Maïwenn Danno

Integer Optimization project (15.083)

Micro-transit optimization with transfers

July 27, 2023

1 Introduction

As part of my Research Assistantship with Prof. Jacquillat, I am developing an model to optimize (off-line) microtransit operations and consolidate passengers trips requests in urban areas. This model relies on a digital platform that collects trips requests (origin and destination locations, departure time and number of passengers) in batchs and assign them to available buses, providing a convenient and efficient way of traveling for commuters. Buses travel along a network of virtual bus stops to facilitate the vehicle-customer coordination: they pick-up passengers at a bus stop within walking distance of their origin and will drop-off them off at another bus stop within walking distance of their destination.

This research problem falls into the broad umbrella of vehicle routing problems with vehicle-customer coordination and time windows, with 3 additional levels of complexity:

- We consider **high-capacity busses** that can carry multiple passengers up to a capacity Q.
- We want to allow **transfers** at some hubs locations, where passengers may have to step out of a bus to wait for another bus that would bring him closer to his destination. Due to the penibility of transfers, we limit the number of transfers per passenger to 1.
- We want to add **flexibility** in the allocation of passengers to virtual bus stops for pick-up and drop-off, within walkable distance from origin and to destination.

As part of this project, we have developed and implemented two models for microtransit operations: one incorporating vehicle indices, and another without, and we have conducted a scalability analysis on synthetic data. Furthermore, we have compared the solution quality produced by our model against several benchmarks that do not incorporate all three aforementioned considerations.

2 Methodology

2.1 Time-space network definition

In order to avoid big-M formulations for capacity and time windows constraints and create a tighter formulation, I built my optimization model on a time-space network that encapsulates such considerations. The time-space network is built based on the depots and virtual bus stops locations, and such that:

- We have as many locations as many virtual bus stops + one location for the bus sink.
- The different times correspond to a discrete scale $1 : step : horizon$. The step choice is crucial, since an increased granularity would lead to an increased precision and efficiency of our solution, but would also result in a larger network and longer running times.
- All arcs between nodes that are located at virtual bus stops (or depot locations) are consistent with (rounded) driving time between them.
- Each arc has a cost, proportional to the driving time between the two locations. Arcs to sink are costless.

- No arc leaving the sink, No arc incoming a bus depot, such that busses have to leave depot and arrive at the sink.

Once the time-space network is set, I create different parameters that encapsulates some constraints of my model (s.t. assignation to close vbs, time constraints, etc) based on the customer requests inputs. For each customer request i, we consider as given the passengers load q_i, the expected departure time t_i and the origin and destination locations (x_{oi}, y_{oi}), (x_{di}, y_{di}).

Non-exhaustive list of the model parameters:

- I: set of customer requests, indexed by i
- $\mathcal{T}$: set of discrete times $= \{0 : step : T\}$
- H: set of hubs locations, i.e. set of vbs that can serve as transfer places for customers.
- K: set of available busses, initially located at some virtual bus stops, denoted as bus depots $d_1, ..., d_K$.
- vo_i (vo_d): set of possibles locations for pick-up (resp. drop-off) for customer request i[1], based on acceptable walking distance from requested origin and destination.
- N: set of all time-space nodes (l, t), with $l \in L$ and $t \in \mathcal{T}$
- N^*: set of all time-space nodes (l, t) except those located in bus depots or sink
- A: set of time-space arcs
- $\tilde{A}_d^+$: set of all possible traveling time-space arcs from each depot location d
- γ_i: shortest travel (walking to pick-up + driving + walking from drop-off) time for customer i
- $\delta_i = t_i + (1 + G)\gamma_i$: Time by which customer i must arrive at destination by, defined with a *detour ratio* G.
- $O_i \subset N$: set of possible time-space nodes (l^o, t^o) for pick-up of request i, based on acceptable walking distance, expected departure time i and maximum arrival time δ_i, with:
  - $l^o \in vo_i$
  - $t^o \in \mathcal{T} \cap [t_i + wo_i^{l^o}, \delta_i - (\gamma_i - wo_i^{l^o})]$[2]
- $D_i \subset N$: set of possible time-space nodes (l^d, t^d) for drop-off i, with:
  - $l^d \in vd_i$
  - $t^d \in \mathcal{T} \cap [t_i + wo_i^{\gamma} + s(vo_i^{\gamma}, l), \delta_i - wd_i^{l^d}]$ [3]

