Regional Transit Priority Setting

Reliability Metrics Methodology

January 2019

Where is traffic slowing down buses and trolleys? Where can improvements be made along routes that are negatively impacted by traffic but still demonstrate high ridership? Regional Transit Priority Setting seeks to answer these questions, because targeted improvements can deliver the highest return on investment. This document provides an overview of the logic and methodology used to compute reliability metrics.

The entire process can be condensed into the seven constituent parts outlined below.

Variable	Input	Process	Output 1	Process	Output 2	Process	Output 3
Average Speed	LinkSpeed_byLine.shp linkspeedBylineNameCode.csv	Linkspeed.R	linkspeed50.tif				
On-Time Performance	statsbyline_allgeom.shp	OTP.R	otp50.tif				
Travel Time Index	combined_tti_wtrolley.shp	TTI.R	tti.tif				
Ridership	Surface_Transit_Loads.shp statsbyline_allgeom.shp	Ridership.R	rider50pa.tif rider50.tif riderPA.tif riderNJ.tif	Raster Calc 1	linkExists.tif linkQual.tif lineExists.tif lineQual.tif addRast.tif riderOverall.tif	Focal Means and Raster Calc 2 / Export.R	weightedScore.tif riderOverlay.tif ws_focal.tif ws_smoothed.tif priority_score.tif priority_score_incl_ridership.tif class20Score_jk.shp class20Rider_jk.shp
	1	**2**	**3**	**4**	**5**	**6**	**7**

1: Input

This project considers four variables: average speed, on-time performance, travel time index (TTI), and ridership. The goal is to identify areas with an overlap of slow average speed, low on-time performance, high TTI, and (optionally) high ridership as targets for investment.

Variable	Level	File Name	Format	Source
Average Speed	Line	LinkSpeed_byLine.shp	Spatial	GTFS/TIM 2.3 (2015 Base Year)
	Link	linkspeedBylineNameCode.csv	Tabular	--
On-Time Performance	Line	statsbyline_allgeom.shp	Spatial	Survey of Transit Operators (2015-2017)
TTI	Link	combined_tti_wtrolley.shp	Spatial	2017 INRIX XD, TIM 2.3 (2015 Base Year)
Ridership	Line	statsbyline_allgeom.shp	Spatial	Survey of Transit Operators (2015-2017)
	Link	Surface_Transit_Loads.shp	Spatial	SEPTA

The shapefiles used to create link- and line-based scores present spatial problems both within and between files. Within files, several links or lines can run on top of one another. Consider Market Street, which has trolleys, buses, regional rail, and the subway line sharing approximately the same latitudes and longitudes, but at different elevations. This is part of the logic behind removing lines and links with exclusive rights-of-way, as including lines such as regional rail may artificially inflate the performance and ridership of lines that share space with traffic on the surface. Between files, the source shapefiles come from different sources. This means that the exact same line appears in slightly different places in every shapefile.

Rasterizing the data alleviates some of these concerns. Within datasets, it ensures we do not accidentally eliminate reliability metrics for links or lines that run on top of one another—for example, by only considering the topmost layer. Between datasets, it acts as a spatial buffer, ensuring that reliability metrics pertaining to the exact same line will stack on top of one another, even though each metric came from a different source and might have slightly different latitudes and longitudes.

2: Assign priority scores and rasterize

Because the intent of the project is to identify areas with an overlap of slow average speed, low on-time performance, high TTI, and high ridership, Part 2 first assigns priority scores to all features in each dataset. The analysis approaches vary slightly depending on the input dataset — for example, because SEPTA ridership data is available at the point level, priority scores are computed for each point and the set is transformed to areal data through inverse distance weighted interpolation.

Average speed

1. Read in `LinkSpeed_byLine.shp` (n = 67,458).
2. Remove empty and zero values (*resultant* n = 63,895).
3. Create a data frame `uniqueLinks` which identifies unique origin and destination nodes in the dataset (*resultant* n = 42,688). This is necessary because at least some node combinations have multiple links laid on top of one another.

