Multi-Objective Bus System Design and Optimization
A Case Study on Transit from Manhattan to LaGuardia Airport

Deven Barth

Final Report

Abstract. Public transit from demand-heavy regions of Manhattan (Midtown, Lower Manhattan) to LaGuardia Airport is fragmented and heavily dependent on last-mile bus trips to ensure journey completeness. This report aims to investigate the current state of a passenger's journey through the transit system and proposes optimization-backed solutions in the form of direct bus routing from selected Manhattan locations to LaGuardia Airport. Specifically, multi-objective optimization techniques are used to maximize coverage while also minimizing system cost in building these proposed bus systems. Each system is analyzed with the goal of ensuring a fair trade-off between capturing ridership demand and time spent in the system. The proposed direct bus-links offer significant improvements in system efficiency from a passenger's perspective (time/accessibility), while also successfully obtaining the existing demand.

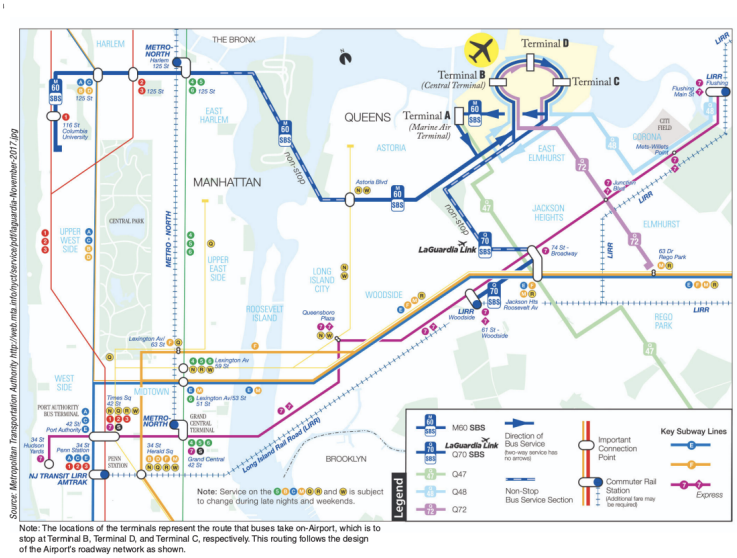

Fig. 1: [Source: Metropolitan Transportation Authority] Current Map of Connections Linking LGA to areas throughout NYC

1 Motivation

1.1 Current State

The Greater New York City Area boasts by far the largest share of transit users as a proportion compared to other MSAs throughout the United States, yet despite this, accessibility to its major airports (LaGuardia, JFK, and Newark) are relatively disjointed with respect to the scale of built-up transit coverage. Singling out LaGuardia Airport in particular, no direct rail or bus options exist from some of the highest density regions of the New York City area, resulting in a majority of airport travelers opting for personal transport such as Taxis and Ubers/Lyfts in order to fill the efficiency gap, despite spending significantly more in personal travel costs.

Table 1-1
Ground Access Mode Choice at LGA

Mode	Air Passengers	Airport Employees
Auto park / short term	5.6%	55.7%
Auto park / long term	1.0%	0.0%
Auto park / off-airport	1.5%	0.0%
Auto passenger / drop-off or pick-up	20.0%	1.6%
Taxi/limousine/Uber/Lyft	51.2%	1.3%
Bus, subway, LIRR	6.2%	40.1%
Van/shuttle/hotel courtesy	5.6%	0.0%
Rental car on-airport and off-airport	7.8%	0.0%
Other modes	1.1%	1.3%
Total	**100.0%**	**100.0%**
Sources: 2017 LGA Ground Access Survey and 2014–2016 LGA Customer Satisfaction Surveys.		

Fig. 2: [Source: Port Authority of NY and NJ] Survey of Transportation Mode Choice across LGA Customers

A few bus rapid-transit-adjacent services (branded as "Select Bus Services" by the MTA) connect LaGuardia Airport to regions throughout NYC proper, specifically the M60+ which services Upper Manhattan and Astoria, Queens, and the Q70+, a free shuttle service that operates between LaGuardia Airport and subway/LIRR hubs in Woodside and Jackson Heights, Queens. Transit demand from the Manhattan CBD regions (areas south of 60th Street, akin to the future congestion pricing zone approved recently) is often funneled through the Q70+ route in particular, provided that the route has adequate rail connectivity to many points throughout the city (the E,M,R,7 trains at Jackson Heights, the 7 and LIRR at Woodside). However, from an accessibility standpoint, many riders have to deal with the transfer at these respective hubs in order to complete their journey, on top of waiting for the Q70+ service, which is prone to operational bottlenecks during rush hour. This is a sub-optimal setup considering the time-sensitive nature of catching a flight for a majority of passengers making the trip to LGA Airport.

1.2 Proposals

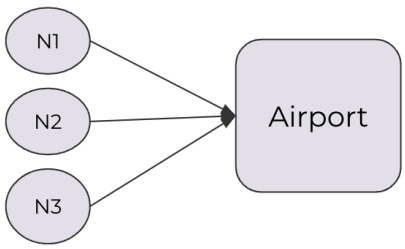

Fig. 3: Illustration of Alternative Scenario A

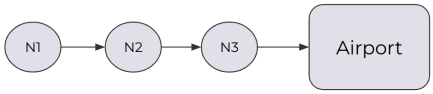

Fig. 4: Illustration of Alternative Scenario B

Three core systems are studied in this report:

- **The Benchmark:** The current passenger journey including the transfer at Jackson Heights and the Q70+ service.
- **Alternative Scenario A:** Multiple parallel bi-nodal routes from select Manhattan locations to LGA.
- **Alternative Scenario B:** A single sequential-based multi-stop route in Manhattan terminating at LGA. More of a traditional bus route.

Currently, no direct bus-links exist between the Manhattan CBD regions and LGA Airport. The closest analogue is a recently launched shuttle service operated by Uber, which has three pre-defined locations for passenger pick-up and offers direct service to LGA Airport. However, from a consumer standpoint it has more in common with a taxi/ride-sharing service than a true bus service, as its operational model relies on ride-hailing, as opposed to pre-determined headways. Nonetheless, this is an example of a model that could be implemented in the form of a public good, as it is vaguely similar to the setup in Scenario A.

Two core metrics are used to evaluate performance across the two proposed systems: **total user time in system** and **proportion of demand covered**. These two performance metrics illustrate the "multi-objective" nature of the problem, as the goal is to simultaneously maximize system coverage/utilization while also minimizing costs accumulated in the system (time spent as a passenger in transit). We are also interested in validating the hypothesis that a direct-link as outlined in the proposals is at the bare minimum an improvement cost-wise with respect to the current passenger transit flow, which is outlined as a "benchmark system".

