Frugal Graph-Based Spatially-Explicit Algorithm for Public Transport Accessibility Computations

Danielle Ben Shlomo, BGU
Prof. Noam Tractinsky, BGU
Dr. Eran Ben-Elia, BGU
Prof. Itzhak Benenson, TAU

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev
جامعة بن-غوريون في النقب

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Goal

Compute and visualize spatially-explicit **accessibility gaps** between private cars and public transportation users.

Accessibility: Definition

The ability of people to reach desired destinations and activities using available transportation modes.

Motivation

Understanding and improving travelers’ accessibility using public transportations has impact on:

- Social justice (reducing social exclusion)
- Environmental sustainability (reducing pollution)
- Market output (reducing traffic and increasing productivity)

Some aspects relate more to rural areas, while others are highly relevant to large metropolitan areas.

Background

- Previous research mostly measured accessibility at the TAZ level, which is a very low spatial resolution, and does not fully represent the traveler's perspective correctly. *Example: Israel has ~1.5 million buildings but only ~3K TAZs.*

- Most accessibility computations focus solely on the reachable areas and activities from different origins (*access areas*), rather than also dealing with the **accessibility of the different destinations** themselves (*service areas*) such as schools, hospitals etc.

- No standard visualizations that capture the full spatiotemporal aspect of accessibility, and are integrated in the public transportation planner's work.

What's New?

We aspire to:

1. **Compute high-resolution spatially-explicit accessibility gaps* of origins and destinations.**

2. Build a visualization tool for public transportation planners and policy makers.

(*) **Accessibility Gaps:** Accessibility using public transportation compared to using private cars.

Basic Terminology

Transportation Modes (M)

Transportation mode is denoted as $M \in \{B, C\}$ where B represents Bus (of generally Public Transportation) and C represents (private) Cars.

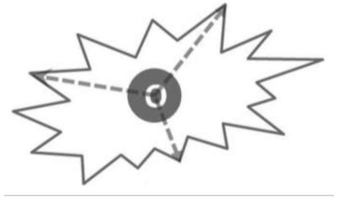
Mode Travel Time (MTT)

The total travel time from origin O to destination D using the transportation mode M.

Cars: walk from O to parking + time in vehicle + walk from parking to D

PT: walk from O to stop + total travel in buses + total transfer time (walk & wait) + walk from stop to D

Access and Service Areas



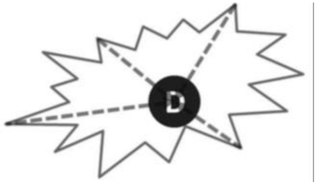
Access Area

$O = Origin \quad M = Mode \quad \tau = travel\ time$

Mode Access Area = $MAA_O(\tau)$

$= The\ area\ S, containing\ all\ destination\ buildings\ D\ that\ can\ be\ reached\ from\ O\ using\ M\ during\ MTT < \tau.$

Example: where can I reach from my home (O) within 40 minutes (τ) using publi



Service Area

$D = Destination \quad M = Mode \quad \tau = travel\ time$

Mode Service Area = $MSA_D(\tau)$

$= The\ area\ S, containing\ all\ origin\ buildings\ O\ from\ which\ D\ can\ be\ reached\ using\ M\ during\ MTT < \tau.$

Example: from where can people reach the hospital (D) within 1 hour (τ) if they use their private car (M)?

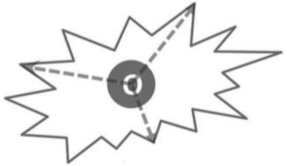
Capacity

$D_{k,capacity} = A\ quantitative\ measure\ describing\ the\ capacity\ of\ destination\ D\ for\ activity\ type\ k.$
Example: Number of jobs in destination D.

$O_{k,capacity} = A\ quantitative\ measure\ describing\ the\ capacity\ of$ origin O $for\ activity\ type\ k.$
Example: Number of potential employees residing in O.