In order to model the flexible assignation to virtual bus stops for pick-up and drop-offs, I am also defining the set of possible combinations of pick-up and drop-off locations for each customer. For each customer request $i \in I$, I am thus defining the possible paths sets P_i^D and P_i^I, such that:

- $P_i^D = \{$direct paths (o, d), $\forall o \in vo_i, d \in vo_d\}$
- $P_i^I = \{$indirect paths (o, h, d), $\forall o \in vo_i, h \in H, d \in vo_d$

Selecting any path $p \in P_i$ for customer i is associated with a cost $wo_i^o + wo_i^d + \lambda\tau^p$, where $\tau^p = 1$ if the path p is indirect and $= 0$ otherwise. Since paths only contain location information, I am also defining the set of pick-up, transfer and drop-off time-space nodes that are compatible with each path:

- $O_i^p \subset O_i$ contains pick-up time-space nodes whose location is consistent with path p
- $D_i^p \subset O_i$ contains drop-off time-space nodes whose location is consistent with path p

[1] wo_i^l (resp. wd_i^l) denotes the walking time from any vbs l from origin (resp. destination) location of customer i

[2] $(\gamma_i - wo_i^l)$ is an upper bound of the travel time for customer i between l and his destination

[3] $s(vo_i^{\gamma}, l)$ is the shortest travel time between the best pick-up location (as defined by γ_i) for customer i and a location l, which is computed with Dijkstra

- H_i^p: contains all time-space nodes that customer i can use for transfer with the path $p \in P_i^I$; is the empty set for any $p \in P_i^D$.

In a final step, I leverage the Dkikstra algorithm in order to reduce the arcs set A_i accessible to any customer $i \in I$: I built an arc reduction heuristic that selects all arcs that any customer could travel along in order to meet the deadline δ_i, i.e. there is a route starting from any element of O_i and ending at any element of D_i and that uses this arc.

2.2 Model formulation

I have developed two different models to optimize the bus routes and the vbs-customers assignment, one that tracks the index k of the vehicle that each customer travels in, and one that does not need this tracking. The two complete models, that do generate the same solutions, are presented in Appendix. In the next subsections, I will develop the decision variables and constraints that compose them.

A) First model : with index k tracking

Decision variables

Passengers and vehicles routing across the network, and the passenger-path assignation are driven by three decision variables which are defined as:

- $\xi_i^p \in \{0,1\}, \forall i \in I, \forall p \in P_i$
  $\xi_i^p = 1$ if the customer i is assigned to path p, $\xi_i^p = 0$ otherwise.
- $x_{ia}^{pk} \in \{0,1\}, \forall i \in I, \forall a \in A, \forall p \in P_i, \forall k \in K$
  $x_{ia} = 1$ if the request i is assigned to arc a according to path p; $x_{ia}^p = 0$ otherwise.
- $z_a^k \in \{0,1\}, \forall a \in A, \forall k \in K$
  $z_a^k = 1$ if a vehicle k is assigned to arc a; $z_a^k = 0$ otherwise.

Constraints

The feasible set for these decision variables is defined with many constraints, listed below:

- **1 Path:** Each passenger may be assigned to maximum one path.
- **Linking ξ:** Passengers can only travel along arcs of the selected path.
- **Pick-up:** Each customer must be picked-up at one of the pick-up nodes of the selected path.
- **Drop-off:** Each customer must be dropped-off at one of the drop-off nodes of the selected path.
- **Transfer:** Each customer can have maximum one transfer, and *only if* the selected path is indirect.
- **Pass flow:** Every passenger that comes into a node within a vehicle leaves this node within the same vehicle (outside pick-up, drop-off and transfer nodes).
- **Pass transf:** For every customer path that includes a transfer, balance between the incoming and outgoing flow at the transfer node (over all vehicles).
- **1 in O/D:** Customers cannot be assigned to arcs that come into one of his possible pick-up locations, and cannot be assigned to arcs that come out of one of his possible drop-off locations.[4]
- **Capacity:** Maximum Q passengers travel within a same vehicle at any given moment.
- **Linking z:** Passengers may only travel along an arc with a vehicle if that same vehicle that travels along that arc, except if this arc is stationary at the selected hub location (to allow passengers to wait outside for another bus).
- **Depot:** Each vehicle can leave its depot maximum once
- **Veh flow:** Balance between the incoming and outgoing flow of each vehicle at each node (outside depot loations).