2 The Benchmark

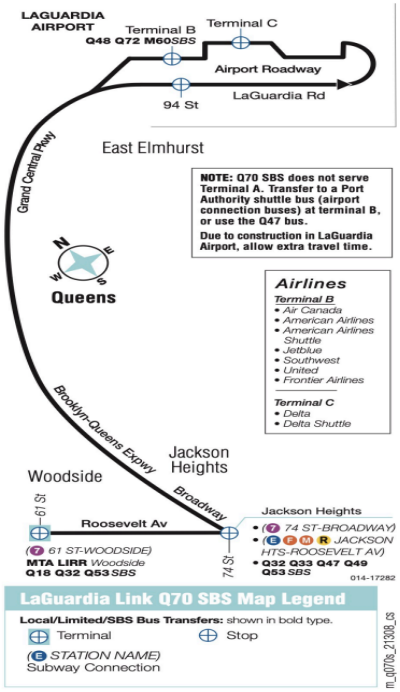

Fig. 5: [Source: Metropolitan Transportation Authority] Q70+ Route

For the Benchmark System, we consider the most typical transit journey of an individual traveling from Manhattan to LGA Airport:

- Subway from Manhattan (assume south of 60th St.) to Jackson Heights. Assume **20-40 minutes** according to standard Google Maps routing.
- Transferring at Jackson Heights (74th/Broadway) and waiting for the Q70+ SBS. We are interested in estimating passenger wait-times based on available empirical data.
- The Q70+ ride-time itself. According to the schedule set forth by the MTA, ride-time from 74th/Broadway to all terminals is **11 minutes**. Negate added congestion effects for the sake of this analysis.

No hard optimization is needed to parameterize and model the benchmark, as it follows deterministic routing conditions. Instead, the key goal is to estimate areas of uncertainty pertaining to the passenger journey itself, in which the main focus is to identify the added cost (in this case, user time in system) of a required transfer in a system with respect to proposals made to eliminate this step. To do so, estimate this by leveraging existing data pertaining to the Q70+ service.

2.1 Data Collection

Two key components were considered:

- Bus location/schedule-performance with respect to the next called upon stop (in this case, buses arriving into 74th/Broadway). Obtained via the MTA Bus-time API.
- Ridership demand, through the NYC Open Data Portal API (MTA Bus Hourly Ridership)

Finding reliable ridership data on the Q70 service was difficult; since it's a free service, OMNY/Metrocard taps were no longer recorded as of April 2022, so demand data was obtained at an aggregate level between January 2022 to April 2022, through the NYC Open Data Portal API. This dataset is considered when modeling demand and system utilization.

Location-based data was obtained through the MTA Bus-time API, which serves as a wrapper on top of the MTA's GTFS data serialization. This data is refreshed in real-time, as positioning data is conditional on the current time of service and well as overall bus route status. In obtaining this dataset into something usable for analysis, a Python script was automated on a minute-based cadence to obtain the records of all buses active on the Q70+ route over a 48 hour period. These records were pre-processed in order to obtain the scheduling deltas for each active bus at a given sync, as well as distance to the next stop, with the end result pushed to a local SQLite database for reference when performing a cross analysis on the ridership dataset.

	VehicleID	RecordTime	Longitude	Latitude	StopPointName	ArrivalDelta
0	MTABC_6276	2024-11-25 23:54:47-05:00	-73.899794	40.756002	ROOSEVELT AV/75 ST	9.322750
1	MTABC_6274	2024-11-25 23:54:35-05:00	-73.894239	40.765666	LAGUARDIA RD/94 ST	-0.203917
2	MTABC_6274	2024-11-25 23:57:41-05:00	-73.876018	40.771637	LAGUARDIA RD/94 ST	-0.124600
3	MTABC_6276	2024-11-25 23:57:51-05:00	-73.892445	40.747161	ROOSEVELT AV/75 ST	9.211417
4	MTABC_6276	2024-11-26 00:06:05-05:00	-73.901208	40.744209	WOODSIDE AV/61 ST	7.217850

Fig. 6: Snapshot of (cleaned) real-time bus location data

Figure 6 illustrates a sample of bus location pings off of the Q70 service. Most relevant is the 'ArrivalDelta' field, which measures the difference between the "aimed" (scheduled/expected) bus arrival and the projected arrival based on current conditions. Sections 2.2 and 2.3 below touches on the distribution of this field, and how it's used to parameterize transfer wait-time in the system, in conjunction with the current Q70+ schedule.

2.2 Additional Parametrization/Assumptions

According to the Q70+ schedule, there are two service patterns, an AM block which consists of 10-20 minute headways (dependent on hour) and a PM block, which runs more frequently with 8 minute headways. Combining these assumptions with the ArrivalDelta denoted in Figure 6 allows for an understanding on an upper and lower bound in terms of passenger wait time. The best case scenario is relatively straightforward; a passenger disembarks the subway at 74th Broadway and catches the Q70+ right as it pulls into the terminal, which is assumed to take around **2 minutes** to walk from the subway fare-zone to the bus bay. The worst case scenario, however, considers the expected wait-time under the headway schedule on-top of delays incurred

by bottlenecks in the system. Because the headways are defined hourly according to the schedule, the delays (the ArrivalDelta variable) are grouped/aggregated over all active vehicles (by taking a mean) to the matched hour. Hence, the worst case scenario is defined as $headway \ + \ max[0, -ArrivalDelta]$. The wait-time approximations per hour are parameterized as the following:

$$wait(hr) \in [2, headway(hr) \ + \ max[0, -ArrivalDelta(hr)]$$

hour	max_wait_approximation	headway_gap	expected_bus_arrivals (mu_exp)	actual_bus_arrivals (mu)
0	25.072357	20.0	3.0	2.393074
1	21.471162	20.0	3.0	2.794446
2	20.488450	20.0	3.0	2.928479
4	20.893600	20.0	3.0	2.871693
5	20.254562	20.0	3.0	2.962296
6	12.602926	10.0	6.0	4.760799
7	11.043188	10.0	6.0	5.433214
8	12.022996	10.0	6.0	4.990437
9	20.938081	10.0	6.0	2.865592
10	10.695248	10.0	6.0	5.609968
11	10.125225	10.0	6.0	5.925794
12	10.000000	10.0	6.0	6.000000
14	8.416652	8.0	7.5	7.128726
15	8.877671	8.0	7.5	6.758530
17	14.292386	8.0	7.5	4.198039
18	23.183346	8.0	7.5	2.588065
19	24.124424	8.0	7.5	2.487106
20	12.959124	8.0	7.5	4.629943
21	8.192383	8.0	7.5	7.323877
22	8.033630	8.0	7.5	7.468604
23	9.025817	8.0	7.5	6.647598

Fig. 7: Max. Wait-Time Estimates: 74th/Broadway Transfer for the Q70+ Service

Figure 7 illustrates the max passenger-transfer wait-time estimates for the Q70+ service, in addition to the expected and ground truth service rates (in terms of bus arrivals per hour). Note that the bus arrivals aren't integers due to 60 (minutes) not being perfectly divisible by some of the headways and ground truth arrival deltas; however, for consistency sake, we'll keep our analysis at hour level. When assigning headways to the alternate systems, this is addressed via ceiling division, which is touched on in section 3.5.

