predictions. An adaptive data set is also used in AdaBoost to train predictive models. In AdaBoost weak classifiers are iteratively trained on data that is reweighted based on the prediction errors of previous iterations; training samples that were incorrectly classified have their weight increased, while the weights of correctly classified samples are decreased (Scikit-Learn, a). Again, a sample’s class is predicted based on the class labeled identified by the majority of the AdaBoost generated models. Gradient boosting also iteratively trains new prediction models, but it does so by training with respect to the error of the model ensemble generated in the previous iteration (Natekin and Knoll, 2013).
The selected machine learning algorithms were chosen because of their widespread application, and the fact that they have been implemented in multiple programming languages, including open-source languages like Python and R. There are other types of machine learning algorithms that can be used to predict reservation outcomes, in addition to the multiple versions that the overviewed algorithms can assume. The algorithms considered, however, provide a sound basis for the purposes of this study, namely, the analysis of the prediction efficacy and the application of common machine learning algorithms in the context of paratransit demand management.

2.2.2 Clustering Approach to Class Probability Assignment

The previous models are examples of supervised learning algorithms, that is, the models are trained using data in which each feature vector is paired with its class label; the models’ structures are inferred with respect to an error (or loss) function that captures the differences between the model predictions and the observed classes. In this section a simple procedure is proposed to assign class probabilities using clustering algorithms, such as k-means clustering. K-means clustering is an example of an unsupervised learning algorithm, meaning that the algorithm does not use class membership information in its learning process. The objective of k-means clustering is to partition the available data sample into \( K \) clusters. The clusters are generated by finding the cluster assignments that minimize the sum of squared difference between the data samples (or points) and the centers of each cluster, where a cluster center is the sample mean vector of the data in the cluster (hence the name of the algorithm) (Bishop, 2006).

The clustering approach proposed assumes that reservations can be clustered based on their features, which can easily include information on the past behavior of the individual who made the reservation. For example, people with high rates of cancellations could be automatically clustered into a group by the algorithm. Then, the outcome class probabilities of each cluster would be computed as its historical class proportion. Any new reservation would be assigned to one of the cluster and assigned its class probabilities. The following algorithm provides a step-by-step description of the process:

\[\text{Step 1.}\] Select a data sample as the training data. Define the outcome classes of interest. Let \( c \ (c = \{1, 2, \ldots, C\}) \) denote the outcome class index.

\[\text{Step 2.}\] Apply the clustering algorithm to partition the data into \( K \) clusters. Let \( k \ (k = \{1, 2, \ldots, K\}) \) denote the cluster index.

\[\text{Step 3.}\] For each cluster \( k \) and for each class \( c \), determine the proportion \( (p_{kc}) \) of reservations that have outcome \( c \). The outcome vector \( p_k \) contains the class membership probabilities for any reservation assigned to the cluster (e.g., if \( p_k = [0.5, 0.2, 0.1, 0.2] \), a reservation that is in cluster \( k \) has a 0.5 probability of belonging to class 1, 0.2 probability of belonging to class 2, and so on).

\[\text{Step 4.}\] For each reservation for an upcoming trip, determine which cluster center is closest to the reservation’s feature vector (e.g. based on their Euclidian distance), and assign to the reservation the class probability vector \( p_k \).

In an on-line application of this algorithm, the clusters and their class proportion could
be updated as new reservation and outcome data are obtained. In addition to the k-means clustering, other clustering algorithms are suited to handle mixed data (data composed of numeric and categorical features), such as the k-prototypes algorithm (Huang, 1997) could be considered. In this study k-means and k-prototypes were used.

2.3 Accuracy Tests and Model Applications

The performance of the selected machine learning methods was examined using standard accuracy tests, as well as two illustrative applications of practical relevance to paratransit operations. The two applications were: the prediction of total daily trips and the generation of reservation outcome scenarios. Table 1 reports basic decisions made to specify the models used. Unless otherwise specified, the default parameters in the scikit-learn module were used. The next subsections describe the features used in the tests, as well as their results.

Table 1: Details on the Specifications of the Machine Learning Algorithms

<table>
<thead>
<tr>
<th>Methods</th>
<th>Specification Decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>The multinomial logistic regression was used for the multiclass problem.</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>The Gaussian Naïve Bayes algorithm was used for the binary problem, and the Multinomial Naïve Bayes algorithm was used for the multiclass problem.</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>The class weight parameter was set to “balanced”.</td>
</tr>
<tr>
<td>Artificial Neural Network</td>
<td>The Multi-Layer Perceptron algorithm was used with a maximum of 2,000 iterations.</td>
</tr>
<tr>
<td>Random Forest</td>
<td>The class weight parameter was set to “balanced”.</td>
</tr>
<tr>
<td>Voting</td>
<td>Logistic regression, decision tree, and random forest were used as voters.</td>
</tr>
<tr>
<td>Bagging</td>
<td>Logistic regression was set as the base estimator.</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>Decision tree was set as the base estimator.</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>The default parameters were used.</td>
</tr>
<tr>
<td>K-means</td>
<td>Five clusters selected based on the elbow method</td>
</tr>
<tr>
<td>K-prototypes</td>
<td>Five clusters selected based on the elbow method</td>
</tr>
</tbody>
</table>

2.3.1 Features, Feature Selection, and Feature Influence in the Likelihood of a Reservation Outcome

Categorical and numerical (real-valued and integer-valued) features were used in the tests. As previously discussed, 72 binary variables were generated based on the original dataset. The binary (or dummy) features fall into the following feature groups: trip purpose dummies (10 features), disability dummies (41 features), mobility aid dummies (11 features), day-of-travel dummies (6 features), male dummy (one feature), and reservation type dummies (three features). Apart from the user age, the numeric features were computed by analyzing the reservation booking information of each user. The numeric features used were: age; the rates of cancellation, late cancellation, and no-show that the user had in last 30 reservations; the
number of reservations made during one call; and the days in advance that the reservation was made.