[4]This constraint aims at preventing from undesirable customer routes between possible two possible pick-up nodes and two possible dro-off nodes

Objective function

The goal is to minimize three things in parallel:

- The total travel time for customers and the presence of transfers, computed with the function

$$Cust(x, \xi; \mu, \beta, \lambda) = \mu Walk(\xi) + \beta Wait(x) + Traveling(x) + \lambda Tr(\xi) \tag{1}$$

  - Total Walking time : $Walk(\xi) = \sum_{i \in I} \sum_{p \in P_i} (w^o_{oi} + w^d_{di}) \xi^p_i$
  - Total Waiting time : $Wait(x) = \sum_{k \in K} \sum_{i \in I} \sum_{p \in P_i} \sum_{n \in O^p_i} \sum_{a \in A^+(n)} (T(a)^5 - t_i - w^o_{oi}{}^6) * x^{pk}_{ia}$
  - Total Traveling time : $Traveling(x) = \sum_{k \in K} \sum_{i \in I} \sum_{p \in P_i} \sum_{a \in A_{noH}} c_a x^{pk}_{ia}{}^7$
  - Total Transfer time : $Transfer(x) = \sum_{k \in K} \sum_{i \in I} \sum_{p \in P_i} \sum_{a \in A_H} c_a x^{pk}_{ia}{}^8$
  - Transfer cost: $Tr(\xi) = \sum_{i \in I} \sum_{p \in P_i} \tau_p \xi^p_i$

- The vehicle driving time and utilization, computed as:

$$Veh(z, \nu) = \sum_{k \in K} \sum_{a \in A} c_a z^k_a + \nu \sum_{k \in K} \sum_{a \in \tilde{A}^+_d} z^k_a \tag{2}$$

- The unmet demand, computed as:

$$Unmet(\xi) = \sum_{i \in I} (1 - \sum_{p \in P_i} \xi^p_i) \tag{3}$$

These three objectives will be ponderated in the global objective function, using parameters α_1, α_2.

B) Alternative model : no index k tracking

More recently, I have come up with a new model that does not need to track the k index to model all the constraints of the problem. First of all, I made some modifications on the time-space network generation algorithm, to deal with the absence of the index k:

- I substituted each hub location h by 3 hub locations, designated with $h_1, ..., h_3$, which correspond to the 3 parking slots available at the hub.
- I added K locations in my time-space network, corresponding to the busses depot locations, designated by $d_1, ..., d_K$.
- No travelling arc from nodes at the sink location, nor to nodes located in depot locations
- No arc between two parking slots at the same hub location
- I removed from A_i any arc that leaves depot locations or arrives at the sink, since those arcs are reserved for the vehicles.

I also introduced new notations:

- d: list of busses depot locations, indexed with vbs locations id.
- N_{d_k}: set of time-space nodes located in the k^{th} depot location
- N_s: set of time-space nodes located at sink location
- $N^* = N \setminus \{N_s \cup N_d\}$, where $N_d = \cup_{d_k \in d} N_{d_k}$
- $H^p_i(t) \subset H^p_i$: contains all the nodes of H^p_i that are at time t
- $A_{H^p_i}$: Stationnary arcs at hub locations of H^p_i (empty if the path p is direct)

[5] $T(a)$=Starting time of arc $a \in A$
[6] w^o_{oi} = Walking time between customer i origin and pick-up location o
[7] $A_{noH} = A \setminus A_H$
[8] A_H: arcs set that contains all the stationnary arcs at hubs locations

Decision variables

- $\xi_i^p \in \{0,1\}, \forall i \in I, \forall p \in P_i$
  $\xi_i^p = 1$ if the customer i is assigned to path p, $\xi_i^p = 0$ otherwise.
- $z_a = 1$ if a vehicle travels along arc a [9]
- $x_{ia}^p = 1$ if customer i travels along a and the path p is selected

Constraints

I also modified some constraints to handle the absence of index to track the vehicles.

- I added one constraint (”Max 1 incoming”) to impose that maximum 1 vehicle can arrive in a time-space node, in order to avoid undesirable transfers with the passenger flow balance constraint.
- I modified the ”Transfer” constraint to allow transfers of passengers between two parking slots located at the same hub location, while ensuring that passengers arrive in a hub before they can leave it with another vehicle.