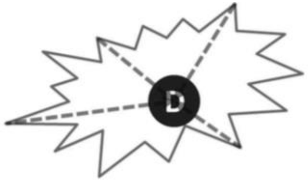
Potential Accessibility

$$\mathrm{MPA_{O,k}}(\tau) = \sum_{D}\{D_{k,capacity} | D \in MAA_{O}(\tau)\}$$



The total capacity of destinations one can reach from origin *O* using mode *M* while *MTT*<τ in order to engage in activity of type *k*.

$$\mathrm{MPS_{D,k}}(\tau) = \sum_{O}\{O_{k,capacity} | O \in MSA_{D}(\tau)\}$$



The total capacity of origins from which people can reach destination *D* using mode *M* while *MTT*<τ in order to engage in activity of type *k*.

Relative Accessibility

Bus to Car accessibility ratio for *Access Area*

$$AA_{O,k}^{Bus:Car}(\tau) = \frac{BPA_{O,k}(\tau)}{CPA_{O,k}(\tau)}$$

Bus to Car accessibility ratio for *Service Area*

$$SA_{D,k}^{Bus:Car}(\tau) = \frac{BPS_{D,k}(\tau)}{CPS_{D,k}(\tau)}$$

We want to aspire for the largest possible values

Numerator:
Public Transportation

$$AA_{O,k}^{Bus:Car}(\tau) = \frac{\boldsymbol{BPA_{O,k}(\tau)}}{CPA_{O,k}(\tau)}$$

$$SA_{D,k}^{Bus:Car}(\tau) = \frac{\boldsymbol{BPS_{D,k}(\tau)}}{CPS_{D,k}(\tau)}$$

<u>Task:</u> Compute $BPA_{O,k}(\tau)$ and $BPS_{O,k}(\tau)$

Numerator: Public Transportation

$$AA_{O,k}^{Bus:Car}(\tau) = \frac{\boldsymbol{BPA_{O,k}(\tau)}}{CPA_{O,k}(\tau)}$$

$$SA_{D,k}^{Bus:Car}(\tau) = \frac{\boldsymbol{BPS_{D,k}(\tau)}}{CPS_{D,k}(\tau)}$$

Three steps (Access Area focused for simplicity):

1. Compute reachable PT stops from each other stop in the country.
2. Increase resolution for building clusters.
3. Apply capacities on results to receive potential accessibility.

Reminder:
$BPA_{O,k}(\tau)$ = *Capacity of all reachable destinations from O using PT within* τ *time for activity type k.*
$BPS_{O,k}(\tau)$ = *Capacity of all origins from which people can reach D using PT within* τ *time for activity type k.*

Step 1: Compute reachable PT stops from each other stop in the country.

PT Data

The *General Transit Feed Specification* (GTFS):

Full timetable in txt format, representing all trips through 28K stops in Israel.

TGraph (Transit Graph)

We can represent the PT network as a mathematical graph (G) made of vertices (V), edges (E) and weights (W). Let's assume that a path P in G represents a journey in the PT network, and W(P) is the total travel time.

Access Areas: By computing shortest paths between all of the vertices pairs, and limiting to a maximal path weight, we can find all reachable nodes from all others.

Service Areas: By constructing a reversed graph $G' = (V, E', W')$ such that $E' = \{(v,u) | (u,v) \in E\}$ and $W'(v,u) = W(u,v)$, similarly computing the shortest paths will give us for each node n, all other nodes from which n can be reached within the time limit.

How do we apply the limitations of the PT schedule?
Time-expanded graph: Every event in the transit system is represented by a vertex.

TGraph

<u>Vertices (Nodes)</u>

$\langle LineID, TerminalDepartureTime, StopID \rangle$, with the attributes $StopArrivalTime,\ StopSequence$.

<u>Edges</u>

$(v_i, v_j) \in E$ iff one of the following:

A. *Direct*: The same physical vehicle *AND* $StopSequence_j = StopSequence_{i+1}$.

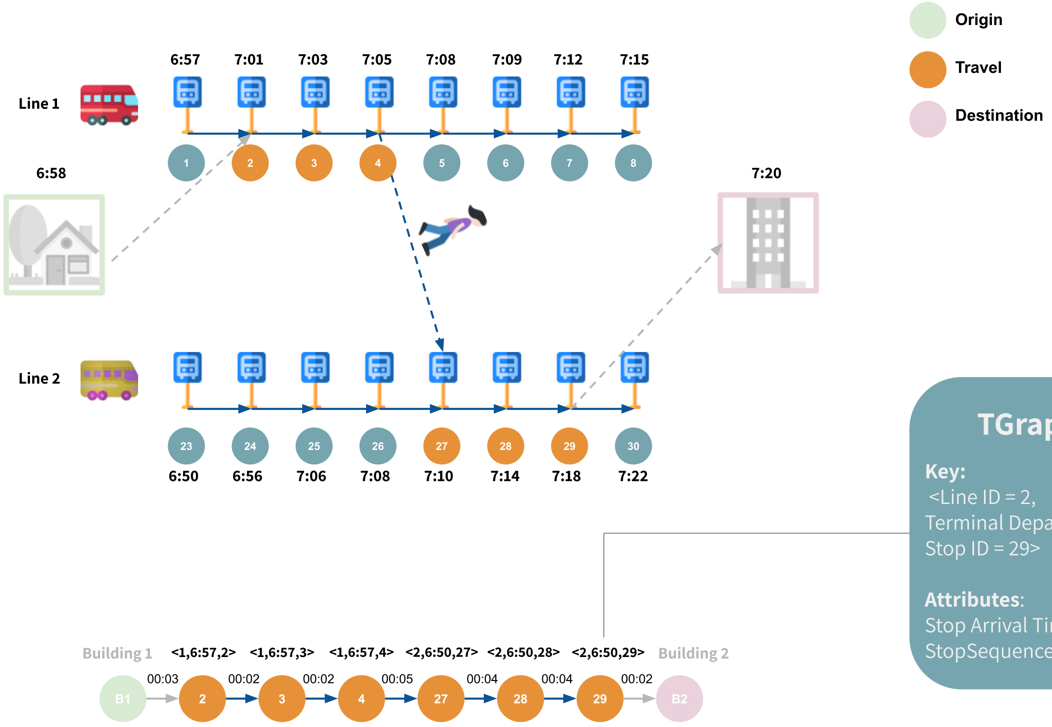
B. *Transfer*: Possible transfer by alighting $LineID_i$ and boarding $LineID_j$ with distance and time limitations.

<u>Weights</u>

(A) edges: time in vehicle

(B) edges: walking + waiting time

TGraph: Example



TGraph Node

Key:
<Line ID = 2,
Terminal Departure Time = 6:50,
Stop ID = 29>

Attributes:
Stop Arrival Time = 7:18
StopSequence = 7

Results

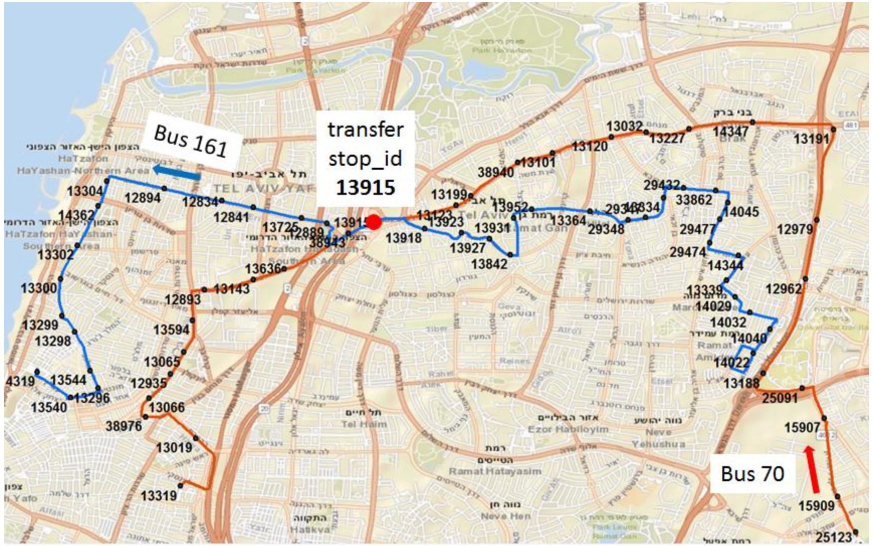
We constructed a TGraph for all of Israel with initial time boundaries (only a 90-minute morning time frame was used to construct the graph):

- 325K nodes
- 4.5M edges (300K direct edges and 4.2M possible transfers)