In total, 78 features were created. Feature selection procedures were applied to select the feature set that would produce the most accurate models. Feature selection can also help improve the efficiency of the training process by removing extraneous data. As implemented in the scikit-learn Python module (Pedregosa et al., 2011; Scikit-Learn, b), univariate tests (including the false positive rate, the false discovery rate, and the family wise error rate tests using as score functions the $\chi^2$ test statistic and the analysis of variance F-value) and the recursive feature elimination method (with the logistic regression, decision tree, and random forest models) were used to identify different sets of influential features. Not surprisingly, different methods produced different sets of feature selections, but there were features that were consistently identified as important, chief among them being the numerical features and the dummies for the work, medical appointment, dialysis, and shopping trip purposes. Model accuracy tests were conducted with the different sets of feature selections, as well as with all the features. No improvement in model accuracy was observed from using the different feature sets, and the model training times were not an issue of practical concern. Therefore, in the accuracy and application tests reported in the following subsections, the models were trained using all available features.

In the interest of extracting some general insights into the effects of a set of features that were consistently identified as important by the feature selection methods, a parsimonious logistic regression model with a binary dependent variable (1 if trip was performed, 0 otherwise) is presented; the model features and their coefficients are reported in Table 2. In addition, this type of estimation exercise illustrates how logistic regression models could be useful to test model specification ideas, as logistic regression models are relatively easy to interpret (as opposed to black-box models like neural networks). Interesting results from this logistic regression model were that odds of a reservation being used increase by 5.9% if the user is male, that as the age of the user increases so does the odds of a trip, although only slightly (odds increase 0.2% per year), and that the odds that a subscribed user performs a reserved trip are 7.5% lower than for a user who is not subscribed to the service. In addition, the model shows that having non-zero rates of cancellations, late-cancellations, and no-shows during a 30-day reservation period are good predictors that future reservations would not be used (as indicated by the parameters negative sign), which is not surprising. Perhaps not as obvious, the model also suggests that with each day in advance that a reservation is made, the odds of a trip being made reduces by 6.2%.

### 2.3.2 Model Accuracy Tests

Standard model accuracy tests were performed for the binary and multi-class classification problem. A model’s accuracy is defined as the percent of samples whose class membership is correctly predicted. The data was split into a training data set (on which the prediction models are trained) and a testing data set (used to test the predictive power of the models). Two approaches were used to split the data: the standard random splitting of the dataset, with 80% of the data allocated for training and 20% for testing, and a time-based split to simulate its use in practice. In the first approach, all the data was used in the prediction exercise, whereas in the second approach the data from the period of January 2014 to De-
December 2015 was used for the model training and the data for 2016-2017 period was used for the prediction tests.

The results of the model accuracy tests for the binary and multiclass classification problems are reported in Table 3. Random forest was the best performing model in the case of the binary and multiclass classification problems in which random data splits were used as the test sample, with accuracies of 77% and 74% respectively. In the 2016-2017 test sample, the gradient boosting ensemble was the most accurate, with accuracies of 70% for the binary classification problem, and 69% for the multiclass problem. For context, over the whole sample 67.3% of reservations resulted in trips, while in the 2016-2017 period this was the case for 66.6% of reservations. Therefore, although the best models perform better than random classification (as indicated by the fact that the area under the receiver operating characteristic curve in Table 3 is not equal to 0.50 for any method), they perform only marginally better than the accuracy that would be obtained if every reservation had been assigned to the performed class.

Table 2: Coefficients and Odds-Ratio for a Simple Logistic Regression Model

<table>
<thead>
<tr>
<th>Variable (Feature)</th>
<th>$\beta$</th>
<th>St. Dev.</th>
<th>p-value</th>
<th>Odds-ratio ($\exp(\beta)$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1.821</td>
<td>0.015</td>
<td>&lt;0.001</td>
<td>-</td>
</tr>
<tr>
<td>Age</td>
<td>0.002</td>
<td>0.000</td>
<td>&lt;0.001</td>
<td>1.002</td>
</tr>
<tr>
<td>Male Dummy</td>
<td>0.058</td>
<td>0.005</td>
<td>&lt;0.001</td>
<td>1.059</td>
</tr>
<tr>
<td>Disability: Kidney Disease</td>
<td>0.132</td>
<td>0.013</td>
<td>&lt;0.001</td>
<td>1.141</td>
</tr>
<tr>
<td>User Type: Subscriber</td>
<td>-0.078</td>
<td>0.008</td>
<td>&lt;0.001</td>
<td>0.925</td>
</tr>
<tr>
<td>Purpose: Medical Appointment</td>
<td>0.066</td>
<td>0.007</td>
<td>&lt;0.001</td>
<td>1.069</td>
</tr>
<tr>
<td>Purpose: Entertainment</td>
<td>0.155</td>
<td>0.008</td>
<td>&lt;0.001</td>
<td>1.168</td>
</tr>
<tr>
<td>Purpose: Education</td>
<td>0.124</td>
<td>0.011</td>
<td>&lt;0.001</td>
<td>1.133</td>
</tr>
<tr>
<td>Purpose: Work</td>
<td>0.216</td>
<td>0.008</td>
<td>&lt;0.001</td>
<td>1.241</td>
</tr>
<tr>
<td>Purpose: Dialysis</td>
<td>0.380</td>
<td>0.013</td>
<td>&lt;0.001</td>
<td>1.462</td>
</tr>
<tr>
<td>Day: Saturday</td>
<td>-0.009</td>
<td>0.001</td>
<td>&lt;0.001</td>
<td>0.991</td>
</tr>
<tr>
<td>Day: Sunday</td>
<td>-0.078</td>
<td>0.009</td>
<td>&lt;0.001</td>
<td>0.925</td>
</tr>
<tr>
<td>Days in Advance</td>
<td>-0.063</td>
<td>0.014</td>
<td>&lt;0.001</td>
<td>0.938</td>
</tr>
<tr>
<td>Cancellation Rate</td>
<td>-3.480</td>
<td>0.013</td>
<td>&lt;0.001</td>
<td>0.031</td>
</tr>
<tr>
<td>Late Cancellation Rate</td>
<td>-3.928</td>
<td>0.026</td>
<td>&lt;0.001</td>
<td>0.020</td>
</tr>
<tr>
<td>No-Show Rate</td>
<td>-3.616</td>
<td>0.052</td>
<td>&lt;0.001</td>
<td>0.027</td>
</tr>
</tbody>
</table>