Objective Function

The computation of the objective function without the index k is very close to the original one:

- The total travel time for customers and the presence of transfers, is computed with the function

$$Cust(x, \xi; \mu, \beta, \lambda) = \mu Walk(\xi) + \beta Wait(x) + Traveling(x) + \beta Transfer(x) + \lambda Tr(\xi) \tag{4}$$

  - Total Walking time : $Walk(\xi) = \sum_{i \in I} \sum_{p \in P_i} (w_{oi}^o + w_{oi}^d) \xi_i^p$
  - Total Waiting time : $Wait(x) = \sum_{i \in I} \sum_{p \in P_i} \sum_{n \in O_i^p} \sum_{a \in A^+(n)} (T(a)^{10} - t_i - w_{oi}^{o\,11}) * x_{ia}^p$
  - Total Traveling time : $Traveling(x) = \sum_{i \in I} \sum_{p \in P_i} \sum_{a \in A_{noH}} {}^{12} c_a x_{ia}^p$
  - Total Transfer time : $Transfer(x) = \sum_{i \in I} \sum_{p \in P_i} \sum_{a \in A_H} {}^{13} c_a x_{ia}^p$
  - Transfer cost: $Tr(\xi) = \sum_{i \in I} \sum_{p \in P_i} \tau_p \xi_i^p$

- The vehicle driving time and utilization, computed as:

$$Veh(z, \nu) = \sum_{a \in A} c_a z_a + \nu \sum_{\substack{d_k \in d \\ a \in \tilde{A}_{d_k}^+}} z_a \tag{5}$$

- The unmet demand, computed as:

$$Unmet(\xi) = \sum_{i \in I} (1 - \sum_{p \in P_i} \xi_i^p) \tag{6}$$

3 Results

3.1 Scalability analysis

While the model presented in Appendix is very promising in terms of vehicle-passengers coordination in urban mobility, it faces obvious scalability issues. Indeed, the computation of solutions can take hours as the size of the time-space network and the number of custome requests increase. The Table 1b presents the running time of the two models on two simplified maps that I generated synthetically:

[9] The binary nature of z imposes that maximum one vehicle travels along each arc
[10] $T(a)$=Starting time of arc $a \in A$
[11] w_{oi}^o = Walking time between customer i origin and pick-up location o
[12] $A_{noH} = A \setminus A_H$
[13] A_H: arcs set that contains all the stationnary arcs at hubs locations

- A uniformly-distributed map, as displayed on Figure 1, with of uniformly-distributed virtual bus stops and customers, willing to travel from the left to the right of the map.
- A clustered map, as displayed on Figure 2, with clustered groups of customer in the top left and bottom left of the picture, and bigger x-axis dimension.

The Table 1b shows clearly the performance improvement that removing the k-index has brought: the model without the index k runs up to 20 times faster than with the index k! Not surprisingly, it also shows how the number of vbs is the parameter that affects the most the solving time, as it directly affects the network size.

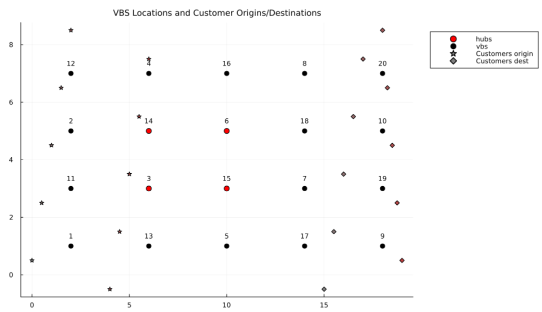


Figure 1: Uniformly-distributed customers and vbs

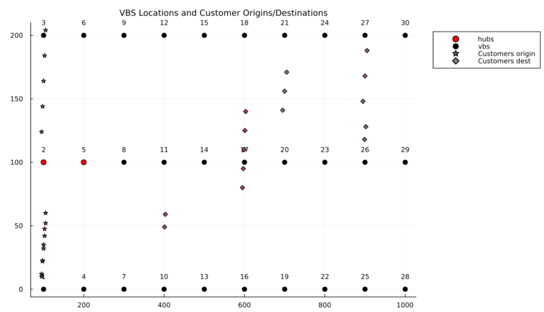


Figure 2: Clustered customers and vbs

Run time (sec)	With k	Without k
5 cust, 6 vbs, 3 veh	0.9	0.6
5 cust, 10 vbs, 5 veh	26.3	2.9
10 cust, 6 vbs, 3 veh	2.4	1.6
10 cust, 6 vbs, 5 veh	2.3	1.5
10 cust, 10 vbs, 5 veh	205.1	9.5