2.3 Findings and Discussion: Passenger Wait-Times

It turns out that based on the measured 48 hour period, around 60% of bus pings into 74th/Broadway were deemed late, defined as having $ArrivalDelta < 0$, and this is further exacerbated by bottlenecks occurring during the PM rush-hour, on-top of the frequent operational headways during that time. In fact, as demonstrated in Figure 8 below, all buses during the PM rush hours (17-20) were deemed late. Do note that there could be an element of bias as this was only captured over 2 days versus this being a global average, and what happened on both days were potentially anomalous. However, it is sufficient for the sake of modeling a "worst case scenario", especially as the objective is to demonstrate the impact a required transfer has on overall passenger time-in-system.

Based on the analysis time-slot, it appears that rush-hour-associated delays had a heavy impact on passenger wait-time, as well as the PM/AM headway shift at midnight, which is potentially attributable to internal/operational factors (versus traffic congestion). Next, considering the entire system from start-to-finish by generalizing the approximate cost incurred provided the definitions stated above, the best case scenario would be a 20 minute subway ride (the closest point in Manhattan with respect to Jackson Heights), a no-wait transfer (2 minutes), and the time in-ride on the Q70+ service (11 minutes), which equates to **33 minutes**. On the other hand, a worst case scenario would be a 40 minute subway ride (the furthest point in the Manhattan CBD), a max-wait of 25 minutes (plus 2 minutes to walk) at 74th/Broadway, and 11 minutes on the Q70+, yielding an upper bound of **78 minutes**. A key consideration is to ensure that the alternative systems abide by these general targets, by maintaining a system cost below the benchmark upper-bound at minimum.

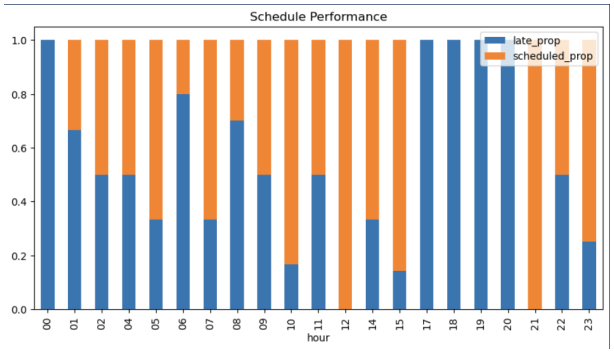

Fig. 8: Proportions of late/scheduled arrivals into 74th/Broadway per hour via the 48 hour GTFS Data Dump

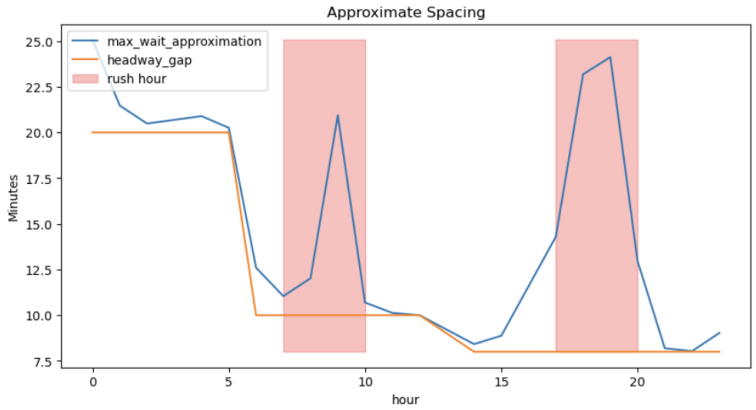

Fig. 9: Headways (Expected Wait) vs Max Wait-times per hour. Note the peaks during the rush hour blocks.

2.4 Additional Discussion: System Utilization

Now, consider combining the wait-time parametrization with system demand. Applying the MTA Bus Hourly Ridership dataset to parse out average hourly ridership on the Q70+ service (let's define this as $\lambda_i, \forall i \in \{1, .., 24\}$), assume passengers arrive into 74th/Broadway according to a memoryless process at rate λ_i, dependent on the time of day. Let the service (bus arrival) rates be defined as μ_i, which is described above in Figure 7. While this scenario has M/M/1-like properties (instead of a single individual to a server, think a collection of individuals mapped to a single bus), it is a bit too simplistic of a model to fully capture all dynamics of this system, as it is in reality a network of inter-dependent queues per system state across different levels of granularity (the "servers", buses encountering traffic, and the "passengers" waiting for the bus). What could be inferred from these two processes (passenger arrivals and bus arrivals) is an overall utilization ratio of passengers mapped to buses ($\phi_i = \lambda_i/\mu_i$), which could be a good proxy for determining capacity limitations on the system. Assume a standard articulated bus can seat 48 passengers. As a result of delays in the system during the PM rush-hour, Figure 10 below illustrates the strain on utilization to a hard capacity limit such as this.

3 Alternative Systems

Two alternative systems are proposed, as denoted and described in section 1.2. Specifically with the goal of building bus routes directly linking Manhattan to LaGuardia Airport, the resulting system designs (stop placements, specifically) are conceptualized via multi-objective, mixed integer programs. The primary objective is to:

- **(Scenario A)** Find pick-up locations throughout Manhattan that maximize demand coverage while minimizing system-wide time-based costs.
- **(Scenario B)** Find stop placements and a resulting route sequence that maximizes demand coverage along a route while minimizing system-wide time-based costs.

Figure 11 illustrates the scope of the optimization space in building these routes, in terms of candidate pick-up locations/bus stops and coverage points. Each red "node" represents a geographical demand as it pertains to transit flows to LaGuardia

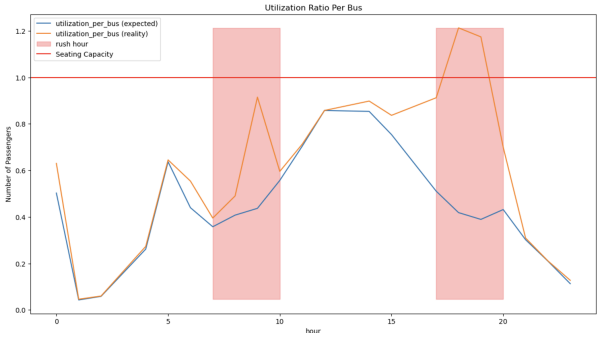

Fig. 10: Q70+ Utilization Ratio by Hour

Fig. 11: Total Study Space (Red: coverage/facility nodes scaled in size with respect to demand. Blue: the current Q70+ route)

Airport. These demands are approximated through a combination of Subway OD-pair ridership data and assumptions pertaining to the Benchmark system (the Q70+ service), which is further described in section 3.1. The nodes are representative of present day NYC Subway stops. Costs (time and distance) between each node are calculated through osmnx, a Python package that provides functionality for robust network analysis on top of Open Street Map.

3.1 Data Collection and Parameters

Three core deliverables are required in order to solidify our study space for optimization: h, a vector-like quantity of demands per node, D, a pair-wise matrix containing distances between each node pair, and T, another pair-wise matrix describing the travel-times between each node pair.