Number of observations: 978411
Pseudo R-squared: 0.1187
Log-likelihood: -542800
Log-likelihood ratio test p-value: <0.001
Table 3: Accuracy Test Results for Classification Problems

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Sample Data: Random Splits</th>
<th>Test Sample Data: 2016-2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Binary Problem</td>
<td>Multiclass Problem</td>
</tr>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>AUC</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>68</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>0.55</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>73</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>0.54</td>
</tr>
<tr>
<td>Random Forest</td>
<td>77</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>0.61</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>65</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>0.62</td>
</tr>
<tr>
<td>Neural Network</td>
<td>59</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>0.72</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>69</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>0.67</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>71</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0.72</td>
</tr>
<tr>
<td>Voting</td>
<td>74</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>0.60</td>
</tr>
<tr>
<td>Bagging</td>
<td>68</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Note: AUC stands for area under the receiver operating characteristic curve.

2.4 Daily Demand Estimation Tests

Estimates on the number of trips for a day (or, conversely, the number of cancellations) can be useful for making demand management decisions, such as how many trips to overbook in a particular day or how much money to allocated to outsource demand to contractors. Naturally, different questions require different model specifications. Here it is assumed that the operator is interested in predicting the total number of trips in a day. This modeling decision is partly motivated by the fact that the data does not contain information on when trips were cancelled. In this illustrative application the aim is to: i) train a set of models, ii) gather the reservations for each test day, iii) use the trained models to predict which reservations will result in trips, and iv) aggregate those predictions for the tests day. The models were initially trained with data for the January 2014 to December 2015 period, and the tests days were sequentially drawn from the 2016-2017 data (547 test days). The models were retrained with the observed data of test days once predictions were made. The logistic regression, random forest, gradient boosting, and the k-means and k-prototypes clustering methods were used for this exercise. The class-based trips predictions \( t_{C}^{md} \) for day \( d \) by a model \( m \) where computed using:

\[
    t_{C}^{md} = \sum_{r} \hat{y}_{mdr}
\]

where \( \hat{y}_{mdr} \) equals one if model \( m \) predicts that reservation \( r \) will result in a trip, and zero otherwise. \( t_{C}^{md} \) was computed using logistic regression, random forest, and gradient boosting. In addition, the expected total trips \( t_{E}^{md} \) for each day \( d \) were computed using the class membership probabilities generated by the models, as follows:

\[
    t_{E}^{md} = \sum_{r} p_{mdr1}
\]

where \( p_{mdr1} \) is the probability that reservation \( r \) will result in a trip according to model \( m \). The logistic regression, random forest, gradient boosting, and cluster-based methods were used to compute \( t_{C}^{md} \). For a baseline comparison, daily demand was also predicted by multiplying the historical base share of reservations that result in trips by the total number
of reservations of each test day. As the proportion is somewhat stable (see Figure 1), it was expected that this simple method would produce relatively good results.

Table 4 reports the mean absolute percent error (MAPE) of the demand predictions for each method, as well as the error standard deviation. The k-means $t^E_{md}$ predictions resulted in the lowest MAPE, followed by the k-prototype $t^E_{md}$ predictions and $t^C_{md}$ predictions random forest. The k-means prediction error was 1.1% lower than the error obtained by predicting total demand using the base share. Interestingly, for the non-clustering methods the $t^C_{md}$ prediction generally overestimated the demand while the $t^E_{md}$ generally underestimated the demand. This result suggests that an ensemble of the aggregate demand predictions could be generated to produce a more accurate model.

<table>
<thead>
<tr>
<th>Aggregation Type</th>
<th>Method</th>
<th>MAPE</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3^*t^C_{md}$</td>
<td>Logistic Regression</td>
<td>53.5</td>
<td>58.3</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>13.9</td>
<td>28.1</td>
</tr>
<tr>
<td></td>
<td>Gradient Boosting</td>
<td>28.8</td>
<td>35.7</td>
</tr>
<tr>
<td>$6^*t^E_{md}$</td>
<td>Logistic Regression</td>
<td>51.6</td>
<td>12.4</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>52.2</td>
<td>22.4</td>
</tr>
<tr>
<td></td>
<td>Gradient Boosting</td>
<td>52.0</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>K-means clustering</td>
<td><strong>13.8</strong></td>
<td>36.2</td>
</tr>
<tr>
<td></td>
<td>K-prototypes</td>
<td>13.9</td>
<td>36.4</td>
</tr>
<tr>
<td></td>
<td>Base share</td>
<td>14.0</td>
<td>37.0</td>
</tr>
</tbody>
</table>

### 2.4.1 Scenario Generation for Vehicle Scheduling

Demand stochasticity in pick-up and delivery problems can be handled using the sample average approximation method (Ghilas et al., 2016). As part of this solution approach, demand scenarios are generated, which are then used to produce robust vehicle schedules. A demand scenario is a set of hypothetical service requests that are generated using probability distributions. In the context of paratransit operations, a demand scenario would be a simulation of outcomes for a set of reservations; based on these scenarios robust vehicle schedules can be constructed. The machine learning algorithms considered in this study could be used to generate outcome class probabilities, at the reservation-level, to simulate the service scenarios. Note that the focus is on generating a representative sample of possible service scenarios given that a full enumeration of all the possible reservation scenarios would be prohibitive in most practical applications. The best predictive model would be the one whose class probabilities resulted, on average, on scenario sets that more closely resembles the observed sets of reservation outcomes.