(a) Run times for uniformly-distributed customers map

Run time (sec)	With k	Without k
5 cust, 6 vbs, 3 veh	1.3	0.5
5 cust, 10 vbs, 5 veh	97.5	21.4
10 cust, 6 vbs, 3 veh	2.7	0.8
10 cust, 6 vbs, 5 veh	5.2	0.8
10 cust, 10 vbs, 5 veh	444.8	100.7

(b) Run times for clustered customers map

Table 1: Scalability analysis

3.2 Comparison with benchmark models

In order to analyze the quality of the solutions of my model, I built three benchmark models to compare with:

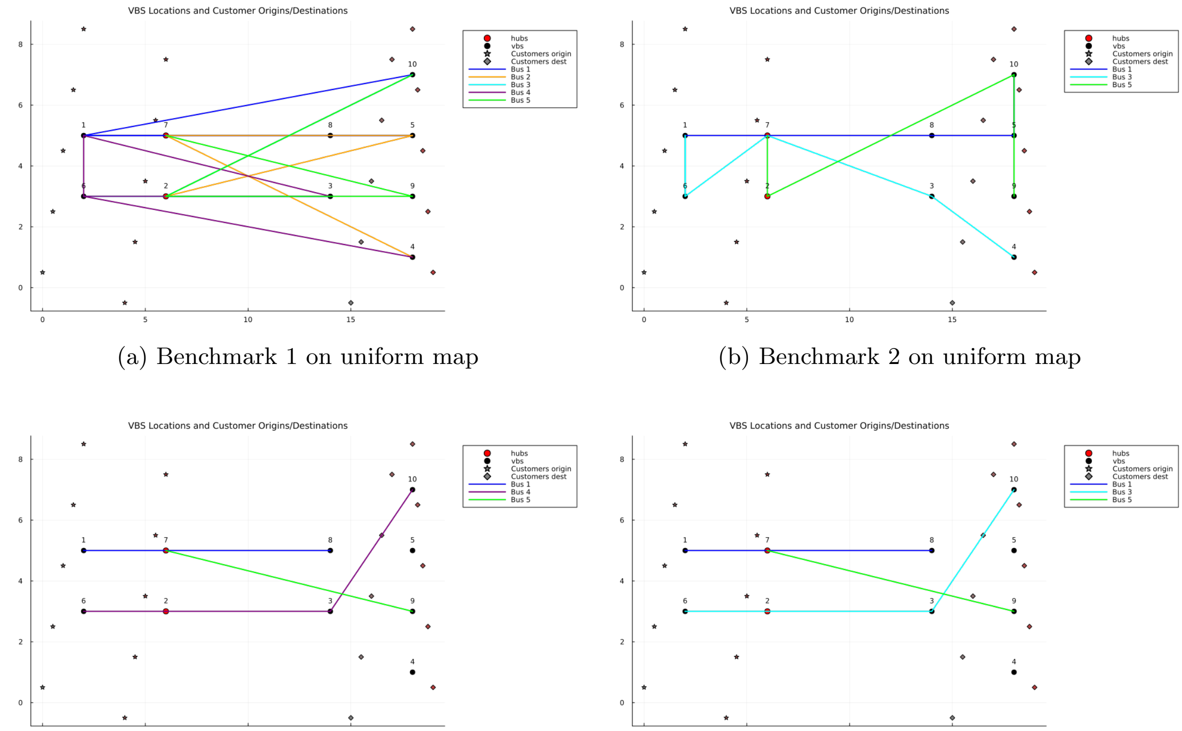
- **Benchmark 1:** Uses vehicles of capacity 1, no transfers and assignation to closest vbs.
- **Benchmark 2:** Uses vehicles of capacity Q=10, no transfers and assignation to closest vbs.
- **Benchmark 3:** Uses vehicles of capacity Q=10, no transfers but flexible assignation to vbs within walking distance.