4. Merge `uniqueLinks` with `linkspeedBylineNameCode.csv` to identify the mode type of each link.
5. Because we are primarily concerned with routes that do not have exclusive rights-of-way, keep only `tsyscodes Bus` and `Trl` (trolley) in `uniqueLinks`.
6. Now that `uniqueLinks` identifies each unique origin-destination combination, compute the weighted average link speed for these records. The weight is `cnt`, the number of times the route traverses the link. Results are saved in data frame `indivLinkSpeed`.
7. The weighted mean link speeds are positively skewed. Compute the natural logarithm of these scores to create a more normal distribution of scores.
8. Compute the cumulative probability of the logarithm of each weighted mean link speed. Because we are interested in links with the slowest average speed, the cumulative probability density function aggregates from ∞ to $-\infty$. Links with the slowest speeds receive scores closest to `100`.
9. Rasterize the shapefile at a `50m` resolution. The process for this and all other files is as follows:
    a. Generate a blank raster at a `50m` cell resolution, CRS `NAD83 UTM Zone 18N` (`EPSG: 26918`) and extent corresponding to the study area (the extent of the TIM TTI referenced below, or `xmin 374610.3`, `xmax 586310.3`, `ymin 4309335`, `ymax 4538285`).
    b. Rasterize the shapefile, taking the mean of all records that overlap with a given cell. By computing the mean cell value, we account for shapefiles that have multiple observations, at either the line or link level, running on top of one another.
    c. Replace all cells with empty values to zero.
    d. Write the resulting raster file.

On-time performance

1. Read in `statsbyline_allgeom.shp` (*n* = `172`).
2. Remove empty and zero values (*resultant n* = `162`).
3. Compute the cumulative probability of each record of on-time performance. Because we are interested in links with the lowest on-time performance, the cumulative probability density function aggregates from ∞ to $-\infty$. Links with the lowest on-time performance receive scores closest to `100`.
4. Rasterize the shapefile at a `50m` resolution.

Travel time index

The INRIX TTI and the TIM TTI are both used in this project. While the INRIX TTI is preferred because it obtains the TTI from cell phone location data, the TIM TTI covers more of the DVRPC region. `combined_tti_wtrolley.shp` uses INRIX TTI where it is available and fills in with TIM TTI where it is not.

1. Read in `combined_tti_wtrolley.shp` (*n* = `24,395`).
2. Remove zero and negative values (*resultant n* = `21,149`). Note that negative values occur for TTI traveling the opposite direction of traffic on a one-way street.
3. Compute the cumulative probability of each record of TTI. Because we are interested in links with the highest TTI, the cumulative probability density function aggregates from $-\infty$ to ∞. Links with the highest TTI receive scores closest to `100`.
4. Rasterize the shapefile at a `50m` resolution.

Ridership

SEPTA surface transit loads at the link level and regional line-level ridership are both used in this project. While the SEPTA ridership data is preferred because of its geographic resolution, it is not available for buses operated by NJ Transit. `Ridership.R`: 1) computes priority scores for NJ Transit based on the line ridership for the entire region; 2) computes priority scores for SEPTA bus loads relative to the SEPTA loads dataset; and 3) creates two raster files that aid in differentiating the locations of SEPTA, NJ Transit, or the overlap of both.

Line-level ridership

1. Read in `statsbyline_allgeom.shp` (n = `172`).
2. Remove records with zero reported ridership (*resultant* n = `134`).
3. Compute the cumulative probability of each record of the natural logarithm of relative ridership. Because we are interested in links with the highest ridership, the cumulative probability density function aggregates from $-\infty$ to ∞. Links with the highest ridership receive scores closest to `100`.
4. Rasterize the shapefile at a `50m` resolution.

Link-level loads

1. Read in `Surface_Transit_Loads.shp` (n = `158,513`).
2. Because multiple surface transit lines share the same road (sometimes nearly 50 lines on the same road segment), rasterize the shapefile at a `50m` resolution, taking the local sum of surface transit loads.
3. Some grid cells span intersections and otherwise duplicate surface transit loads. The maximum recorded transit load for a single cell is `325,219.9`. If a cell's value exceeds the `95`th percentile of surface transit loads (`10,349`), replace the cell with the 95th percentile value.
4. Compute the cumulative probability of the surface transit loads. Because we are interested in cells with the highest loads, the cumulative probability density function aggregates from $-\infty$ to ∞. The highest loads receive scores closest to `100`.

Locations of NJ Transit and SEPTA service

1. Read in `statsbyline_allgeom.shp` (n = `172`).
2. Subset the field `name`. When the first three characters of `name` are `sep`, this indicates the line is a SEPTA line. When the first three characters of `name` are `njt`, this indicates the line is a NJ Transit line.
3. Subset `statsbyline_allgeom.shp` into separate NJ and PA files.
4. Rasterize the shapefiles at a `50m` resolution.

3: Output 1

The table below outlines the outputs from `Linkspeed.R`, `OTP.R`, `TTI.R`, and `Ridership.R`.