Obtaining Demand (h):

Given that the benchmark specifically assumes the first step of a passenger's journey is to enter the Subway system, each node (or rather, coverage point/facility location) corresponds to a Manhattan Subway stop, which is assumed to be a decent proxy for approximating locational demand. Specifically, the average daily OD-pair ridership (using the NYC Open Data Portal) between each node to Jackson Heights/Woodside is used in conjunction with the total (Manhattan-origin) ridership on the Q70+ service, in order determine the share of Q70+ riders at different points of the Manhattan CBD regions. In this set-up, Manhattan-origin ridership proportion on the Q70+ service is assumed to be equivalent to the proportion of all LGA customers regardless of transit mode, which is cited at 44% according to a 2017 survey conducted by the Port Authority of NY/NJ. Note that this assumes Q70+ demand per Manhattan stop is directly proportional to the share of riders going to Jackson Heights/-Woodside, which is a potentially reductive assumption in a realistic scenario. Nonetheless, due to limitations in existing data at the granularity of each stop, this is a decent enough estimation. The calculations for each $h_i, \forall i \in nodes$ are further detailed in the list of parameters at the end of this section.

Obtaining Costs (D, T):

Next, given our set of candidate nodes in Manhattan as well as the destination node (LGA Airport), we are interested in enumerating all possible OD-pair costs within this space (both $d_{ij} \in D$ and $t_{ij} \in T, \forall (i,j) \in$ *node pairs*). Doing so required some geo-spatial analysis via osmnx; all nodes are identified and tagged based on their point-based location (latitude/longitude) on the NYC streetmap template obtained via Open Street Map/osmnx functionality, by taking the closest street-side intersection corresponding to the node-based lat/lon pair. Once each node is identified (by a numerical string-based value), a "shortest path" is calculated between each possible pair of nodes, which is said to use a variant of Dijkstra's algorithm in the osmnx/networkx backend according to documentation. Note that these shortest-path computations take into account all street-side intersections included in the NYC base-map, not only the tagged nodes in the study space for this problem. The code snippet below illustrates the process of:

- (1) Retrieving the base map
- (2) Obtaining edge-based speed/travel-time layers on top of the base map
- (3) Tagging the study-space nodes for the problem
- (4) Calculating the (distance and time-based) shortest paths between each node-pairing specified in (3)

```
import osmnx as ox

G = ox.graph_from_place('New York, New York', network_type='drive')

G = ox.add_edge_speeds(G)
G = ox.add_edge_travel_times(G)
ox.save_graphml(G, filepath="inputs/nyc.graphml")

lats = subway_od_pair_estimates['lat'].astype(float)
longs = subway_od_pair_estimates['long'].astype(float)

stops = ox.nearest_nodes(G, longs, lats)
subway_od_pair_estimates['osmnx_node_id'] = stops

def calculate_costs(G, lst, i, weight_type='length'):
    '''
    Inputs:
    **G: an osmnx graph object (the NYC streetmap)
    **lst: the list of all study nodes
    **i: the reference node to calculate costs with respect to lst
    **weight_type: flag what type of cost we're calculating (length/distance vs travel time)
    '''

    temp = []
    #apply dijkstra's algo to every node pair...
    for j in lst:
        r = [k for k in ox.shortest_path(G, i, j, weight=weight_type)]
        if len(r) == 1: l = 0
        else:
            d=ox.routing.route_to_gdf(G, r)
            l = d[weight_type].sum()
        temp.append(l)
    return temp

def create_matrix(G, stops, weight_type):
    '''
    Builds a matrix of costs per node-pairing
    '''
```

```
    m = [calculate_costs(G, stops, i, weight_type=weight_type) for i in stops]
    m_df = pd.DataFrame(m, columns=stops, index=stops)
    return m_df

distances = create_matrix(G, stops, weight_type = 'length', file = 'distances')
times = create_matrix(G, stops, weight_type = 'travel_time', file = 'times')
```

With 66 total study nodes in the problem, the results are two matrices of size 66x66 containing link-wise distances (meters) and times (seconds). Figure 12 below is a snapshot of T, our time-based representation of the study space. Note that because osmnx isn't a traffic simulation tool, many of these values are likely to be underestimates as they're referencing edge-based speed limits only.

	42439972	9140654137	42430304	42430329	42430352	42449961
42439972	0.0	104.61876874349277	74.26035840700713	39.102253458627274	88.1265742497811	196.8351818951168
9140654137	117.6177088522355	0.0	112.52898400339228	92.25057281909082	105.13139095432052	102.74952490148924
42430304	74.72259234056851	111.61240603627316	0.0	49.35545630742384	98.37977709857763	207.08838474391337
42430329	25.367136033144688	92.8716719948256	99.6274944401518	0.0	49.02432079115382	157.73292843648952
42430352	88.37835573678383	103.68512225473457	155.23220923606016	123.75646934632483	0.0	108.70860764533572
42449961	195.8182860623725	119.01351661953284	221.82470750302687	201.07392014536023	136.97686389027967	0.0
42435644	88.56981457436434	158.99960242703222	49.218413683992229	96.7426526981829	145.7669734893367	254.4755811346724
42453952	484.3606567734411	491.4254315875613	409.6380644328725	458.99352074029633	508.0178415314502	568.4675242156673
42447020	225.5842409210115	148.929277046326552	251.59066236316659	230.83987500399993	166.74281874891875	71.126553259339397
42428493	268.84565584320666	208.09256894328170	194.12306350263813	243.478519810062	292.5028406012157	301.09350897236845
42431549	107.64828164636798	181.5665250551964	181.90864005337508	143.026395255909	82.04816142537389	132.27761380738792
42430233	273.5444373935078	310.4342510892125	198.8218450529393	248.17730136036317	297.201622151517	405.91022979685277
42435598	222.03652754102703	292.4663153936949	182.55187290163565	230.2093656648456	279.2336864559995	387.9422941013352
42430271	153.1513504833714	190.04116417907608	78.42875814280289	127.78421445022671	176.8085352413805	285.5171428867163
42436700	135.44315414829882	109.90248860762756	161.44957558895317	140.69878823128656	76.60173197620601	62.452501143498424
42430633	160.002297041011	230.4320848936789	122.66002107196104	168.17513516482956	217.1994559559834	325.90806360131904
42435684	88.05098756217225	162.7747317228626	162.31134596917937	123.42910117171326	62.45086734117817	142.08859160227829
1825841742	150.445608241459	225.16935240214937	224.70596664846607	185.823721851	124.8454880204649	177.77016285059887
4347534767	176.69548445022997	158.66980321194578	210.2168901932714	189.4661028356048	125.36904658052424	109.28942002180182
42430253	216.21451615031964	253.10432984602429	141.4919238097511	190.84738011717496	239.87170090832876	348.5803085536645
42457319	377.5919573337549	309.11533559289074	302.86936499318637	352.22482130061024	401.2491420917641	386.15742822209966
42440012	86.3433158665918	160.2615592754202	160.6036742735989	121.72142947613278	60.743195645597694	114.75657672150476
42440287	521.783271193462	528.8480460075823	447.06067885289355	496.4161351603174	545.4404559514713	574.5888659208799
42430378	146.22225440209462	159.50389804583742	211.05098502716305	181.6003680116356	120.62213418110056	110.12351485569346
5706569898	433.8916663420589	390.43507671044074	359.1690740014903	408.52453030891417	457.548851100068	467.4771693385466
42427427	159.37573537004738	229.8055232227153	119.90572866392598	167.54857349386597	216.5728942850198	325.2815019303555
42430126	396.3699384239805	410.0979256634496	321.64734608341195	371.00280239083577	420.0271231819896	494.3211857381085
42446266	173.2046357908632	100.78545360799228	98.48204345029468	147.83749975771852	195.7885430171427 6	198.43699005943552
4597668020	146.28077610543136	73.76672178047946	71.55818376486285	120.91364007228668	168.86468333171092	171.41825823192272
42448693	208.84158055398845	136.4830084737074	134.11898821341595	183.47444452083985	231.425487780264	234.1345449251507
42451665	398.28095821245654	325.92238613217795	323.558365871888	372.91382217931186	420.8648654387361	388.9985100416264
42439996	50.84295673940279	125.83380603441134	125.10331514640993	86.2210703489438	25.242836511840871	133.95144416374444
42429782	471.01043538570434	398.6518633054273	396.2878430451359	445.64329935255967	493.594342611984	442.4240037540063