I also defined several Key Performance Indicators (KPIs) to evaluate the solutions quality:

- KPIs on customer pain:
  - Average number of transfers
  - Average waiting time at pick-up location
  - Average walking time (origin + destination)
  - Average traveling time
  - Average efficiency (ratio ideal trip duration / real trip duration)
- KPIs on vehicle utilization
  - Number of vehicle used

- Total distance traveled by the vehicles
- Average capacity of used vehicles
- Average service time (from leaving the depot to dropping the last passenger)
- Average empty time (in service but without any passenger)

The Figure 3d displays the routing of the busses from the solution of the 3 benchmarks and my model on the uniform map with 10 customers, 10 vbs and 5 vehicles. Similarly, the Figure 4d displays the routing of the busses from the 4 models applied on the clustered map with 10 customers, 10 vbs, and 5 vbs. The Tables 2, 3, and 4 display the KPIs values and the run times of the solutions we obtain for the clustered map with the 4 models.



(a) Benchmark 1 on uniform map

(b) Benchmark 2 on uniform map

(c) Benchmark 3 on uniform map

(d) My model on uniform map

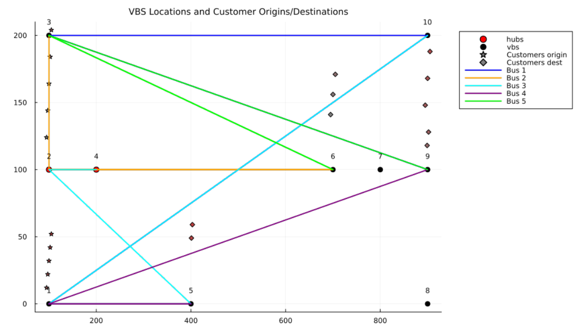
Figure 3: Comparison of the solutions on uniform map

Customer KPIs	Benchmark 1	Benchmark 2	Benchmark 3	My Model
Mean walking time	42	42	64	76
Mean waiting time	158	18	11	11
Mean traveling time	162	176	153	150
Route efficiency	73%	93%	94%	91%
Mean # transfers	0	0	0	0.2

Table 2: Customer KPIs values for the different benchmark models on clustered map

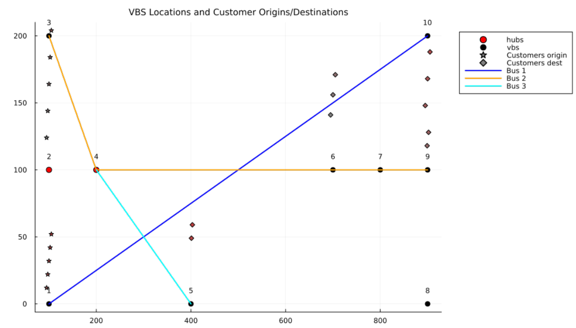
Vehicle KPIs	Benchmark 1	Benchmark 2	Benchmark 3	My Model
Vehicles used	5	3	3	2
Total distance traveled	930	129	113	48
Mean service time	930	215	189	120
Mean empty time	281	0	0	0
Mean capacity	0.4	1.6	1.3	2.1

Table 3: Vehicle KPIs values for the different benchmark models on clustered map

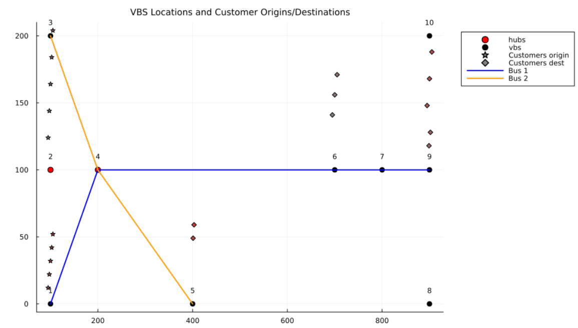


(a) Benchmark 1 on clustered map

(b) Benchmark 2 on clustered map



(c) Benchmark 3 on clustered map



(d) My model on clustered map

Figure 4: Comparison of the solutions on clustered map

Run time (sec)	Benchmark 1	Benchmark 2	Benchmark 3	My Model
Solving time (sec)	3	1.6	23	101

Table 4: Solving times for the different benchmark models on clustered map

4 Takeaways and Next steps

First of all, the scalability analysis has shown that by eliminating the variable 'k' in our formulation, we were able to improve the running times by up to 20 times. However, our model still scales very badly compared to the benchmarks. However, we were able to highlight the benefits of our three innovative strategies (i.e. high capacity busses, flexible assignation and transfers) by comparing the KPIs values achieved by our solution on the clustered map with three other benchmark models. First, we discovered that high-capacity buses dramatically improved customer waiting time by 90%, route efficiency by 20%, and reduced vehicle distance by 80%. Flexibility in the allocation of pick-up and drop-off points improved vehicle utilization by 12%, maintaining similar customer walking times and route efficiency. Furthermore, enabling transfers at strategic hub locations resulted in an additional 33% improvement in vehicle utilization and 60% reduction in distance, particularly for clustered customer requests.

