Meeting the Clock: Real-Time Optimization of Johns Hopkins Shuttle Services Using Dial-a-Ride Algorithms

Aryavrat Gupta

Abstract

This paper presents a comprehensive analysis of the shuttle services at Johns Hopkins University, identifying key inefficiencies and proposing optimization solutions using the Dial-a-Ride Problem (DARP) framework. The study begins by delineating the existing problems with the shuttle system, including suboptimal shuttle allocation and the resultant financial burdens. It then models these challenges within the multi-vehicle DARP framework, defining relevant variables and formulating both linear programming solutions and heuristic approaches. The research incorporates a thorough literature review, notably incorporating findings from Vallée et al. and Häll et al., to adapt dynamic and static DARP solutions respectively. Additionally, the paper extends its theoretical underpinnings by analyzing the $O((\log n)^3)$-approximation algorithm by Inge Li Gørtz, Viswanath Nagarajan, and R. Ravi, which aligns closely with the capacitated demands of the university's shuttle system. This integration of dynamic and static models, supported by empirical data and advanced algorithms, offers robust strategies for enhancing shuttle service efficiency and responsiveness.

1 Introduction

Transportation within university campuses is crucial for maintaining timely academic schedules and ensuring that students can navigate large campuses effectively. At Johns Hopkins University, the Blue Jay shuttles[1] serve as a vital link, offering students a reliable mode of transport within a specific radius around the campus. However, with the dynamic nature of academic schedules and real-time demands of student life, traditional scheduling methods often fall short in efficiency and flexibility. This paper introduces a novel optimization framework for the Blue Jay shuttle service, employing a dynamic Dial-a-Ride Problem (DARP) approach to enhance the responsiveness and efficiency of this critical service.

In response to the growing need for adaptive transport solutions in educational institutions, this study leverages advanced routing algorithms to manage the shuttle fleet. The aim is to address the inherent variability in demand and ensure timely transportation for all users. This approach not only improves operational efficiency but also significantly enhances user satisfaction by reducing wait times and ensuring adherence to a strict 30-minute delivery window from booking to drop-off.

This paper explores the challenges associated with traditional fixed-route shuttle services and demonstrates how a dynamic, demand-responsive system can overcome these limitations. By focusing on Johns Hopkins University's specific needs and infrastructure, this study provides valuable insights into the application of sophisticated routing algorithms in a real-world context, setting a precedent for future improvements in campus transportation systems globally.

2 The Problem

2.1 Defining the problem

At Johns Hopkins University, the existing shuttle service faces significant operational challenges that impact its efficiency and cost-effectiveness. Designed to facilitate convenient transportation for students within and around the campus, the system currently falls short in optimizing shuttle allocations

[1] Johns Hopkins University Shuttle Services

in response to real-time student demand. This inefficiency has led the university to supplement transportation needs with third-party services like Uber and Lyft, particularly during operational hours from 6 PM to 2 AM. This solution, while practical, introduces an additional financial burden that could otherwise be mitigated through more effective use of the existing shuttle resources.

Core Issues Identified

1. Suboptimal Shuttle Allocation: The primary issue is the non-optimal allocation of shuttles. The current scheduling and routing system does not adapt dynamically to the fluctuating demand patterns among students. This often results in shuttles running below capacity or, in some cases, completely empty. This inefficiency in allocation not only wastes resources but also fails to meet the actual transportation needs of the students.

2. Cost Implications: The reliance on external ride-sharing services to fill the gap during underperformance of the shuttle system leads to unnecessary expenditures. These costs are a direct consequence of the inability of the current system to effectively manage and deploy shuttle resources based on actual student needs.

3. Student Satisfaction and Service Utilization: Inadequate service leads to lower student satisfaction and possibly a reduction in the usage of the shuttle services. Students might choose alternative modes of transportation if the shuttle service is perceived as unreliable or inefficient.

2.2 Defining the variables

Independent Variables

These are variables that we can manipulate or control in the algorithm to observe how they affect the dependent variables.