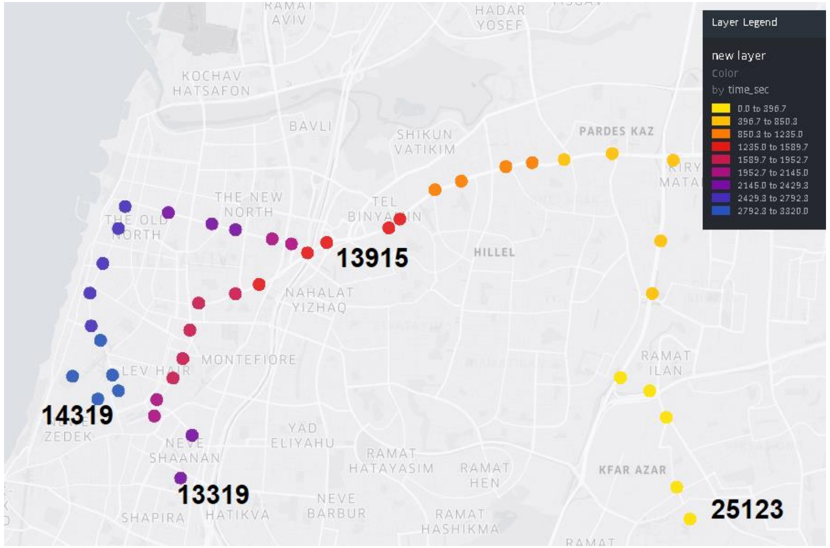
We performed all-pairs Dijkstra shortest-path computations in parallel on a 40-core machine with 128GB RAM within ~1 hour.

We used a 20 minute time-frame for allowed departure (*Access Area*) and arrival (*Service Area*).
We set the maximal walking distance to 200 meters and limited the total transfer time to 15 minutes.

Results: Simple Example



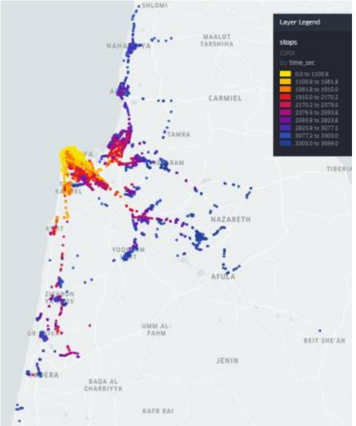
Raw data: 2 bus lines with one transfer

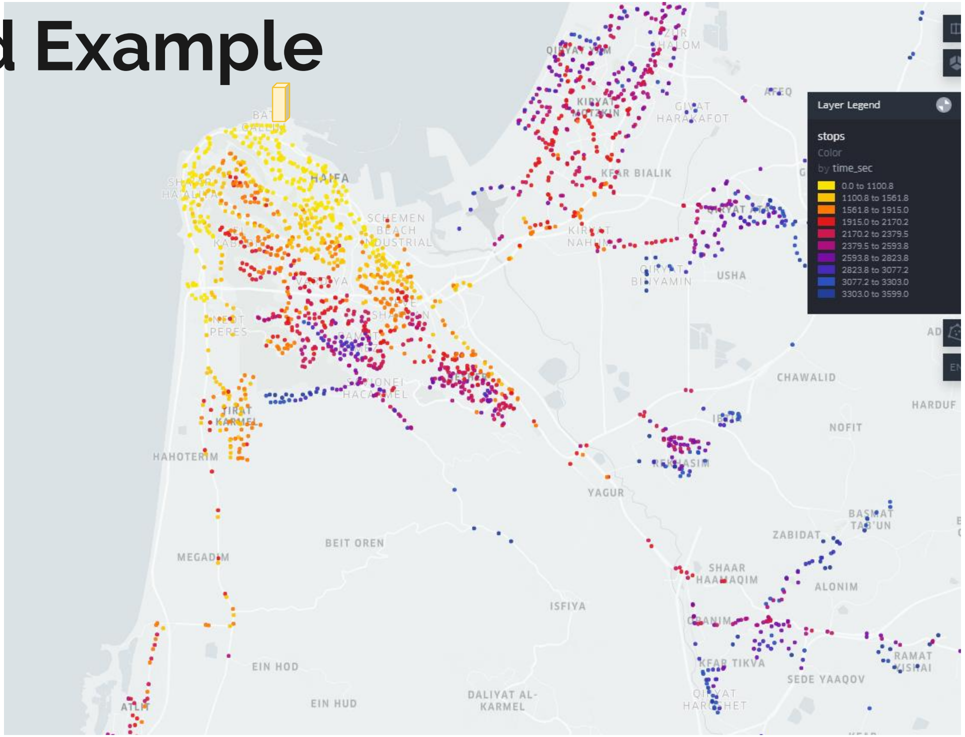


Result: travel times from origin 25123

Step 1: Compute reachable PT stops from each other stop in the country.