In conclusion, this project has created a solid foundation for enhancing microtransit operations, and there is great potential for real-world applications that could substantially reduce transportation costs while boosting the flexibility of transport solutions. Looking forward, the next steps involve further improving model scalability by leveraging column and/or row generation. In addition, real-world tests using actual customer requests will be carried out, with the objective function coefficients adjusted to produce solutions that effectively meet the needs of both the customers and the transportation companies.

A Model with the k index

Decision variables

- $\xi_i^p \in \{0,1\}, \forall i \in I, \forall p \in P_i$
  $\xi_i^p = 1$ if the customer i is assigned to path p, $\xi_i^p = 0$ otherwise.
- $x_{ia}^{pk} \in \{0,1\}, \forall i \in I, \forall a \in A, \forall p \in P_i, \forall k \in K$
  $x_{ia} = 1$ if the request i is assigned to arc a according to path p; $x_{ia}^p = 0$ otherwise.
- $z_a^k \in \{0,1\}, \forall a \in A, \forall k \in K$
  $z_a^k = 1$ if a vehicle k is assigned to arc a; $z_a^k = 0$ otherwise.

Model formulation:

$$\min_{x,\xi,z} \quad Cust(x,\xi;\mu,\beta,\lambda) + \alpha_1 Veh(z,\nu) + \alpha_2 Unmet(\xi) \qquad \text{(Multi-Obj)}$$

$$\text{s.t.} \quad \sum_{p\in P_i} \xi_i^p = 1 \qquad \forall i \in I \qquad \text{(1 Path)}$$

$$x_{ia}^{pk} \le \xi_i^p \qquad \forall i,k,\forall p \in P_i, \forall a \in A_i \qquad \text{(Linking } \xi\text{)}$$

$$\sum_{k\in K}\sum_{p\in P_i}\sum_{n\in O_i^p}\sum_{a\in A_i^+(n)} x_{ia}^{pk} = 1 \qquad \forall i \in I \qquad \text{(Pick-up)}$$

$$\sum_{k\in K}\sum_{p\in P_i}\sum_{n\in D_i^p}\sum_{a\in A_i^-(n)} x_{ia}^{pk} = 1 \qquad \forall i \in I \qquad \text{(Drop-off)}$$

$$\sum_{k\in K}\sum_{p\in P_i^T}\sum_{n\in H_i^p}\sum_{a\in A_i^-(n)} x_{ia}^{pk} = \sum_{p\in P_i^T} \xi_i^p \qquad \forall i \in I \qquad \text{(Transfer)}$$

$$\sum_{a\in A_i^-(n)} x_{ia}^{pk} = \sum_{a\in A_i^+(n)} x_{ia}^{pk} \qquad \forall i,k,p,\forall n \in N_{\setminus vo_i, vd_i, H_i^p} \qquad \text{(Pass Flow)}$$

$$\sum_{k\in K}\sum_{a\in A_i^-(n)} x_{ia}^{pk} = \sum_{k\in K}\sum_{a\in A_i^+(n)} x_{ia}^{pk} \qquad \forall i,p,\forall n \in H_i^p \qquad \text{(Transfer)}$$

$$\sum_{k\in K}\sum_{p\in P_i}\sum_{n\in N_{vo_i}}\sum_{a\in A_i^-(n)} x_{ia}^{pk} = 0 \qquad \forall i \in I \qquad \text{(1 in } N_{vo}\text{)}$$

$$\sum_{k\in K}\sum_{p\in P_i}\sum_{n\in N_{vd_i}}\sum_{a\in A_i^+(n)} x_{ia}^{pk} = 0 \qquad \forall i \in I \qquad \text{(1 in } N_{vd}\text{)}$$

$$\sum_{i\in I_a}\sum_{p\in P_i} q_i x_{ia}^{pk} \le Q z_a^k \qquad \forall a \in A, \forall k \in K \qquad \text{(Capacity)}$$

$$x_{ia}^{pk} \le z_a^k \qquad \forall i \in I, a \in A_i, \forall k \in K, \forall p \qquad \text{(Linking z)}$$

$$\sum_{a\in \tilde{A}_{d_k}^+} z_a^k \le 1 \qquad \forall k \in K \qquad \text{(Depot)}$$

$$\sum_{a\in A^-(n)} z_a^k = \sum_{a\in A^+(n)} z_a^k \qquad \forall k \in K, \forall n \in N_{\setminus d_k} \qquad \text{(Veh Flow)}$$