- **Number of Shuttles (i)**: Total number of shuttles available during operational hours. This variable directly influences the system's ability to meet the transportation demand.
- **Shuttle Capacity (c_s)**: Maximum number of passengers that each shuttle can accommodate. This is crucial for planning the logistics and ensuring efficiency.
- **Operational Radius (R)**: The maximum distance that the shuttles cover from the center of Johns Hopkins Homewood campus (The Beach). This determines the scope of service provided by the shuttle system.

Dependent Variables

These variables are expected to change in response to the manipulation of the independent variables.

- **Wait Time (w)**: Time that a student waits from booking until being picked up by a shuttle. This variable is essential for assessing the efficiency of the shuttle service.
- **Travel Time (t)**: Duration from pickup to drop-off at the student's desired destination. Shorter travel times are indicative of a more efficient shuttle service.

3 Modeling the Problem

3.1 Mapping the problem to multi-vehicle DARP

The Dial-a-Ride Problem (DARP)[2] [GN23] serves as a robust framework for addressing complex transportation logistics that require both high flexibility and efficiency. Typically, DARP encompasses several distinct characteristics that make it particularly suited to modeling the shuttle service system designed for Johns Hopkins University:

[2] Dial-a Ride Problem

- **Transportation Requests:** Each request specifies exact pickup and drop-off locations, aligning well with our need to manage discrete points of service for students across the campus.
- **Time Windows:** A critical element of DARP is managing time windows effectively. Our shuttle system must accommodate this by ensuring all students are transported within predefined intervals, crucial for maintaining an effective academic schedule.
- **Route Optimization:** At the core of DARP is the optimization of routes. Our objective extends beyond simple cost reduction; we aim to minimize travel time and maximize overall service quality. This ensures that students spend less time in transit and more time engaged in academic and extracurricular activities.
- **Dynamic Requests:** Perhaps most critically, the model must handle requests dynamically. Students may need impromptu rides due to changes in their schedules, necessitating a system that can adapt in real-time to varying demands without compromising on efficiency or service quality.

3.2 Formulation of the problem

Input

- **Graph Representation:** $G = (V, E)$ where V represents the set of vertices (locations for pick-up and drop-off and shuttle locations) and E represents the set of edges connecting these vertices, with weights indicating travel times.
- **Shuttle Information:** A set of available shuttles S, ($|S| = i$), where each shuttle s has:
  - A capacity c_s
  - The current shuttle location $l_s \in V$
  - Operating constraints such as operational hours (typically 6:00PM - 2:00AM).
- **Requests (dynamic):** A set of ride requests R_{req}, where each request r includes:
  - A pickup location $p_r \in V$
  - A drop-off location $d_r \in V$
  - A time window $[a_r, b_r]$ within which the passenger must be picked up
  - A maximum ride time t_{max_r} to ensure service quality from pickup to drop-off
  - Number of riders $n_r < c_s$
- **Operational Parameters:** Radius R within which the shuttles can operate, and base locations or depots where shuttles can start or end their shifts.

Feasible Solution

A feasible solution to this DARP involves creating a set of routes $\{P\}$ for each shuttle s such that:

- **Capacity Constraints:** At no point should the number of passengers in shuttle s exceed its capacity c_s.
- **Service Constraints:** Each request r must be served exactly once, with the passenger being picked up and dropped off within their specified time windows $[a_r, b_r]$.
- **Ride Time Constraints:** The total time from pickup to drop-off for each request r must not exceed t_{max_r}.
- **Route Validity:** Each route P_s for shuttle s must start and end at a designated depot or within operational constraints like time limits or geographical bounds.

Objective Function

The objective function aims to optimize the operation based on several possible criteria. Here are a few common objectives:

- **Minimize Total Travel Time:** $\min \sum_{s \in S} \sum_{(i,j) \in P_s} d_{ij}$ where d_{ij} is interpreted as the travel time from location i to j in route P_s.