Variable	File Name	Format	Description
Average Speed	linkspeed50.tif	50m Raster	Priority scores for average speed (0-100). Cell values nearest 100 are the slowest observed average speeds.
On-Time Performance	otp50.tif	50m Raster	Priority OTP scores (0-100). Cell values nearest 100 are the lowest observed OTP.
TTI	tti.tif	50m Raster	Priority TTI scores (0-100). Cell values nearest 100 are the highest observed TTI.
Ridership	rider50pa.tif	50m Raster	Preliminary priority ridership scores (0-100) derived from SEPTA ridership data. Cell values nearest 100 are the highest observed ridership.
	rider50.tif	50m Raster	Preliminary priority ridership scores (0-100) derived from transit operators. Cell values nearest 100 are the highest observed ridership.
	riderPA.tif	50m Raster	Indicates presence of SEPTA bus, trolley, and BRT lines.
	riderNJ.tif	50m Raster	Indicates presence of NJ Transit bus lines.

4: Raster calculations 1

The ridership variables use a mixture of multiple files. Link-level data on transit loads should be used where available, and line-level ridership data should fill in the gaps where the link-level data is unavailable, especially in New Jersey. The result is a NJ-PA composite ridership index. All calculations in this section are performed in the QGIS raster calculator.

Simplified schematic of link- and line-level ridership scores

Link		
44	68	0
56	0	0
7	0	0

Line		
47	71	32
55	23	0
9	87	96

1. Reclassify `rider50pa.tif` and `rider50.tif` according to the rubric below. Outputs from this step are saved as `linkExists.tif` and `lineExists.tif`.
    a. if $link \neq 0, cell = 1$; else $cell = 0$
    b. if $line \neq 0, cell = 4$; else $cell = 0$

In QGIS raster calculator:

```
linkExists.tif = ("rider50pa@1" > 0) * 1 + ("rider50pa@1" = 0) * 0
lineExists.tif = ("rider50r@1" > 0) * 4 + ("rider50r@1" = 0) * 0
```

Simplified schematic of link- and line-level ridership coverage

Link		
1	1	0
1	0	0
1	0	0

Line		
4	4	4
4	4	0
4	4	4

2. Load `linkExists.tif` and `lineExists.tif`. Add the two rasters together. The added raster is saved as `addRast.tif`. Raster results indicate the following:
    a. if $cell = 5, link$ and $line$ exist ∴ use $link$
    b. if $cell = 4, link$ does not exist ∴ use $line$
    c. if $cell = 1, line$ does not exist ∴ use $link$
    d. if $cell = 0, link$ and $link$ do not exist ∴ $no\ data$

In QGIS raster calculator:

```
addRast.tif = "linkExists@1" + "lineExists@1"
```

Simplified schematic of raster addition

addRast		
5	5	4
5	4	0
5	4	4

3. Reclassify `addRast.tif` twice: first, as a multiplier for link-level surface transit scores when available; and second, as a multiplier for line-level ridership scores when link-level data is not available. The following rubric applies, and results are saved as `linkQual.tif` and `lineQual.tif`.
    a. `linkQual.tif`: if $cell = 5$ or $cell = 1, cell = 1$; else $cell = 0$
    b. `lineQual.tif`: if $cell = 4, cell = 1$; else $cell = 0$

In QGIS raster calculator:

```
linkQual.tif = (("addRast@1" = 5) OR ("addRast@1" = 1)) * 1 + (("addRast@1"
!= 5) OR ("addRast@1" != 1)) * 0
lineQual.tif = ("addRast@1" = 4) * 1 + ("addRast@1" != 4) * 0
```

Simplified schematic of link- and line-level ridership multipliers

linkQual		
1	1	0
1	0	0
1	0	0

lineQual		
0	0	1
0	1	0
0	1	1

4. Finally, calculate the NJ-PA composite ridership index by implementing the formula below. Results are saved as `riderOverall.tif`.

$$(linkQual \times link\ loads\ score) + (lineQual \times line\ ridership\ score)$$

In QGIS raster calculator:

```
riderOverall.tif = ("linkQual@1" * "rider50pa@1") + ("lineQual@1" *
"rider50r@1")
```

5: Output 2

The table below outlines the outputs that will be used in the second phase of raster calculations.