Fig. 12: Cost representation of system (T), used for optimization

Other Fixed Parameters:

Additional considerations include the number of pick-up locations (for Scenario A) or stops (for Scenario B) system-wide, the total coverage radius for each selected "facility" (to maximize demand), the total geographical footprint of the system (to balance equity of geographical access with density of demand coverage), and lastly, the weighting coefficient to customize the optimization objective, which is touched on in the next section (3.2). While these fixed parameters can be made adjustable in the workflow, for this analysis, let's assume **1500 meters** to be a reasonable coverage radius, which is just under a mile, and set the geographical coverage to cover **at least 50%** of all study nodes (though, if we're prioritizing demand coverage in the objective, this constraint will likely not be needed). Lastly, set **4 facility nodes** for both scenarios.

Now, bringing everything together, the full list of parameters for the problem space can be defined as follows:

- (Set of all Manhattan candidate stops) N
- (Terminal Node) $N_{terminal}$
- (Number of Manhattan Facilities/Stops) $S \in \mathbb{Z}^+$
- (Coverage Radius per Stop) $R \in \mathbb{R}^+$

- (Geographical Coverage Prop. of Covered Nodes) $C \in [0,1] \subset \mathbb{R}^+$
- (Q70 Manhattan Origin Daily Ridership Estimate) $r_{q70} = 0.44 * (average\ daily\ Q70\ ridership)$
- (Subway demand, node i to Jackson Heights/Woodside) $r_i, \forall i \in N$
- (Node Demand) $h_i = r_{q70} r_i / \Sigma_{\forall i \in N} r_i, \forall i \in N$
- (Pairwise Distance Costs) $d_{ij} \in D, \forall (i,j) \in N \cup N_{terminal}$
- (Pairwise Time-based Costs) $t_{ij} \in T, \forall (i,j) \in N \cup N_{terminal}$
- (Weighting Coefficient) $\alpha \in [0,1] \subset \mathbb{R}^+$

3.2 Optimization Methodology and Discussion

With a two-sided goal to lower time-based cost but also expand coverage to our proposed systems, the problem is framed in terms of maximizing the weighted covered demand subtracted by the weighted time-oriented costs incurred by the system. In other words, the problem can be generalized as the following:

$$maximize[\alpha * (covered\ demand)\ -\ (1-\alpha) * (incurred\ system\ time)]$$

The first term of the objective function illustrates the process as a Maximal Coverage Location Problem (proposed by Church, et al.), as the goal is to find stops/drop-off locations that maximize demand based on a preset coverage condition, which is defined in the constraints. The second term is akin to the shortest-path/minimal-cost flow problem, which ensures the time to travel from Manhattan to LGA is minimized in accordance to the demand/coverage conditions. This is especially evident in the Scenario B formulation, which is a direct interpretation of the shortest-path problem embedded into the multi-objective formulation. This formulation was initially proposed by Current, et al., labeling it as the "Maximal Covering - Shortest Path" Problem, which attempts to bridge the two objectives into a single formulation, assuring Pareto optimality. The MCSP in Scenario B is further extended with respect to the problem assumptions (fixed constraints such as enforcing geographical fairness, and other methods to reduce the constraint space when enforcing coverage over each node/link), as well as an additional set of constraints to perform sub-tour elimination (we want the solution as a single "linked-list").

These optimization workflows were executed via Pulp, a Python package geared towards solving large mixed-integer programs such as the ones proposed here. Two core Python classes were built to perform flexible analysis on the two scenarios respectively, as well as to ensure interoperability between the generated system-definition datasets (such as those evaluated in section 3.1) into the workflow. Specifically, the node demands h and time-based costs T are utilized in the objective function, while the distance-based costs D are used in the coverage enforcement constraints. The code snippet below illustrates this workflow for Scenario A (Scenario B follows the same workflow with an added variable — a fixed origin node), which abstracts away much of the optimization-related code.

```
import pandas as pd
import numpy as np

#import optimization script functionality
from build_parallel_route import *

#fixed global parameters
NUM_STOPS = 4
COVERAGE_RADIUS = 1500
GEO_COVERAGE_PROP = 0.5

parallel_route_models = dict()
alphas = np.arange(0,1.05,0.05)

for alpha in alphas:

    #instantiate variables
```

```
model = ParallelRouteOptimation(
    alpha=alpha,
    num_stops=NUM_STOPS,
    coverage_radius=COVERAGE_RADIUS,
    geo_coverage_prop=GEO_COVERAGE_PROP
)

#solve problem, find optimum
model.solve_problem()
parallel_route_models[alpha] = model

#build route parameters/summary statistics based on the optimal solution
model.build_route_parameters()

#visualize the route overlayed on top of the osmnx plot
model.plot_route()
```

3.3 Scenario A Formulation: Parallel Routing

$$maximize[\alpha \Sigma_{\forall i \in N} h_i x_i - (1-\alpha)\Sigma_{\forall i \in N} t_{i,N_{terminal}} y_i]$$

Subject to:

- (Enforce S number of stops) $\Sigma_{\forall i \in N} y_i = S$
- (Balance with fair system-based coverage) $\Sigma_{\forall i \in N} x_i / |N| = C$
- (Individual node/facility coverage) $x_i \leq \Sigma_{j \in N_i} y_j, N_i = \{j | d_{i,j} \leq R\}, \forall i \in N$

Where:

- (Is node i covered?) $x_i \in [0,1] \subset \mathbb{Z}^+$
- (Is node i a stop/facility?) $y_i \in [0,1] \subset \mathbb{Z}^+$

3.4 Scenario B Formulation: Sequential Routing

$$maximize[\alpha \Sigma_{\forall i \in N} h_i x_i - (1-\alpha)\Sigma_{(i,j) \in N \cup N_{terminal}} t_{ij} y_{ij}]$$

Subject to:

- (Enforce S number of stops) $\Sigma \Sigma_{\forall (i,j) \in N \cup N_{terminal}} y_{ij} = S - 1$
- (Balance with fair system-based coverage) $\Sigma_{\forall i \in N} x_i / |N| = C$
- (Node preservation/flow) $\Sigma_{\forall j \in N \cup N_{terminal}} y_{ij} - \Sigma_{\forall j \in N \cup N_{terminal}} y_{ji} = f_i, \forall i \in N \cup N_{terminal}$, where:

$$f_i = \begin{cases} 1 & i = N_0 \\ -1 & i = N_{terminal} \\ 0 & otherwise \end{cases}$$

- (Individual node/link-wise coverage) $x_k \leq \Sigma_{j \in S'_k} \Sigma_{i \in N_j} y_{ij}, \forall k \in N \cup N_{terminal}$
- (Coverage set per node k) $S'_k = \{j | d_{j,k} \leq R\}, \forall k \in N \cup N_{terminal}$
- (Upstream nodes relative to j) $N_j = \{i | i \neq j \ \wedge d_{i,N_{terminal}} \leq d_{N_0,N_{terminal}}\}, \forall j \in S'_k$
- (Sub-Cycle elimination) $t'_i - t'_j + M y_{ij} \leq M - 1, \forall (i \neq j) \in (N \setminus N_0) \cup N_{terminal}$
- (A route shouldn't consist of a node going to itself) $y_{ij} = 0, \forall (i = j) \in N \cup N_{terminal}$

Where:

- (Is node i covered?) $x_i \in [0,1] \subset \mathbb{Z}^+$
- (Is link i,j included in the route?) $y_{ij} \in [0,1] \subset \mathbb{Z}^+$
- (Cycle elim. dummy variables) $t'_i \in \mathbb{R}^+$

Note: N_0 is fixed, which represents the origin node for the route. Set $N_0 = 42428368$, which corresponds to the node at the Wall St subway stop. We would ideally want a route that traverses through the entire Manhattan CBD without having to route away from the airport at any point, so choice of an initial location is important here.

3.5 Findings

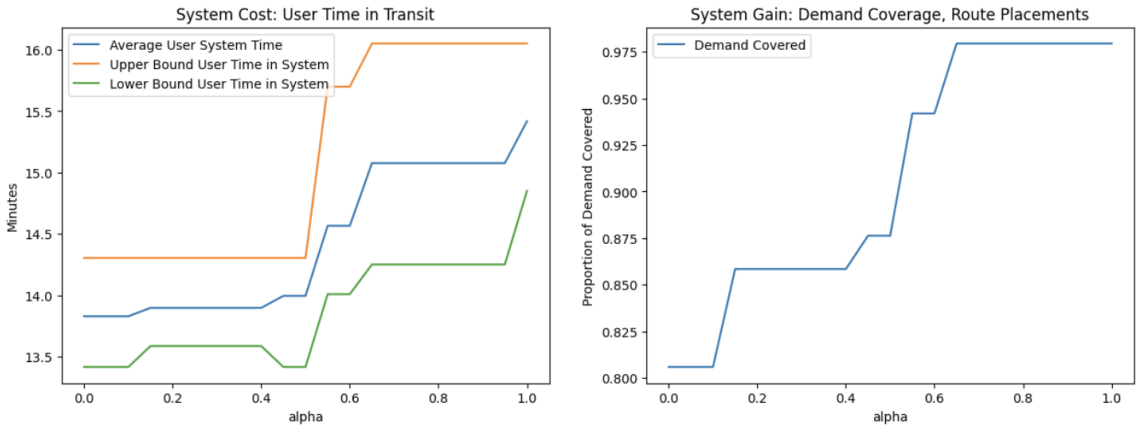

Fig. 13: Cost-Benefit Visualizations under Scenario A

Fig. 14: Scenario A Routing Scheme: $\alpha = 0.25$. Note that coverage is biased towards the east-side of Manhattan as travel-time minimization is prioritized.

Fig. 15: Scenario A Routing Scheme: $\alpha = 0.75$. With coverage taking greater precedence, all candidate pick-up locations seem to be more "equally spaced" throughout the Manhattan CBD Region.

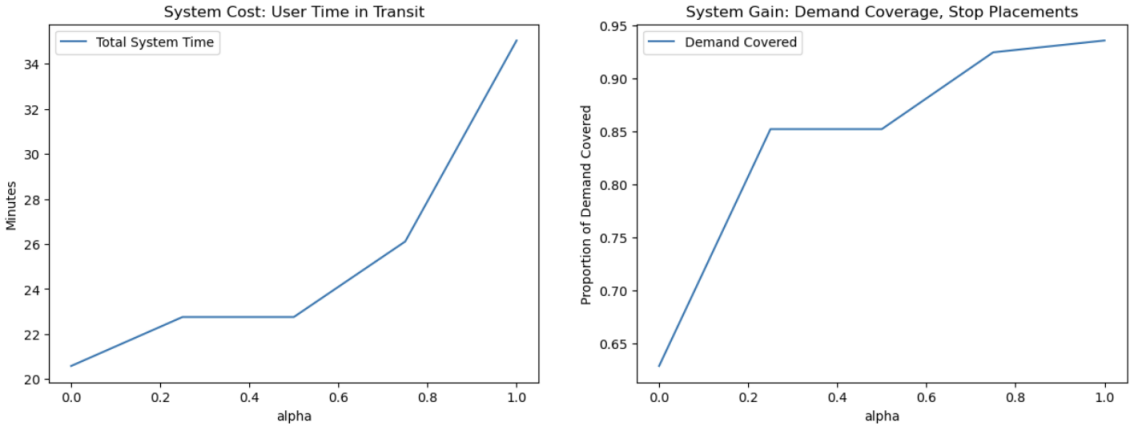

Fig. 16: Cost-Benefit Visualizations under Scenario B

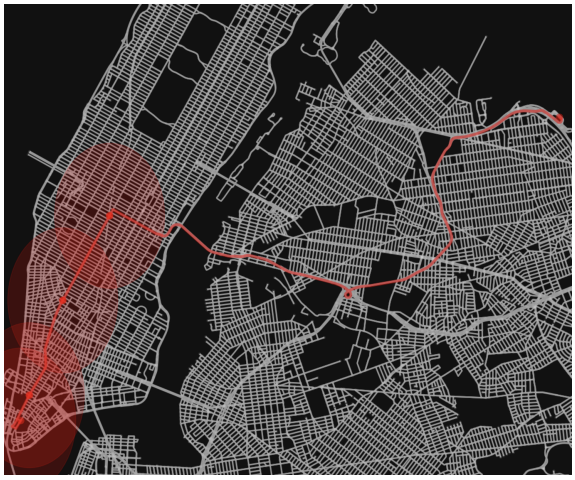

Fig. 17: Scenario B Routing Scheme: $\alpha = 0.25$. All coverage seems to be limited to areas south of Times Square/42nd and hence routed via the Queens-Midtown tunnel (higher speed limits permitted than other crossings?) in order to reflect the priority of cost-based optimization.