4 Graph Visualization Example and Key Assumptions

Given below is a graph $G = (V, E)$[3] where V represents the set locations for pick-up and drop-off (and shuttle locations) and E represents the set of edges connecting these locations, with weights indicating travel times (assume travel times are proportional to distances in this example). In this example, there are three shuttles at vertices I, N and J ($|S| = i = 3$), the operational radius R of shuttles is 20 units ($R = 20$ units) and all shuttles have a capacity of 12 people ($c_s = 12$).

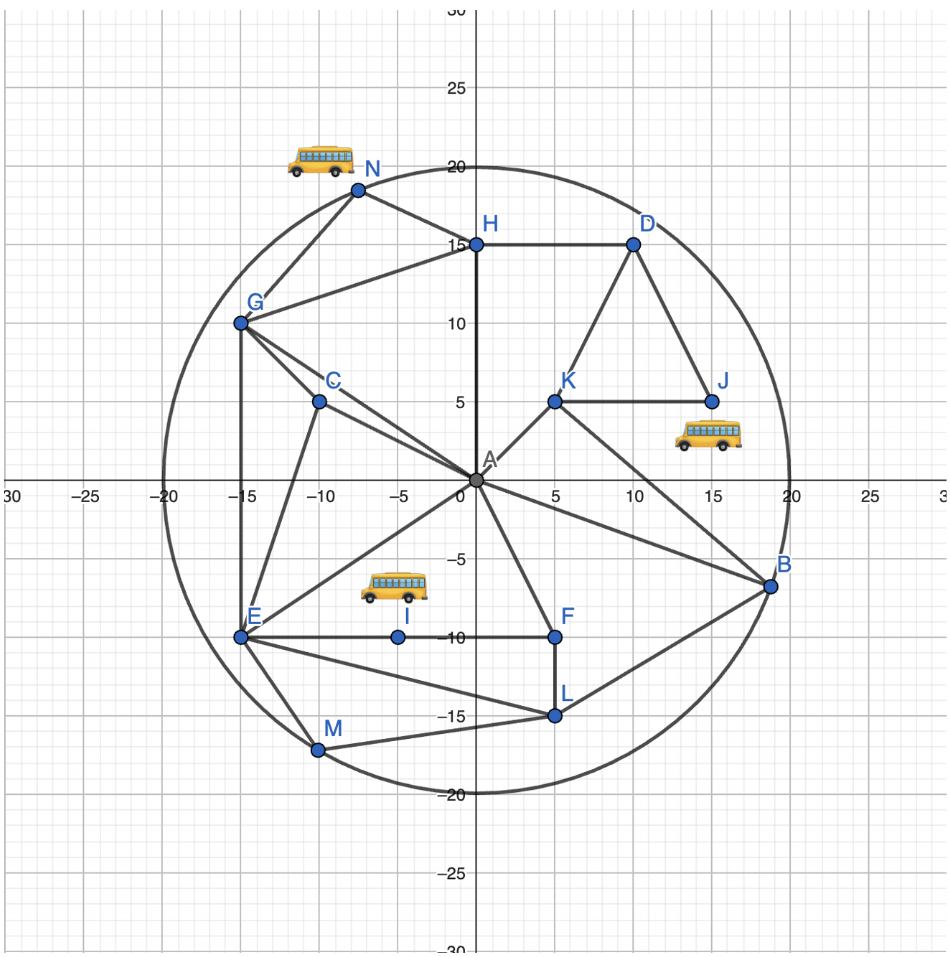

Figure 1: Dial-a-Ride Problem Visualization for Shuttle Services

Key Assumptions: For simplicity, let's assume the maximum time window w_{max} and maximum ride time t_{max} are constant constraints which are 10 minutes and 30 minutes respectively i.e. $w_{max} = 10$, $t_{max} = 30$. Also, let's assume we get ride requests uniformly at random i.e. each request is equally likely to occur at any given time or location, assuming a constant probability across the specified domain (requests are added to the set R_{req} correspondingly).