Variable	File Name	Format	Description
Average Speed	linkspeed50.tif	50m Raster	Priority scores for average speed (0-100). Cell values nearest 100 are the slowest observed average speeds.
On-Time Performance	otp50.tif	50m Raster	Priority OTP scores (0-100). Cell values nearest 100 are the lowest observed OTP.
TTI	tti.tif	50m Raster	Priority TTI scores (0-100). Cell values nearest 100 are the highest observed TTI.
Ridership	riderOverall.tif	50m Raster	Priority ridership scores (0-100) derived from NJ-PA composite ridership index. Cell values nearest 100 are the highest observed ridership.

6: Focal means and raster calculations 2

Calculate the overall priority score by implementing the formula below in the QGIS raster calculator. The calculation produces a weighed score without considering ridership, saved as `weightedScore.tif`.

$$\texttt{weightedScore.tif} = \frac{Average\ Speed + On-Time\ Performance + (TTI \times 2)}{4}$$

In QGIS raster calculator:

```
weightedScore.tif = (("tti@1" * 2) + "otp50r@1" + "linkspeed50r@1") / 4
```

Unfortunately, `weightedScore.tif` appears choppy because scores differ dramatically between neighboring cells. This is likely an artefact of the raster overlay of several different datasets—even though the 50m cell serves as a type of buffer between data sources, links and lines still do not overlay perfectly.

To smooth out the choppy appearance of `weightedScore.tif`, we take the focal mean of all cells with nonzero values. The process requires the following steps and is completed in `Export.R`:

1. Read in `weightedScore.tif`.
2. Reclassify cells with zero values to `NA`.
3. Compute the focal mean with a kernel of `3 x 3` of all non-`NA` cells.
4. Reclassify cells with `NA` values back to zero.
5. Export the results as `ws_focal.tif`.

`ws_focal.tif` smooths out the appearance of `weightedScore.tif`, but it also expands the width of all lines. For this reason, a Boolean multiplier raster similar to those in Raster Calculations 1 must be created, and `ws_focal.tif` must be clipped to the extent of `weightedScore.tif` using the QGIS raster calculator. Results are saved as `ws_smoothed.tif`.

In QGIS raster calculator:
```
ws_smoothed.tif = "ws_focal@1" * "ws_multiplier@1"
```

`ws_focal.tif` is also overlaid with ridership in the QGIS raster calculator using the formula below. Results are exported as `riderOverlay.tif`.

$$\texttt{riderOverlay.tif} = \frac{\texttt{weightedScore.tif} \times \textit{Ridership}}{100}$$

In QGIS raster calculator:
```
riderOverlay.tif = "ws_focal@1" * "riderOverall@1"
```

Finally, `ws_smoothed.tif` and `riderOverlay.tif` are prepared for export as shapefiles. While these two files could theoretically comprise scores ranging from `0` to `100`, they do not in practice. The range of these two files is extended in `Export.R` by dividing by each raster's respective maximum value and multiplying by `100`. These rasters are exported as `priority_score.tif` and `priority_score_incl_ridership.tif`.

Then, these two rasters are reclassified into a twenty-class Jenks classification scheme using the `raster` function `cut`. The rasters are transformed into vector polygons, dissolved by class, reprojected into CRS `WGS84` (`EPSG: 4326`), and exported as shapefiles. The shapefile version of `ws_smoothed.tif` and `riderOverlay.tif` are saved as `class20Score_jk.shp` and `class20Rider_jk.shp`, respectively.

7: Output 3

The table below outlines the properties of the output files.

File Name	Format	Description
weightedScore.tif	50m Raster	Priority scores for average speed, OTP, and TTI with possible range 0-100. A cell with a value of 100 would have the lowest observed average speed, lowest observed OTP, and highest observed TTI.
priority_score.tif	50m Raster	Priority scores for average speed, OTP, and TTI stretched to range 0-100.
ws_focal.tif	50m Raster	Focal mean of *weightedScore.tif* with 3 x 3 kernel.
ws_smoothed.tif	50m Raster	Preferred version of *weightedScore.tif*, as this file's values are smoothed by focal mean.
riderOverlay.tif	50m Raster	The overlap of ridership with priority scores for average speed, OTP, and TTI with possible range of 0-100. A cell with a value of 100 would have the lowest observed average speed, lowest observed OTP, highest observed TTI, and highest observed ridership.
priority_score_incl_ridership.tif	50m Raster	The overlap of ridership with priority scores for average speed, OTP, and TTI stretched to range 0-100.
class20Score_jk.shp	Shapefile	Vectorized *ws_smoothed.tif*, dissolved into 20 classes ([0,10],(10,20],…(90,100]).
class20Rider_jk.shp	Shapefile	Vectorized *riderOverlay.tif*, dissolved into 20 classes.