As before, the data is divided into the January 2014-December 2015 training set and the 2016-2017 testing set. For each test day and for each model, the objective is to generate $S$ service scenarios via Monte Carlo simulation. In the case of the binary problem, for each scenario $s$, test day $d$ and reservation $r$, the probability of performing a trip $p^*_{mdvr1}$ is determined using model $m$. A random number $w$ is drawn from a standard uniform distribution $U(0,1)$, and if $w \leq p_{mdvr1}$ the reservation is labeled as a trip; otherwise, in that
simulation the reservation does not result in a trip. A similar procedure is used for the multiclass problem.

In this application, the class probabilities computed by the random forest and cluster-based models were used to generate 100 scenarios for each test day. The prediction performance metric used for this exercise was the average Hamming loss ($H^m$). Let $\hat{y}_{mdrsc}$ equal one if reservation $r$ in day $d$ is assigned to class $c$ by model $m$ in scenario $s$, and zero otherwise. Also, let $y_{drc}$ equal one if reservation $r$ on day $d$ is observed to belong to class $c$. $H^m$ is defined here as:

$$H^m = \frac{1}{D} \sum_{d=1}^{D} \frac{1}{S} \sum_{s=1}^{S} \frac{1}{R_d} \sum_{r=1}^{R_d} \sum_{c=1}^{C} (|\hat{y}_{mdrsc} - 1|) y_{drc}$$

(3)

where $D$ is the number of test days, $S$ is the number of scenarios generated per test day, and $R_d$ is the number of reservations on day $d$. Table 5 reports on the result of the simulation runs, which show that the scenarios generated using the gradient booting probabilities were, on average, closer to what was actually observed. The gradient boosting scenarios had an average error $H^m$ of 39% and 43% for the binary and multiclass cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>Binary Classification</th>
<th>Multiclass Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>42</td>
<td>49</td>
</tr>
<tr>
<td>Random Forest</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>39</td>
<td>43</td>
</tr>
<tr>
<td>K-means</td>
<td>43</td>
<td>47</td>
</tr>
</tbody>
</table>

### 2.5 Closing Remarks

Changing demographics and emerging mobility options present unique challenges to paratransit systems, which are best addressed by the continued adaptation and optimization of the procedures used to manage paratransit demand and services. The results reported in this section suggest that machine learning algorithms could be among the tools used by transit agencies for these purposes. The tests results show that machine learning algorithms have predictive capabilities even when fitted using limited data, such as the MBA data used here. And the applications presented illustrated the models’ use for short-term demand prediction, as well as to generate sets of possible service scenarios. The former application could be a component of an online booking system that makes overbooking decisions, while the latter application could be part of a robust vehicle scheduling system that accounts for reservation uncertainties.

Machine learning algorithms can complement the professional judgment and rules of thumb used in paratransit planning and operations by objectively formalizing what is subjectively understood. For example, the MBA operators know that reservations by dialysis patients are likely to result in trips, and they have a sense of which users are likely to cancel their reservations based on past behavior. These subjective impressions can be converted
into specific model elements of a cohesive decision-making process. Even when the advantage of machine learning algorithms over simpler methods is minor (such as in the case of the base share demand model), they can be useful to operators as they offer an examinable and adaptive basis on which to understand a system and make decisions. And fortunately, the cost of using machine learning algorithms continues to drop, as free online courses continue to be developed that can be used to train transit analysts on the use of powerful, open-source machine learning software.

Future research could explore the specification and performance of machine learning models for real-time cancellation predictions in the context of a paratransit reservation booking systems, with the ultimate aim of generating overbooking models. The logistic regression model results presented in Table (2) suggest that knowing the timing of reservations is significant for prediction purposes, which suggests that an online system for cancellation prediction might be useful for demand management in paratransit. The results for the clustering methods and the base share method in the demand prediction exercise also suggest that simple prediction methods could produce relatively low prediction errors. For example, future research could explore the specification of a rule-base prediction model in which outcome probabilities are computed and assigned based on a predefined set of user categories. The study of the effects of incorporating spatial information (e.g., trip origins and destinations) into machine learning models for the prediction of reservation outcomes is another useful research direction.
3 Paratransit Routing Models and Overbooking

Paratransit systems managed by public transit agencies typically operate in a static environment where trip requests are received at least a day in advance. Routing and scheduling decisions are commonly made according to the output of commercial software that attempt to minimize operational costs subject to service constraints. To our knowledge, no formal methodology exists in the literature that can be used to make overbooking decisions in paratransit systems, although there are examples of paratransit systems that in practice are overbooking trips in order to improve performance (Nelson and Associates, 2007). This project develops new routing and booking models that are integrated to the data analytics methods previously discussed in Section 2.

Consider a static scheduling problem where decisions about trip reservations are made at least a day prior to the requested trip; this is a standard booking regime in paratransit, and mathematically, this is often referred as the Dial-A-Ride Problem (DARP). In a DARP it is generally assumed that each user makes a trip request \( i \) with pickup and drop off locations indexed \((i, i + n)\), and their corresponding time windows \([a_i, b_i], [a_{i+n}, b_{i+n}]\). The agency attempts to make routing \((X)\) and timing \((T)\) decisions that minimize costs (e.g., operating cost, user cost, total size of required vehicles) subject to the time window constraints imposed by each request \( i \) and related quality of service constraints. Additional information, such as the total number of passengers and vehicle specifications, are captured by \( Y \). The information in \( Y \) is used to formulate basic constraints like total service capacity constraints. DARP formulations and associated solution heuristics have been widely studied, with extensions proposed to account for selective service, dynamic trip requests, and stochastic travel environments (e.g., Cordeau and Laporte (2007); Parragh et al. (2008); Fu (2002)).