[3] Made with Geogebra

For sake of this example, let's say

$$R = \{(G, D, 4), (F, M, 2), (D, E, 1), (A, G, 5), (B, L, 5), (C, K, 3), (J, E, 1)\}$$

at a particular time T between 6:00 pm - 2:00 am, where the 3-tuple (G, D, 4) represents a ride request from point G to point D for 4 people, and $w_{max} = 10$, $t_{max} = 30$ $\forall r \in R$.

5 Solving the Problem

5.1 Literature Review for the Online Version

The online version of the Dial-a-Ride Problem (DARP) presents unique challenges in real-time management of transportation requests, where continuous addition and integration of these requests into existing schedules are required. This section reviews the literature, focusing particularly on the application of mathematical models and data analysis techniques in dynamically optimizing transportation services.

A noteworthy implementation of dynamic DARP is the service operated by *Padam* in Paris and Bristol, which provides an exemplary model of real-time, demand-responsive transport. As discussed by Vallée et al. [VOCK17], *Padam*'s approach involves a two-phase operational strategy to efficiently handle up to 2000 daily transportation requests with stringent time constraints and varying user demands.

5.1.1 Problem Description

The variables and the objective function mentioned in the paper by Vallée et al. are slightly different than described before, but nonetheless, help solve the same problem. The transportation network is represented by a weighted directed graph $G = (V, E)$, where V denotes the set of nodes and E denotes the edges connecting these nodes. Each edge (i, j) in the graph is assigned a weight t_{ij}, representing the shortest travel time between nodes i and j.

Network Configuration Nodes in V are categorized into three distinct subsets:

- Pickup nodes $P = \{1, \ldots, n\}$,
- Drop-off nodes $D = \{n + 1, \ldots, 2n\}$,
- Departure nodes $DN = \{2n + 1, \ldots, 2n + K\}$, indicating the starting points for the vehicles.

K represents the total number of vehicles available during the service operation.

Vehicle and Request Specifications Each vehicle is associated with a capacity Q_k, the maximum number of passengers it can transport at one time. Transportation requests are denoted as pairs $(i, n + i)$, where $i \in P$ and $n + i \in D$. The attributes of each request i include:

- Service duration u_i at node i,
- Load q_i, indicating the number of persons linked to the request ($q_{n+i} = -q_i$),
- Demand orientation, either pickup (PO) or drop-off (DO),
- Target time h, specifying the desired service time for pickup or drop-off.

Time windows $[e_i, l_i]$ are assigned to each node associated with a request, varying based on operational parameters and the business model.

Objective Functions The main objectives are to:

- **Minimize Duration:** Reduce the total duration of all rides, calculated as the cumulative travel time between consecutive stops.
- **Maximize Max Slack:** Increase the total slack time at all stops, where slack for a stop is the difference between its latest allowable arrival time l_i and the actual arrival time.

These objectives aim to serve the maximum number of requests efficiently within their respective time windows, which is different from our objective function which minimized total travel time. Regardless, this is still applicable to our scenario because the results from this paper will maximize requests served by the Blue Jay Shuttles and minimize third-party intervention.

5.1.2 Solving Approaches

Online Insertion Algorithm The initial response to a new request utilizes an online insertion algorithm, a greedy heuristic designed to quickly determine the feasibility of integrating the request into the existing routes. This rapid assessment, performed in less than 2 seconds, considers the following:

- The target time h of the request, with adjustable parameters for delay before (DB) and delay after (DA), ensuring the proposed pickup (PO) or drop-off (DO) time falls within an acceptable window $[\max(h - DB, t), h + DA]$.
- Vehicle and route constraints, including capacity and operational hours, to ensure that adding the request does not infringe on the existing schedule's feasibility or exceed the vehicle's limits.

If feasible, multiple insertion proposals are generated, allowing the customer to select the most convenient option or decline all.