Results: Real World Example





Higher Resolution

For simplicity: let's assume we refer to individual buildings as origins and destinations (in practice we use building clusters).

In Israel: ~28K PT stops, and ~1.5 million buildings. Computing shortest paths between all pairs is computationally very expensive ($\sim 2^{41}$ pairs). We need another solution.

We use SQL-like queries to increase resolution and integrate buildings into the results.

Buildings Data

Higher Resolution: Simple Method

For each building we find nearby relevant stops and the walking time to them
$WalkTime_{Os}$ - Walking time from building O to stop s.

Precomputation

$SSTT_{ij}$ (**S**top to **S**top **T**ravel **T**ime) - The shortest travel time between s_i and s_j.
If stop j is not reachable from stop i within the time limit then $SSTT_{ij} = \infty$.

We have this from step 1

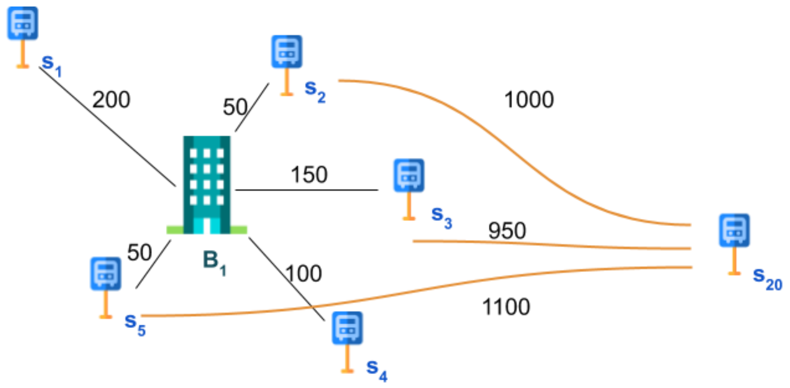
⇒ Building to Stop Travel Time

$$BSTT_{Os_d} = min\{WalkTime_{Os} + SSTT_{ss_d} | s \in S_O\}$$

Building	Nearby Stops
B_1	$S_{B_1} = \{\ldots\}$
B_2	$S_{B_2} = \{\ldots\}$
B_3	$S_{B_3} = \{\ldots\}$
…	…

Step 2: Increase resolution for building clusters.

Higher Resolution: Simple Method



Building	Nearby Stops
B_1	$S_{B_1} = \{s_1, s_2, s_3, s_4, s_5\}$
B_2	…
B_3	…

Nearby Stop	Total Travel Time Using the Stop
s_1	$WalkTime_{B_1 s_1} + SSTT_{s_1 s_{20}} = 200 + \infty = \infty$
s_2	$WalkTime_{B_1 s_2} + SSTT_{s_2 s_{20}} = 50 + 1000 = 1050$
s_3	$WalkTime_{B_1 s_3} + SSTT_{s_3 s_{20}} = 150 + 950 = 1100$
s_4	$WalkTime_{B_1 s_4} + SSTT_{s_4 s_{20}} = 100 + \infty = \infty$
s_5	$WalkTime_{B_1 s_5} + SSTT_{s_5 s_{20}} = 50 + 1100 = 1150$

Higher Resolution: Simple Method

Problem

What if we limit the departure/arrival time to specific times, e.g. departure between 8:00-8:20? We might get that the shortest travel time from s_1 to s_2 represents a trip that leaves at 8:01, and the walking time from building B to s_1 is 3 minutes, and suddenly we're no longer in the departure window we defined.

This is a simple example but can occur in more severe and complex ways, and becomes more relevant for service areas, where we might be late for an appointment/meeting/class etc.