In the presence of high cancellation and no-show rates, operator’s route plan is changed as it realizes a no-show or a cancellation. Both no-show and at-the-door cancellation eliminates the requirement of traveling to its drop-off location, increasing the capacity of the service with the same fleet and drivers. Changes in operational requirement can be accommodated by Dynamic variation of DARP as seen in Chassaing et al. (2015); Ichoua et al. (2000) (Section ??). However, the outcomes of each trip (served, at-the-door cancellation, or no-show) is unknown to operator at the time of reservation as well as service planning. Built upon DARP route planning model, we utilize a Monte-Carlo scenario-based approach of evaluating overbooking with the classification predictions from Section 3.2.

The setup of the overbooking problem is as follows. A user calls the operator and makes a request \( i \) at time \( t \) with time windows \([a_i, b_i]\) and \([a_{i+n}, b_{i+n}]\), with a particular set of pick-up and drop-off locations indexed \((i, i + n)\). The operator must then decide whether to reject or accept the trip request (i.e., the operator is a sequential scheduler). A request is accepted when it can be served with greater than the preset \( \alpha \)% confidence of the possible outcomes of trip reservations. This confidence level is computed via a Monte Carlo feasibility assessment. Assume that \(|S|\) scenarios are generated and \( n_f \) as the number of feasible scenarios. If \( \beta = \frac{n_f}{|S|} > \alpha \), we accept this new request. An overview of the scheme is presented in Figure 4.
3.1 Monte Carlo Scenario-Based Feasibility Assessment

Consider a paratransit system with \( n \) trip requests in a particular day. The operator knows that there is a probability that some of those \( n \) requests will result in no-shows or cancellations. Given the probability information from the classification models, different trip realization (or demand) scenarios can be generated. These scenarios represent sets of potential class membership realizations for each request (i.e., hypothetical assignments of each trip request into the normal trip, no-show, or cancellation classes). To generate these scenarios a Monte Carlo simulation is used. Let \( \theta \) be the probability information from the classification model. Let \( S \) represent the set of demand scenarios, and \( s \in S \) denote the index for a particular scenario. Also, let \( y_s \) represent the vector of all class membership assignments based on \( \theta \) in scenario \( s \) for the \( n \) requests, with each element \( y_{si} \in y_s \) indicating if trip request \( i, (i = \{1, 2, 3\}) \) is performed, a no-show, or an at-the-door cancellation \( (y_s = \{y_{s1}, y_{s2}, y_{s3}\}) \).

For each scenario \( s \) and outcomes of each trip \( y_s = \{y_{s1}, y_{s2}, y_{s3}\} \), the minimum operational cost and feasibility can be determined running a number of scenarios of trip outcomes and whether the dynamic DARP with cancellations can generate a feasible solution each time.

Cancellations and no-shows have received little attention in the DARP literature. To our knowledge, only the general stochasticity in travel environments (e.g., travel times and demand uncertainties) are considered (Xiang et al., 2008; Ichoua et al., 2000; Chassaing et al., 2015) but not explicitly cancellations. In addition, these studies develop dynamic versions that can be applied to reoptimize routes in the presence of stochasticity. Dynamic DARP are undoubtedly useful, but they are not applicable to the planning processes used in most paratransit systems, as these processes begin generating vehicles routes and schedules prior to the start of a service period and before the occurrence of any no-show or late cancellation. Therefore, the planning procedures used by operators can be advanced by predictive analytics that can quantify the uncertainties associated with trip requests and
by the proposed overbooking framework that incorporate these uncertainties. We note that proposed framework is developed based on this dynamic re-optimization approach.

We integrate the classification model outcomes \( y_s = \{y_{s1}, y_{s2}, y_{s3}\} \) for scenario \( s \) into paratransit routing and schedule planning called the Dynamic DARP with Cancellations (3.2) where requests are classified into three types of outcomes (performed, no-show and at-the-door cancellation). It is noted that this Dynamic DARP with Cancellations optimizes the outcome knowing all the outcomes. However, for actual operation, the operational cost will be higher as the planned route should include all reservations, and the performed route is an outcome of this planned route updated as the trip outcomes are realized.

### 3.2 Dynamic Dial-A-Ride Problem with Cancellations

In each Monte-Carlo scenario, \( s \), a dynamic DARP with Cancellations is run given the actual outcome of reservations, \( y_s = \{y_{s1}, y_{s2}, y_{s3}\} \). As this information is not available to operators at the time of planning, the uncertainty of the trip outcomes makes the DARP dynamic. A common way to deal with dynamic models is adaptation of Static Algorithm Chassaing et al. (2015). Such method is built on the rolling horizon approach that every time the operator (driver) realizes the request is an at-the-door cancellation or a no-show, DARP with Cancellation reoptimizes schedules. We note that this approach is built on the work the by Ichoua et al. (2000).

This dynamic DARP with Cancellations has two stages. In the first stage, we need to obtain the initial solution with DARP. That is, we generate a solution of serving all requests using Insertion Method for DARP. If the solution is infeasible, this means we cannot serve all requests assuming that there are no cancellations. In this case, we use the Insertion Method with Waitlist to create initial infeasible solution (sec 3.2.1). Both feasible and infeasible cases, DARP output schedule is executed serving trip \( i \) in sequence. When we realize a at-the-door cancellation or a no-show, DARP with Cancellation is run to update the schedule as in Figure (5).