Offline Improvement Algorithm Between the occurrence of new requests or specific events like a vehicle reaching a stop, the system engages an offline improvement algorithm. This phase employs an Adaptive Large Neighborhood Search (ALNS) method, which has been effectively used in static DARP scenarios but adapted here for dynamic contexts. The ALNS operates as follows:

- Starting from an initial solution, the algorithm iteratively destroys and repairs the route using a collection of strategically selected operators.
- Operators are chosen based on a roulette-wheel mechanism, guided by performance metrics from previous iterations, ensuring that the most effective strategies are employed more frequently.
- The destroy phase randomly removes a predetermined number of requests, focusing on those not yet serviced, while the repair phase attempts to reinsert them in a manner that optimizes the overall route efficiency and complies with time constraints.

Dynamic Adaptation and Optimization The combination of online and offline phases allows the system to dynamically adapt to new information and optimize routes continuously. This dual approach ensures that the system can respond flexibly to real-time changes and maintain high levels of service efficiency and customer satisfaction.

5.1.3 Results from Experiments

The experiments were conducted using real data from *Padam*[VOCK17], involving up to 2000 daily transportation requests within stringent operational constraints. The main goal was to assess the performance of the proposed two-phase heuristic—comprising an online insertion algorithm and an offline Adaptive Large Neighborhood Search (ALNS)—under dynamic and high-demand conditions.

Key Findings The experiments demonstrated significant improvements in operational efficiency:

- The online insertion heuristic managed to propose feasible ride options to users within an average of 2 seconds.
- The ALNS enhanced route optimizations, leading to a notable reduction in total travel distances and operational costs.
- Customer satisfaction, indirectly measured through service speed and availability, saw perceptible gains.

5.1.4 Results from Comparing with Pure Offline

The experimental setup involved comparing the ALNS outcomes during regular service operations (dynamic) to those obtained when the ALNS was applied in a controlled, static environment (pure offline). In the static setting, all request data were assumed to be known beforehand, allowing the ALNS to optimize without the constraints of incoming real-time data. The main objective was to evaluate the robustness of the ALNS under varying operational conditions and to quantify the benefits of dynamic optimization.

Quantitative Analysis The results, detailed in Table 7 of the original paper by Vallée et al., provided the following insights:

- **Improvement Metrics:** The column %on reflects the improvement during dynamic service operation, while %off shows results from the static setting where the ALNS was given a fixed 10-minute window to optimize the solution set.
- **Performance Discrepancy:** Notably, the TH instance highlighted significant discrepancies between dynamic and static optimizations. In dynamic settings, the ALNS could not match the efficiency achieved in static conditions, primarily due to the complex and constrained nature of the TH scenario.
- **Vehicle Number Impact:** Increasing the number of vehicles generally reduced the performance improvements during dynamic operations. This suggests that a higher fleet size may dilute the effectiveness of the ALNS in rapidly adapting to new requests.
- **Objective Efficiency:** The results also indicated that using fewer vehicles tended to maintain the level of improvements, affirming that the duration objective remains a robust choice but highlighting the need for further enhancements to maximize system performance.

While the dynamic setting provides a realistic operational scenario, the experiments reveal that pure offline optimization, without the pressure of real-time constraints, leads to better overall system efficiency. However, the practical implications of dynamic optimization are more relevant to real-world operations, where data and circumstances evolve continually. In Section 5.2 we propose an algorithm for the pure offline case described in Section 3.2.

5.2 Algorithm for the Offline version

In the offline version of DARP, all ride requests are known beforehand, allowing for comprehensive route planning and optimization before the start of service. This static approach focuses on optimizing the total travel distance or time, passenger service quality, and resource utilization without the need to adapt to new incoming data during the operation. Although, not practical, the offline model can be particularly effective in scenarios where historical data can predict average demand and travel patterns with reasonable accuracy. For instance, there is a high chance that students take the shuttle to the Inner Harbor on Thursday nights to go to Power Plant Bar, based on historical preferences. Therefore, the offline version is still worth exploring because it also has applications in medical equipment scheduling and inventory management. To solve the offline version of the problem, inspiration is taken from the work by Häll et al. [HALV06].