From Reachable Nodes to Potential Accessibility

Given a node O with reachable nodes $\{n_1, n_2, \dots, n_k\}$, the potential PT accessibility for node O would be:

$$\mathrm{BPA}_{\mathrm{O,k}}(\tau) = \sum_{D} \{D_{k,capacity} \,|\, D \in BAA_O(\tau)\} = \sum_{i=0}^{k} (n_i)_{k,capacity}$$

Given a node D with nodes that can reach it within the time limit $\{n_1, n_2, \dots, n_k\}$, the potential PT accessibility for node D would be:

$$\mathrm{BPS}_{\mathrm{D,k}}(\tau) = \sum_{O} \{O_{k,capacity} \,|\, O \in BSA_D(\tau)\} = \sum_{i=0}^{k} (n_i)_{k,capacity}$$

(*) Building capacities for a chosen activity need to be computed separately.

Denominator:

Private Cars

$$AA_{O,k}^{Bus:Car}(\tau) = \frac{BPA_{O,k}(\tau)}{\boldsymbol{CPA_{O,k}(\tau)}}$$

$$SA_{D,k}^{Bus:Car}(\tau) = \frac{BPS_{D,k}(\tau)}{\boldsymbol{CPS_{D,k}(\tau)}}$$

<u>Task:</u> Compute $CPA_{O,k}(\tau)$ and $CPS_{O,k}(\tau)$

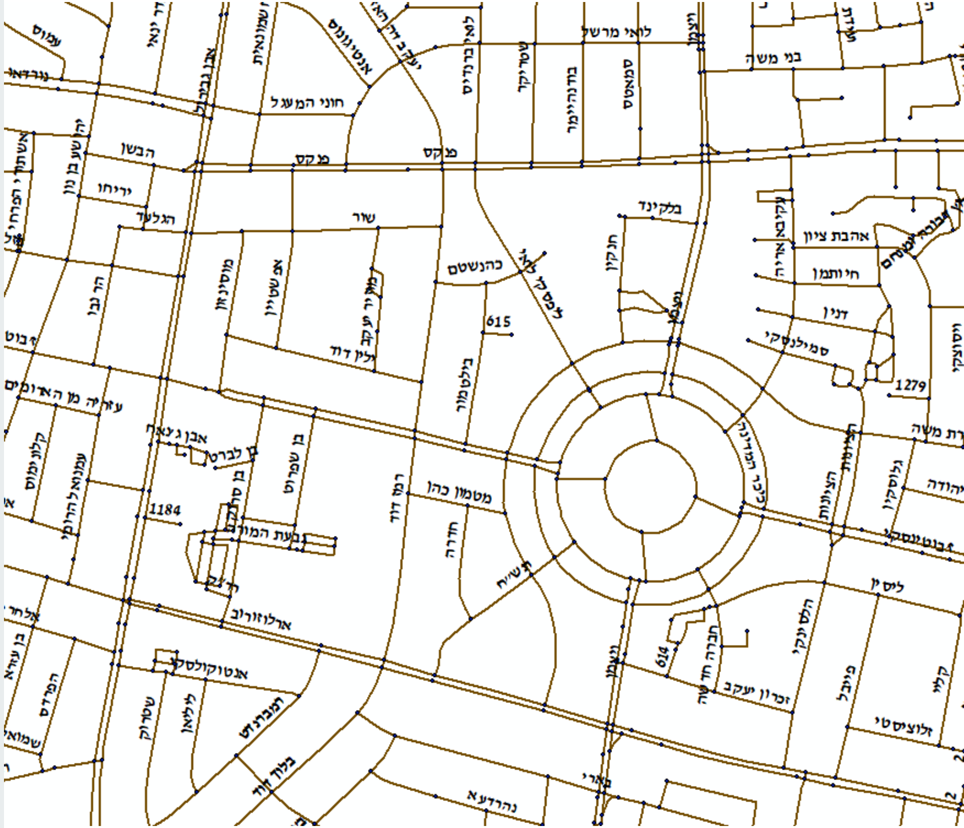
<u>Reminder:</u>

$CPA_{O,k}(\tau)$ = *Capacity of all reachable destinations from O using a private car within τ time for activity type k.*

$CPS_{O,k}(\tau)$ = *Capacity of all origins from which people can reach D using a private car within τ time for activity type k.*

Road Data

GISrael data set (Mapa): multiple shp layers describing in detail all roads in Israel (500K junctions and 650K road segments).