Fig. 18: Scenario B Routing Scheme: $\alpha = 0.75$. All of the Manhattan CBD is served at the expense of being routed through the Queensboro Bridge and surface streets in Queens.

All results seem to feature relatively predictable behavior; smaller α values yield more time-efficient routes but are biased towards regions closer to LGA Airport (in Scenario A) or regions closer to the origin node around Wall St. (Scenario B). On the other hand, larger α values provide much more unbiased coverage at the expense of longer, sub-optimal routes through the city. Hence, finding an ideal α_A and α_B is important in order to ground our alternative designs.

While choosing an optimal α_i in reality requires insight from a more policy-backed perspective, this was calculated via choosing the α with the largest difference between our rewards (demand) and costs (time-in-system), mirroring the same pattern evident in the objective function. Note that both rewards and (averaged) costs are assumed to be both monotonically increasing for an increasing α. As a result, to normalize the two measures, we can compute this by comparing the summary statistics for each $\alpha_i > 0$ with respect to $\alpha_i = 0$, which should also yield a monotonically increasing relationship.

$$tradeoff(\alpha) = [(demand\ covered)_\alpha/(demand\ covered)_0] - [(time\ cost)_\alpha/(time\ cost)_0]$$

$$\Rightarrow \alpha_i^* = argmax\{tradeoff(\alpha_i), \forall \alpha_i \in [0,1]\}, \forall i \in [A,B]$$

The end result is $\boldsymbol{\alpha_A = 0.65}$ and $\boldsymbol{\alpha_B = 0.25}$, and hence, these two parameters/route designs are selected for doing direct comparisons with one another, denoted in Sections 3.6 and 4. Note that the sample of α values used to evaluate Scenario B was not as comprehensive as Scenario A, as run-time per optimization was approximately 3 minutes due to the size and scale of the problem.

3.6 Constructing Headways

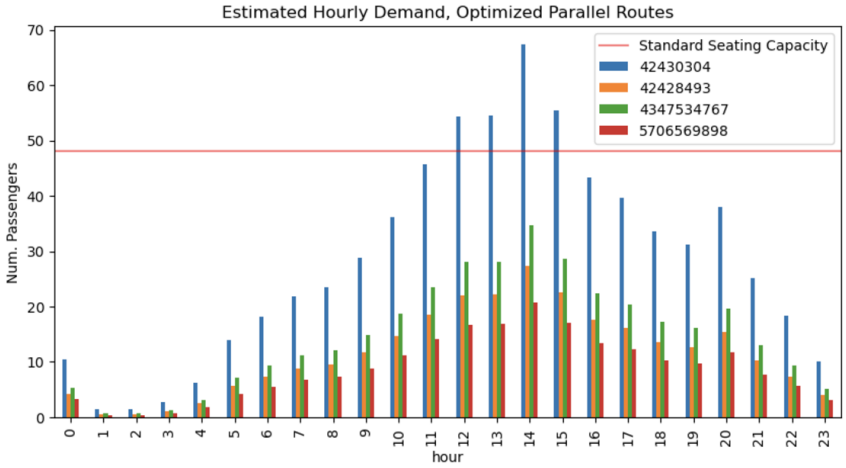

Fig. 19: System Hourly Demand Estimates: Scenario A (per Route)

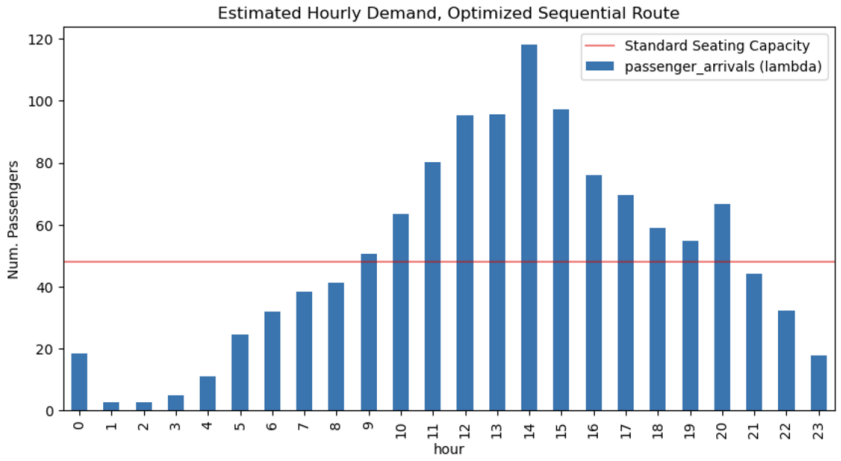

Fig. 20: System Hourly Demand Estimates: Scenario B

The final step in building a bus system is to establish a service pattern along these proposed routes. Leveraging the optimal solutions from Section 3.5 as well as hourly ridership estimates along the Q70+ service, we are interested in how demand is distributed along these stops and what is an optimal service frequency to meet this demand (while maximizing bus-utilization). No hard optimization is involved, as this is a simple calculation based on existing assumptions in the ridership data (the NYC Open Data Portal via API, referenced in section 2.1) as well as the route parameters outputted by the route-design/stop-placement optimization models defined in this section. The calculation is described as follows:

$$headways(t, R) = 60/ceil([r_{q70}(t) * \Sigma_{i \in R} h_i]/B)$$

- $r_{q70}(t)$ = estimated proportion of Manhattan-origin Q70 ridership at hour t
- B = bus seating capacity, fixed at 48
- R is the set of all covered nodes along a set route per the optimization output ($R = \{i | x_i^* = 1, i \in N\}$).

x_i^* is the decision variable output determining whether or not node i is covered by the route under optimality. Summing the corresponding demands to the covered nodes yields the total captured demand in the system. As stated in Section 2.4, assume a standard articulated bus seats 48 passengers. Ideally, we would like to build a schedule that meets this target (full utilization) without exceedance. Lastly, use ceiling division to ensure an integer result for the number of buses needed, to divide the total hourly demand by the bus capacity to ensure the utilization ratio stays below 1. The resulting plan after calculation is described as follows:

- **(Scenario A)** 60 minute headways for all sub-routes, except during the hours between 12:00 - 3:00 PM for route with pick-up node 42430304 (around Herald Square/Penn Station) with 30 min headways.
- **(Scenario B)** 20 minute headways between 2:00 - 4:00 PM, 30 minute headways between 9:00 AM - 2:00 PM and 4:00 PM - 9:00 PM, 60 minutes otherwise.

While it might seem as though the sequential model in Scenario B suggests more frequent service, note that the proposal in Scenario A equates to running 4-5 buses an hour; even though headways at each pick-up zone are for the most part hourly, assuming all routes are active in parallel, this equates to 4 buses per hour system-wide. Because more resources (buses) are supported under Scenario A, utilization per bus is bound to be less than that of Scenario B, assuming the covered demand across both systems are roughly similar (which is addressed in Section 4). This hypothesis is validated when plotting the hour-by-hour bus-utilization curves across both systems, seen in Figure 21. The parallel route utilization curve is an average of all four routes, for the sake of doing a direct comparison cross-scenario.