![Figure 5: Dynamic Dial-A-Ride-Problem with Cancellations](image-url)
3.2.1 Dial-A-Ride Problem

We follow the notations and formulation established in Cordeau and Laporte (2003). Let there be \( n \) customers indexed by \( i \). Associate to the pickup location of customer \( i \) at node \( i \) and to his delivery location a node \( n + i \). Also associate to the depot, nodes 0 and \( 2n + 1 \). This creates a clear distinction between customers, their associated locations and the nodes of the network. Note that different nodes may correspond to the same physical location. The following table shows all notation we used in formulation:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>( {0, 1, 2, ..., n, n + 1, n + 2, ...2n, 2n + 1} )</td>
</tr>
<tr>
<td>( P^+ )</td>
<td>Set of pickup node ( {1, 2, ..., n} )</td>
</tr>
<tr>
<td>( P^- )</td>
<td>set of delivery node ( {n + 1, n + 2, ...2n} )</td>
</tr>
<tr>
<td>([a_i, b_i])</td>
<td>pickup time window for customer ( i )</td>
</tr>
<tr>
<td>([a_{n+i}, b_{n+i}])</td>
<td>delivery time window.</td>
</tr>
<tr>
<td>( s_i )</td>
<td>service time for request</td>
</tr>
<tr>
<td>( V )</td>
<td>Set of vehicles</td>
</tr>
</tbody>
</table>

Table 6: Notation for Dial-A-Ride Problem

Three types of variables are used in the mathematical formulation: binary flow variables \( X_{ij}^v, v \in V, \ i, j \in N, i \neq j \), time variables \( T_i, i \in P \) and \( T_0^v, T_{2n+1}^v, v \in V \), and load variables \( Y_i, i \in P \).
Max $\sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ij}^v$ \hfill (4) \\

$\sum_{v \in V} \sum_{j \in N} X_{ij}^v = 1, \quad i \in P^+$ \hfill (5) \\

$\sum_{j \in N} X_{ij}^v - \sum_{j \in N} X_{ji}^v = 0, \quad i \in P, v \in V$ \hfill (6) \\

$\sum_{j \in P} X_{0j}^v = 1, \quad v \in V$ \hfill (7) \\

$\sum_{j \in P} X_{i2n+1}^v = 1, \quad v \in V$ \hfill (8) \\

$\sum_{j \in N} X_{ij}^v - \sum_{j \in N} X_{j,n+i}^v = 0, \quad i \in P^+, v \in V$ \hfill (9) \\

$T_i + s_i + t_{i,n+i} \leq T_{n+i}, \quad i \in P^+$ \hfill (10) \\

$X_{ij}^v = 1 \Rightarrow T_i + s_i + t_{ij} \leq T_j, \quad i, j \in P, v \in V$ \hfill (11) \\

$X_{ij}^v = 1 \Rightarrow T_i + s_i + w_i + t_{ij} \leq T_j, \quad i \in P^+, j \in P, v \in V$ \hfill (12) \\

$X_{0j}^v = 1 \Rightarrow T_{0j}^v + t_{0j} \leq T_j, \quad j \in P^+, v \in V$ \hfill (13) \\

$X_{i,2n+1}^v = 1 \Rightarrow T_i + s_i + t_{i,2n+1} \leq T_{2n+1}^w, \quad j \in P^-, v \in V$ \hfill (14) \\

$a_i \leq T_i \leq b_i, \quad i \in P$ \hfill (15) \\

$a_0 \leq T_0^w \leq b_0, \quad v \in V$ \hfill (16) \\

$a_{2n+1} \leq T_{2n+1}^w \leq b_{2n+1}, \quad v \in V$ \hfill (17) \\

$X_{ij}^v = 1 \Rightarrow Y_i + d_j = Y_j, \quad i \in P, j \in P^+, v \in V$ \hfill (18) \\

$X_{ij}^v = 1 \Rightarrow Y_i - d_{j-n} = Y_j, \quad i \in P, j \in P^-, v \in V$ \hfill (19) \\

$X_{0j}^v = 1 \Rightarrow Y_0 + d_j = Y_j, \quad j \in P^+, v \in V$ \hfill (20) \\

$Y_0 = 0, \quad d_i \leq Y_i \leq D, \quad i \in P^+$ \hfill (21) \\

$X_{ij}^v \text{ binary, } i, j \in N, v \in V$ \hfill (22) \\

(23)
Constraints (5)-(9) are flow balance constraints. The constraint (5) make sure that drivers pick up all the demand. Constraint (6) is flow balance constraint. Constraint (7) and (8) make sure that vehicle depart from depot and return to depot. Constraints (9) make sure that vehicle will go to the destination of pick up node. Constraints (10)-(14) is the time window constraints. Constraints (18)-(20) is capacity constraints.

**Insertion Method for DARP** Directly solve DARP model is not efficient, especially when the number of request becomes large. Further more, in the overbooking stage, checking whether the request can be successfully severed is more important than finding the best route for buses. So, in the overbooking stage, we will develop variations of insertion method developed by Jaw et al. (1986) to find the solution. Insertion method is a heuristic method that insert the request sequentially while each request follows the constraints. By manipulating different insertion location for each request, we are going to pick the best few of them for the future requests so that optimize the overall schedule.

The algorithm process requests in sequence. For each request $i$, the process goes as follows:

**Step 1:** for each vehicle $j$ ($j = 1, 2, 3,...$) Check all possible insertion schedule for vehicle $j$. If it is infeasible, go to next vehicle $j + 1$ (go to step 1). If it is feasible, evaluate minimum additional time cost caused by adding this request (Step 2).