B Model without the k index

Decision variables

- $\xi_i^p \in \{0,1\}, \forall i \in I, \forall p \in P_i$
  $\xi_i^p = 1$ if the customer i is assigned to path p, $\xi_i^p = 0$ otherwise.
- $z_a = 1$ if a vehicle travels along arc a [14]
- $x_{ia}^p = 1$ if customer i travels along a and the path p is selected

Model formulation:

$$
\begin{array}{llrr}
\min\limits_{x,\xi,z} & Cust(x,\xi;\mu,\beta,\lambda) + \alpha_1 Veh(z,\nu) + \alpha_2 Unmet(\xi) & & \text{(Multi-Obj)} \\
\text{s.t.} & \sum\limits_{p \in P_i} \xi_i^p = 1 & \forall i \in I & \text{(1 Path)} \\
& x_{ia}^p \leq \xi_i^p & \forall i, \forall p \in P_i, \forall a \in A_i & \text{(Linking } \xi) \\
& \sum\limits_{p \in P_i} \sum\limits_{n \in O_i^p} \sum\limits_{a \in A_i^+(n)} x_{ia}^p = 1 & \forall i \in I & \text{(Pick-up)} \\
& \sum\limits_{p \in P_i} \sum\limits_{n \in D_i^p} \sum\limits_{a \in A_i^-(n)} x_{ia}^p = 1 & \forall i \in I & \text{(Drop-off)} \\
& \sum\limits_{p \in P_i^T} \sum\limits_{n \in H_i^p} \sum\limits_{a \in \tilde{A}_i{}^-(n)} x_{ia}^p = \sum\limits_{p \in P_i^T} \xi_i^p & \forall i \in I & \text{(Transfer)} \\
& \sum\limits_{a \in A_i^-(n)} x_{ia}^p = \sum\limits_{a \in A_i^+(n)} x_{ia}^p & \forall i, p, \forall n \in N_{\setminus vo_i, vd_i, H_i^p} & \text{(Pass Flow)} \\
& \sum\limits_{n \in H_i^p(t)} \sum\limits_{a \in A_i^-(n)} x_{ia}^p = \sum\limits_{n \in H_i^p(t)} \sum\limits_{a \in A_i^+(n)} x_{ia}^p & \forall i, p, t \in \mathcal{T} & \text{(x-Transf)} \\
& \sum\limits_{p \in P_i} \sum\limits_{n \in N_{vo_i}} \sum\limits_{a \in A_i^-(n)} x_{ia}^p = 0 & \forall i \in I & \text{(1 in } N_{vo}) \\
& \sum\limits_{p \in P_i} \sum\limits_{n \in N_{vd_i}} \sum\limits_{a \in A_i^+(n)} x_{ia}^p = 0 & \forall i \in I & \text{(1 in } N_{vd}) \\
& \sum\limits_{i \in I_a} \sum\limits_{p \in P_i} q_i x_{ia}^p \leq Q z_a & \forall a \in A & \text{(Capacity)} \\
& x_{ia}^p \leq z_a & \forall i, p, \forall a \in A_i \setminus A_{H_i^p} & \text{(Linking z)} \\
& \sum\limits_{a \in \tilde{A}_{d_k}^+} z_a \leq 1 & \forall d_k \in d & \text{(Depot)} \\
& \sum\limits_{a \in A^-(n)} z_a \leq 1 & \forall n \in N \setminus N_s & \text{(Max 1 incoming)} \\
& \sum\limits_{a \in A^-(n)} z_a = \sum\limits_{a \in A^+(n)} z_a & \forall n \in N^* & \text{(Veh Flow)}
\end{array}
$$

[14] The binary nature of z imposes that maximum one vehicle travels along each arc