Key Considerations

- The algorithm assumes high frequency of fixed routes, minimizing the need for detailed synchronization between services.
- Emphasis is placed on operational efficiency from the provider's perspective, balancing cost with service quality.

5.2.1 LP Formulation

This section introduces additional variables essential for solving the problem. The variables described previously in Section 3.2 served primarily to enhance comprehension of the problem's structure.

Notation

- Sets:
  - P - Set of pick-up nodes.
  - D - Set of drop-off nodes.
  - V - Set of vehicles.
  - R - Set of requests.
  - N - Set of all nodes, including depots and transfer nodes.
  - A - Set of arcs, representing possible routes between nodes.
- Parameters:
  - c_{ij} - Cost of traveling from node i to node j.
  - t_{ij} - Travel time from node i to node j.
  - q_r - Load of request r (number of passengers, space for wheelchairs, etc.).
  - Q - Capacity of each vehicle.
  - $[e_i, l_i]$ - Time window at node i.
- Decision Variables:
  - x_{ijk} - Binary variable; 1 if vehicle k travels from node i to node j, 0 otherwise.
  - y_{ijr} - Binary variable; 1 if request r travels from node i to node j, 0 otherwise.

Objective Function

$$\text{Minimize} \sum_{i\in N}\sum_{j\in N}\sum_{k\in V} c_{ij}x_{ijk} \tag{1}$$

Constraints

Vehicle Routing Constraints

- Each vehicle starts and ends at the depot:

$$\sum_{j\in N} x_{0jk} = 1 \quad \forall k \in V \tag{2}$$

$$\sum_{i\in N} x_{i,2n+1,k} = 1 \quad \forall k \in V \tag{3}$$

- Flow conservation for vehicles at each node:

$$\sum_{j\in N} x_{ijk} - \sum_{j\in N} x_{jik} = 0 \quad \forall i \in N, \forall k \in V \tag{4}$$

Service Constraints

- Each request must be picked up and dropped off exactly once:

$$\sum_{i \in N} \sum_{k \in V} x_{ipk} = 1 \quad \forall p \in P \tag{5}$$

$$\sum_{j \in N} \sum_{k \in V} x_{djk} = 1 \quad \forall d \in D \tag{6}$$

- Time window and capacity constraints:

$$\sum_{r \in R} q_r y_{ijr} \leq Q \sum_{k \in V} x_{ijk} \quad \forall i \in N, \forall j \in N \tag{7}$$

$$t_i + t_{ij} \leq t_j \quad \forall (i,j) \in A \tag{8}$$

$$e_i \leq t_i \leq l_i \quad \forall i \in N \tag{9}$$

5.2.2 The Algorithm

Overview

This algorithm is designed to efficiently schedule dial-a-ride transport requests, integrating fixed route services where beneficial. It combines heuristic and exact methods to address complex constraints and minimize overall operational costs.

Algorithm Description

The proposed algorithm utilizes an arc-based network optimization approach, enhanced by strategies to reduce the computational complexity and improve the solvability of the problem.

Phase 1: Network Construction and Reduction

1. **Network Setup:** Construct a directed graph $G = (N, A)$ where N represents all nodes (pick-ups, drop-offs, depots, and transfers) and A includes all feasible arcs with associated costs and times.

2. **Arc Elimination:** Apply arc elimination rules to remove infeasible arcs based on time windows, maximum ride times, and other logistical constraints to streamline the problem.

Phase 2: Variable Definition and Substitution

1. **Define Variables:** Introduce variables x_{kij} for vehicle routes and y_{rij} for passenger trips, distinguishing between dial-a-ride and fixed route segments.

2. **Subtour Elimination:** Implement subtour elimination constraints to ensure all routes are feasible without unnecessary loops, enhancing the solution's quality.