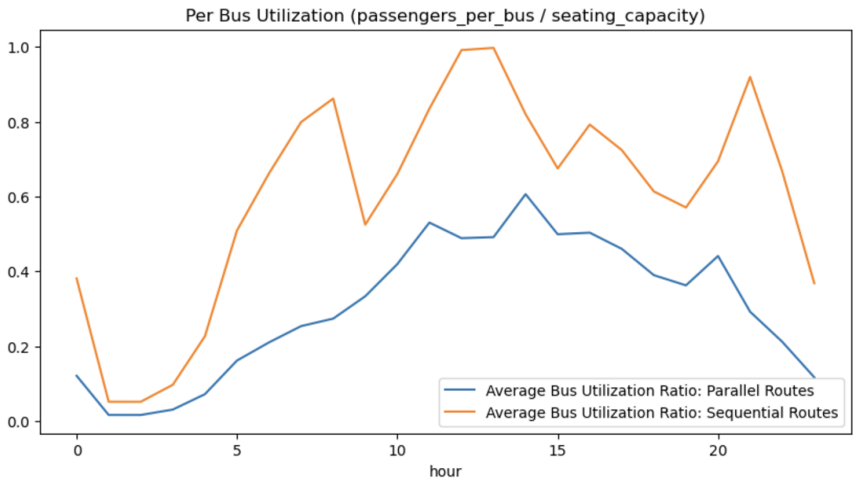

Fig. 21: System Hourly Demand Estimates: Scenario B

One important assumption in building these headways is the absence of congestion in both systems. Referring back to Figure 10, when measuring the Benchmark wait-time performance, despite expected utilization ratio staying under the threshold ($\phi = 1$) under the assumptions of the current Q70+ schedule, the ground truth suggests a different picture, where rush hour delays can build bottlenecks in the system, resulting in more crowded buses at the front of the queue. This could easily be the same case for these alternative scenarios, especially Scenario B which traverses through some of the most vehicle-dense corridors of Manhattan (6 Av/Broadway).

4 Takeaways

4.1 Benchmark vs Alternatives

We're now ready to evaluate the performance of all three systems combined. With respect to the benchmark minimum user system time, both the parallel (A) and sequential (B) direct-links drastically out-perform, with time-in-transit cut by more than half (around 43% for both alternative systems) across both of the best case scenarios and a pretty significant reduction even across the worst case scenarios (48% under the parallel scheme, 79% under the sequential scheme). The Min/Max user times specify the time-in-system from the closest/furthest stops to LGA Airport under a single configured α. Note that the benchmark best-case scenario accounts for the minimized time in transfer (at Jackson Heights/Woodside) as outlined in Section 2.3 (no

	Demand Coverage	Min. User System Time	Max. User System Time	Max. Buses per Hour	Min. Buses per Hour	Max Utilization Ratio per Bus
Benchmark	100%	33	78	7	3	0.857565
Alternative: Parallel	98%	14.25	16.05	5	4	0.951612
Alternative: Sequential	93%	14.3	26.11	3	1	0.99739

Fig. 22: Cross-System Results Summary (Assuming $\alpha_A = 0.65$ and $\alpha_B = 0.25$)

delays), which yields a less biased comparison as none of the alternative systems were measured on the basis of congestion effects. Both alternative systems approximately capture >= 90% of the existing Benchmark demand, under the assumption that demand is absorbed within a 1500 meter radius from each stop (see section 3.1), and riders are willing to use the service within this distance.

As seen in Figure 22, the parallel routing scheme (Scenario A) appears to achieve the best performance across the two key measures with respect to its counterpart. This is potentially because of inherent system-design factors (less entropy on a per-route basis, in terms of complexity), as a rider even under the worst case scenario will save time by traveling directly to LGA Airport, versus the sequential model which forces the rider to traverse the route stop-by-stop. Additionally, because distance is minimized on a per-route basis (versus per "chained" link under the Scenario B formulation), if demand is prioritized in the weighting (α), this allows for greater spatial variance in facility location as the selected nodes aren't bound to the same route/line; increasing demand priority for the sequential model on the other hand yields an increase in system cost at an accelerated rate especially for $\alpha > 0.8$, as the proposed solution will "bind-itself" together to coverage optimize at the expense of traversing away from the destination. This is not the case for Scenario A as each route is conceived separately in the route design and model formulation.

However, implementing Scenario A comes with its own disadvantages. As stated in Section 3.6, Scenario A requires more resources to address the decentralized nature of the system, serving potentially under-utilized routes. While this might be advantageous from the perspective of the rider (more capacity, more comfort), this model is potentially more cost-inefficient from the perspective of the MTA. Especially considering the difference in demand coverage isn't entirely significant across both alternative systems, the gain in potential revenue from riders is unlikely to make up for resource, labor, and maintenance costs associated with adding extra routes/buses under the parallel model. Hence, from an operational perspective, a traditional bus-route (or fewer direct parallel routes) is potentially more feasible.

4.2 Conclusion + Areas of Future Study

The purpose of this report is to evaluate and assess the performance of two proposed bus route designs, through the power of empirical data and various optimization methodologies. As demonstrated, this proof-of-concept is a first step in establishing the case for more efficient transit connections from demand-heavy regions of NYC to LGA Airport, a critical gateway for domestic air-travel for NYC Area residents. Most of the focus was around stop/facility placement and cost-minimization as it pertains to fixed travel-times. However, in reality, costs are often incurred on a dynamic basis (congestion effects), and a future implementation could consider penalizing route selection based on some function of distance, travel-time, and predicted delays in the system, conditional on the time-of-day, location, and other related features. Another potential idea to analyze congestion effects could be to perform a discrete event simulation of the entire system; as stated in Section 2.4, this is too difficult of a scenario to be defined in terms of a single queuing configuration, whether it be of passengers waiting to be served at each node or buses operating under congestion effects. A discrete event simulation could capture the dynamics in a way that is deterministic (fixed states) but also takes into account the variance in behavior from both the server side (buses) and consumer side (riders). Hence, while this report validated some of the benefits of the proposed strategies, a lot more work needs to be done in order to ground these plans into better emulating a realistic setting, which requires more complexity into an already comprehensive modeling framework.

References

1. Current, J. R., Re Velle, C. S., & Cohon, J. L. (1985). The maximum covering/shortest path problem: A multiobjective network design and routing . European Journal of Operational Research, 21(2).
2. Q70-SBS Bus Timetable [Review of Q70-SBS Bus Timetable].(2024, June 30). MTA Bus Company. https://new.mta.info/document/6621

3. THE PORT AUTHORITY OF NEW YORK & NEW JERSEY AIRTRAIN LGA LGA GROUND ACCESS MODE CHOICE MODEL AND AIRTRAIN RIDERSHIP FORECAST. (2018). https://www.anewlga.com/wp-content/uploads/2018/10/LGA-AirTrain-Ridership-Report.pdf

4. Uber Shuttle: LaGuardia Airport (LGA) to/from Midtown Manhattan. (2024). Uber. www.uber.com/global/en/r/airports/lga/shuttle/