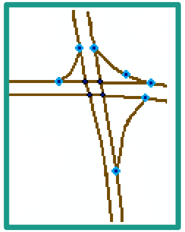
RGraph (Roads Graph)

We constructed a natural graph based on the roads data.

Nodes: Junctions (real junctions and pseudo-junctions)

Edges: Roads

Weights: Avg. time it takes to pass through the road



Pseudo-junctions in a large intersection

RGraph: Special Cases

When constructing the roads graph we encountered two cases the require special handling:

1. **Turn restrictions**

2. **Interchanges, Bridges, Tunnels** (a node is connecting multiple roads in different levels/heights).

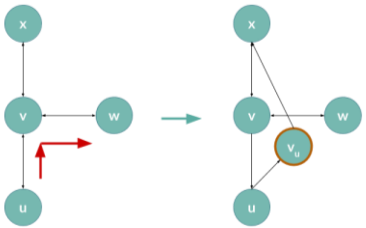
Both cases are handled the same and are referred to as “turn restrictions”.

RGraph: Turn Restrictions

For a given turn restriction $u - v - w$, meaning no turn is allowed from the edge (u, v) to the edge (v, w), we want to create a new pseudo-node, v_u. Such a node ***represents all traffic that goes through*** $\boldsymbol{v}$ ***and came from the edge*** $(\boldsymbol{u}, \boldsymbol{v})$. By deleting (u, v) and adding a new edge (u, v_u), we can control the outgoing traffic from v_u, and not allowing it to go into w. *Formally:*

$W(u, v_u) = W(u, v)$

$E \leftarrow E \backslash \{(u, v)\} \cup \{(u, x) | (v, x) \in E \ and \ x \neq w\}$



With this solution we are not harming the traffic between all other nodes surrounding v.

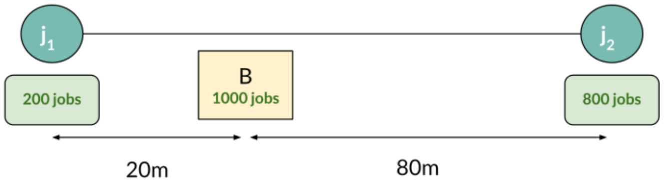
The final graph representing Israel roads network contains 500K nodes and ~950K edges.

Potential Cars Accessibility

To get high-resolution accessibility, two general approaches we are considering (WIP):

1. Add all buildings (or building clusters) as nodes and connect each one to X nearest junctions (Heavy).

2. Applying buildings capacities to junctions. One possible way of doing this is diving the capacity by weight based the building's relative distance from each of the X nearest junctions.
   Example:



(*) Since parking location is unknown, and junctions are very frequent, exact placement of the building is not critical.

(*) Building capacities for a chosen activity need to be computed separately.

Potential Cars Accessibility: Access Areas

Once we have a $RGraph$, we can compute all node pairs shortest-paths (computed in parallel using Dijkstra, 40 cores, 128GB RAM) with maximal travel time threshold τ. This gives us all reachable nodes from each junction within the time limit.

Given a node O with reachable nodes $\{n_1, n_2, \dots, n_k\}$, the potential car accessibility for node O would be:

$$\mathrm{CPA}_{\mathrm{O,k}}(\tau) = \sum_{D} \left\{ D_{k,capacity} \middle| D \in CAA_O(\tau) \right\} = \sum_{i=0}^{k} (n_i)_{k,capacity}$$

Potential Cars Accessibility: Service Areas

A very similar shortest-paths computation is performed to find the service areas for each node. This is done by reversing the graph, as we described for $TGraph$.

Given a node D with nodes that can reach it within the time limit $\{n_1, n_2, \ldots, n_k\}$, the potential car accessibility for node D would be:

$$\mathrm{CPS}_{\mathrm{D,k}}(\tau) = \sum_{O} \{O_{k,capacity} | O \in CSA_D(\tau)\} = \sum_{i=0}^{k} (n_i)_{k,capacity}$$

Next Steps

Computations:

Complete building integration for high-resolution results.

Visualization:

Tool for public transportation planners and policy makers, currently in the design phase.

Examples of user tasks:

- Compare general accessibility to a given destination using private cars and public transportation.
- Identify times of the day with high accessibility gaps (between private cars and public transportation) to a given destination.
- Identify areas that require attention and improved accessibility to a given destination.

Thank You

Questions?