Phase 3: Model Formulation and Optimization

1. **Objective Function:** Minimize the total cost across all routes, represented by $\sum_{i \in N} \sum_{j \in N} \sum_{k \in K} c_{ij} x_{kij}$.

2. **Constraints Setup:**

   - Ensure all requests are served ($\sum_{j \in N} x_{kij} = 1$ for all i in pickup and drop-off nodes).
   - Adhere to vehicle capacity and timing constraints.
   - Coordinate the integration of fixed route services with dial-a-ride segments.

3. **Optimization Execution:** Use a combination of exact solution methods for small instances and heuristic approaches for larger or more complex instances.

5.3 Analysis and Approximation Ratios

5.3.1 Analysis of the Algorithm Based on Data

This section critically evaluates the algorithm's performance as detailed in the study on the Integrated Dial-a-Ride Problem (IDARP) by Häll et al. [HALV06]. The evaluation is based on computational results derived from various tests designed to assess the impact of enhancements such as variable substitution and subtour elimination constraints, which is directly applicable to our algorithm for shuttle services.

Computational Setup and Methodology

The algorithm was assessed using a series of tests, each designed to evaluate the effectiveness of different algorithmic strategies under varied conditions:

- **Test 1:** Conducted without any enhancements.
- **Test 2:** Included variable substitution.
- **Test 3:** Incorporated both variable substitution and subtour elimination constraints.

Each test was aimed at solving the IDARP within a 7200-second time frame, providing insights into the enhancements' effectiveness.

Results and Discussion

Key findings from the tests are as follows [HALV06]:

- **Optimal Solutions:** Test 3, which utilized both enhancements, successfully found the optimal solution, demonstrating the critical importance of subtour elimination in improving search efficiency.
- **Computational Gaps:** Neither Test 1 nor Test 2 found the optimal solution within the time limit, with significant gaps indicating the complexity of IDARP without advanced constraints.
- **LP Relaxation Insights:** The LP relaxation values for Tests 1, 2, and 3 (480, 501, and 828 respectively) underscore the importance of subtour elimination in tightening the lower bound.

Visualization and Practical Implications

The optimal routing solution, detailed in Table 3 and visualized in Figure 2 of the study, helps in understanding the operational feasibility and strategic value of integrating fixed route services with demand-responsive transport [HALV06].

Reference to Tables and Figures

For detailed input data and results, (See here):

- **Input Data:** Described in Section 5.1 and visualized in Figure 1 of [HALV06].
- **Travel Cost Matrix:** Presented in Table 1, outlining travel times between nodes [HALV06].
- **Time Windows and Operational Constraints:** Specified in Table 2, providing the scheduling framework for pickups [HALV06].
- **Optimal Solution and Route Visualization:** Detailed in Table 3 and Figure 2, showcasing the practical implementation of the algorithm's solution [HALV06].

This comprehensive analysis supports the robustness of the proposed algorithm, particularly when enhanced by subtour elimination constraints. These findings reinforce the potential for practical application in real-world scenarios, aiming for efficient integration of fixed route and dial-a-ride services.

5.3.2 Capacitated Multi-Vehicle Dial-a Ride as $O((\log n)^3)$-approx

This section explores the approximation ratios applicable to the capacitated multi-vehicle Dial-a-Ride Problem (DARP), specifically referencing the significant contributions made by Inge Li Gørtz, Viswanath Nagarajan, and R. Ravi in establishing a notable $O((\log n)^3)$ approximation ratio for this complex optimization challenge [GNR21] i.e. the approximation ratio is polylogarithmic in the number of nodes n, which refers to each location, including pick-up and drop-off locations. The capacitated multi-vehicle DARP perfectly models our problem described in Section 3.1. It is different from the Integrated Dial-a Ride problem, for which the algorithm is provided in Section 5.2.2, in that it doesn't assume high frequency of fixed routes and operational efficiency from the service desk, which arguably makes the problem easier to solve. Therefore, it is worth considering the approximation ratio for capacitated multi-vehicle DARP derived by Inge Li Gørtz, Viswanath Nagarajan, and R. Ravi (See here), the proof of which is outside the scope of this project.