**Step 2:** If it is infeasible to insert request to any of vehicles, declare that the customer can not be successfully served. After insert one request, we have a list of insertion schedule. We pick the best few of them to enter the insertion process for next request.

The most important part of insertion method is to examine the minimum additional time cost. To illustrate the insertion process, let’s assume that we are going to assign $i$ request between $m$ and $n$ like picture show in figure 6:

![Figure 6: Insertion Example](image)

The earliest arrive of $+m$ is $a_{+m}$, the latest leaving time is $b_{+m}$ and the actual arrive time is $t_{+m}$. The earliest arrive time of $+n$ is $a_{+n}$, the latest leaving time is $b_{+n}$ and the actual arrive time is $t_{+n}$. If we want to insert request $+i$ between $+m$ and $+n$ and his earliest arrive time is $a_{+i}$ and latest leaving time is $b_{+i}$. The earliest actual arrive time is $Min\{t_{+m}, a_{+i}\}$. The latest leave time is $Min\{t_{+n}, b_{+i}\}$. We use the earliest arrive time as best insertion point.

**Insertion Method with Waitlist for DARP** When we meet the requests that are unable to be inserted to any of current vehicle routes (i.e., the DARP is infeasible), and we expect that there is a chance for them to be able to inserted in the future because of some
requests will be no-shows or at-the-door cancellations. We create a waitlist for them and try to re-insert the requests when no-show or at-the-door cancellation occurs. This is to generate a schedule even when the problem itself is infeasible for the proposed overbooking scheme.

### 3.2.2 Dial-A-Ride Problem with Cancellations

DARP with Cancellations is run everytime a late-cancellation or a no-show is realized. The service time for the buses that encounter the no-show requests will increase by $w_i$, $\forall i \in P^+_n$. Since the bus driver will wait certain time before confirming that this is a no-show.

Different from standard DARP, we have three types of requests during operation. Show-up requests (performed, normal), no-show requests and at-the-door cancellation requests. We denoted them as follow:

- $P^+_s$: Set of pickup node for show up requests
- $P^-_s$: set of delivery node for show up requests
- $P^+_n$: Set of pickup node for no-show requests
- $P^-_n$: Set of delivery node for no-show requests
- $P^+_l$: Set of pickup node for at-the-door cancellation requests
- $P^-_l$: Set of pickup node for at-the-door cancellation requests

<table>
<thead>
<tr>
<th>Table 7: Notation for Three Request Outcomes</th>
</tr>
</thead>
</table>
| $P^+_s$ & Set of pickup node for show up requests  
$P^-_s$ & set of delivery node for show up requests  
$P^+_n$ & Set of pickup node for no-show requests  
$P^-_n$ & Set of delivery node for no-show requests  
$P^+_l$ & Set of pickup node for at-the-door cancellation requests  
$P^-_l$ & Set of pickup node for at-the-door cancellation requests |

For the no-show and at-the-door cancellation requests, constraint (24) and (25) make sure vehicles will pick them up but will not go to their destination.

\[
\sum_{v \in V} \sum_{j \in N} X^v_{ij} = 1, \quad i \in P^+ \\
\sum_{j \in N} X^k_{ij} = 0, \quad i \in P^-_n \cup P^-_l, \quad v \in V
\]

Constraint (26) make sure that when vehicles visit no-show request, they will have to wait $w_i$ time to make sure no one is going to show up. Constraint (27) says that when a request is cancelled right after vehicle arrives, the vehicle will leave right away without serving.

\[
X^v_{ij} = 1 \Rightarrow T_i + w_i + t_{ij} \leq T_j, \quad i \in P^+_n, j \in P, v \in V
\]

\[
X^v_{ij} = 1 \Rightarrow T_i + t_{ij} \leq T_j, \quad i \in P^+_l, j \in P, v \in V
\]

**Network Update**

It is noted as this is realized in the middle of operation when an operator realizes a no-show or an at-the-door cancellation. Then, routes for the entire fleet are reoptimized. For the vehicles that are on the way to the next stop, a dummy node is created for each vehicle. And the customers that they have already taken will remain in their routes.
For the rest of request that are not yet served, we run DARP with Cancellation including pending requests in the waitlist.

Specifically, assume that an at-the-door cancellation or a no-show request occurs at time, $T_o$. A dummy node is created for each vehicle to record the current information. For the vehicle that is at the pickup location of the no-show or at-the-door cancellation request, a dummy node is not needed. All the request that is taken before $T_o$ will remain in this vehicle (e.g. request $i$ in vehicle 1). All the request that is served after $T_o$ will be regarded as unserved in re-optimization process.

For the other vehicles that is on their way to next destination. We will create dummy node for them which is represented in Figure (9), the node that lay on the vertical line. The location for the dummy node is proportional to the distance from current original location to destination. The requests that taken before $T_o$ remain in vehicles, the requests that taken after $T_o$ will be un-served. All the white node in Figure (7a) is the request that need to be assigned in the re-optimization process. We call this process **Network Update**.

**Insertion Method for Reoptimizing DARP with Cancellations** The re-optimization process follows the Network Update. We collect two kinds of requests. The first is the destination of request that is already be taken but have not been delivered yet (e.g. request $i$). The second request is the requests that is not taken yet (e.g. request $k$). In the first stage we insert the destination of first kind of requests to the schedule. In the second stage, we mixed the second kind of requests with the requests in the pending list. And we insert them into the schedule.
3.2.3 Illustrative Example

We generate network with 4 requests. Node 0 is the depot for vehicles. Node \{1, 2, 3, 4\} are pick up nodes and nodes \{5, 6, 7, 8\} are delivery nodes. Among them, request 3 is the no-show request. Two vehicles are dispatched to serve these requests. The capacity of these vehicles is two.

Figure (8a) shows the initial planning solution from the DARP. Vehicle one takes requests 1 and 3, and go to corresponding destination. Vehicle two takes requests 2 and 4 in sequence.
and go to corresponding destination.