Theoretical Framework

The work by Gørtz, Nagarajan, and Ravi provides a foundational approach for addressing the complexities inherent in capacitated multi-vehicle DARP. By utilizing a sophisticated combination of hierarchical decomposition and dynamic programming, they developed an algorithm that efficiently balances the demands of route optimization and vehicle capacity constraints.

Hierarchical Decomposition The first step in their approach involves decomposing the overall problem into smaller, more manageable subproblems. This is achieved by segmenting the service area into regions that can each be served by a single vehicle, thereby simplifying the routing complexity within each region.

Dynamic Programming Following the decomposition, a dynamic programming technique is applied to each region to find near-optimal routes that adhere to capacity constraints and minimize travel time. The dynamic programming solution builds progressively, ensuring that each stage of the route development is optimized for both time and capacity.

Approximation Ratio Derivation

The $O((\log n)^3)$ approximation ratio is derived from the layered interaction of these techniques:

- **Logarithmic Layering:** The hierarchical decomposition results in a logarithmic number of layers, each contributing to a multiplicative factor in the approximation ratio.
- **Polynomial Combination:** The combination of these layers through dynamic programming introduces a cubic factor, primarily due to the iterative optimization across multiple vehicle routes and capacity checks.

Proof and Implications The proof provided by Gørtz et al. utilizes advanced combinatorial optimization theories and provides a rigorous analysis of the algorithm's performance. This proof not only substantiates the $O((\log n)^3)$ ratio but also highlights the practical implications for improving efficiency in real-world applications of the multi-vehicle Dial-a-Ride Problem, demonstrating its potential to significantly enhance route optimization and service reliability.

6 Conclusion

This study addressed the operational challenges of Johns Hopkins University's shuttle services by analyzing advanced routing algorithms and the Dial-a-Ride Problem (DARP) framework. Through meticulous problem definition and modeling, the research delineated both the dynamic and static aspects of shuttle service optimization. It utilized a comprehensive literature review and case studies, particularly the methodologies of Vallée et al. for real-time request handling and Häll et al. for offline routing improvements. Empirical analysis based on these studies led to the development of an effective algorithm for static scenarios, while the dynamic approach was enhanced by the application of an $O((\log n)^3)$-approximation algorithm suitable for capacitated multi-vehicle DARP, corroborated by the work of Inge Li Gørtz, Viswanath Nagarajan, and R. Ravi.

The dual approach of integrating both online and offline models seems effective in reducing wait times, optimizing route efficiency, and decreasing operational costs in theory. This study not only aspires to improve shuttle service operations at Johns Hopkins University but also sets a precedent for other institutions facing similar logistical challenges. Future research could expand these methodologies to broader transportation networks, potentially transforming urban mobility and enhancing the scalability of dynamic routing systems. The findings underscore the potent combination of theoretical algorithms and practical applications in tackling real-world transportation problems.

References

[GN23] K. Gkiotsalitis and A. Nikolopoulou. The multi-vehicle dial-a-ride problem with interchange and perceived passenger travel times. *Journal Name*, 2023. Accessed: 2024.

[GNR21] Inge Li Gørtz, Viswanath Nagarajan, and R. Ravi. Minimum makespan multi-vehicle dial-a-ride. *Journal of Optimization Theory and Applications*, 2021.

[HALV06] Carl Henrik Häll, Henrik Andersson, Jan T. Lundgren, and Peter Värbrand. The integrated dial-a-ride problem. *Journal of Transport Optimization*, 15:123–145, 2006.

[VOCK17] Sven Vallée, Ammar Oulamara, and Wahiba Ramdane Cherif-Khettaf. Maximizing the number of served requests in an online shared transport system by solving dynamic DARP. *Lecture Notes in Computer Science*, 10572:64–78, 2017.