Figure 8: Dynamic DARP with Cancellations: Illustrative Example

During the service, the first vehicle realize that requests 3 is an at-the-door cancellation
request which showed on figure (8b). At this time segment, vehicle 2 is on the way to node 6. So a dummy node $D$ for the second vehicle is created and the travel time between new created dummy node and other nodes is updated (Network Update). Based on this new network, the model re-schedule the routes for two vehicles and assigned request 4 to vehicle 1 which was assigned to vehicle 2 in the initial solution.

### 3.3 Case study

We use the paratransit reservation data from the Metropolitan Bus Authority (MBA) of Puerto Rico which they have the booking date and scheduled time for pick up. The request vary from 200 to 600 for each day. The demand in weekend is relatively small but in average, it is about 400-500 requests daily. We use one weekday’s data which has 545 requests. Each requests have its own probability to cancel the requests one days before. Requests that are not cancelled one day before have 10% chance to be no-show and 10% probability of at-the-door cancellation requests (i.e., $y_s = \{ y_{normal} = 80\%, y_{at-the-door} = 10\%, y_{no-show} = 10\% \}$). We set the feasibility rate criteria as $\alpha = 80\%$. This means, that the probability that we can successfully serve all the customers even with overbooking is 80%. Due to data unavailability, the pickup and drop-off locations are randomly generated in 100 miles × 100 miles map. Each vehicle can take 3 customers and the bus speed is 10 miles/hour. The total number of vehicles is 10.

We apply the insertion method to make decision for overbooking. For each incoming request, we generate $|S| = 10$ scenarios. The result are show in Figure (9).

![Figure 9: Case Study 2](image)

The result is compared with a regular booking which is without overbooking. The strategy for regular booking is, if coming request can be served (i.e., DARP is feasible), we accept it. If not, we reject it. Red represents the rejected requests. Green represents accepted requests. For overbooking strategy, the total of 507 requests are reserved whereas only 199 requests are accepted without overbooking. 308 requests were infeasible but accepted with overbooking.

We further analyze the quality of the solution by comparing the DARP with Cancellations and the Dynamic DARP with Cancellations to solve the scenario with 200 requests accepted
to be served. It is infeasible to accept all these requests if we do not consider overbooking (i.e., DARP is infeasible). The total cost (travel times in minutes) of each of the scenarios are listed in the following table.

<table>
<thead>
<tr>
<th>Scenario No</th>
<th>DARP excluding C</th>
<th>DARP with C</th>
<th>Dynamic DARP with C</th>
<th>Cost of Cancellations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22697</td>
<td>22896</td>
<td>28704</td>
<td>25.3%</td>
</tr>
<tr>
<td>2</td>
<td>22425</td>
<td>22448</td>
<td>23857</td>
<td>5.3%</td>
</tr>
<tr>
<td>3</td>
<td>21539</td>
<td>21559</td>
<td>23704</td>
<td>9.9%</td>
</tr>
<tr>
<td>4</td>
<td>21586</td>
<td>22390</td>
<td>23288</td>
<td>4.0%</td>
</tr>
<tr>
<td>5</td>
<td>21412</td>
<td>22630</td>
<td>29077</td>
<td>28.0%</td>
</tr>
<tr>
<td>6</td>
<td>22663</td>
<td>23189</td>
<td>29439</td>
<td>26.9%</td>
</tr>
<tr>
<td>7</td>
<td>21518</td>
<td>21810</td>
<td>26488</td>
<td>21.4%</td>
</tr>
<tr>
<td>8</td>
<td>20223</td>
<td>20681</td>
<td>22022</td>
<td>6.5%</td>
</tr>
<tr>
<td>9</td>
<td>21451</td>
<td>21471</td>
<td>26492</td>
<td>23.3%</td>
</tr>
<tr>
<td>10</td>
<td>20586</td>
<td>22703</td>
<td>25397</td>
<td>11.8%</td>
</tr>
</tbody>
</table>

Table 8: Comparison of Different DARP Settings

The result from DARP with Cancellations is a lower bound cost for paratransit service since we assume that we know the no-show and at-the-door cancellation requests before operation in DARP with Cancellations. However, during the actual operations, Dynamic DARP with Cancellations has no clue about which requests will be no-show or at-the-door cancellation. Therefore, DARP with Cancellations represent the absolute lowest cost of this operation. DARP excluding all Cancellations represent the cost of serving the requests when no-shows and at-the-door cancellations do not exist and these trip requests are removed from the service list. The difference between DARP excluding no-shows and at-the-door cancellations (Column 2) and Dynamic DARP with Cancellations (Column 4) represents the actual cost of these cancellations.

### 3.4 Concluding Remarks

In order to increase operational efficiency of paratransit services with the presence of high cancellation rates, we propose an overbooking strategy that accepts a request when the probability of service feasibility is \( \beta \). This service feasibility is checked by the framework, Dynamic DARP with Cancellations, that starts with classic DARP of serving all trips and re-optimizes the schedule as no-shows or at-the-door cancellations occur. Even when the DARP model is infeasible, this Dynamic DARP with Cancellation may suggest accepting a service request (overbooking) anticipating cancellations.

Currently, research team is extending this work. First, five classes of reservation outcomes are considered instead of three: ”performed”, ”early cancellations”, ”same day cancellations (cancellation before pickup time”, ”at-the-door cancellations”, and ”no-shows”. Second, in addition to overbooking, a new DARP-based route planning (serving all requests) is being developed. This approach as a planning model, considers serving all trips, does not only minimizes the minimum cost but generates more robust routes that anticipates cancellations.
throughout the day. Lastly, we plan to conduct a case study based on real data through collaborations with paratransit agencies.
References


