Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 287–293

Procedia Computer Science

www.elsevier.com/locate/procedia

The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018)

The optimization of traffic count locations in multi-modal networks

Thomas Koch[a,b,*], Rob van der Mei[a,b], Elenna Dugundji[a,b]

[a] *Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands*
[b] *Vrije Universiteit, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands*

Abstract

In this paper we will investigate ways to optimize the placement and number of traffic counters used in multi-modal transportation analysis studies for motorized vehicles, bicycles and pedestrians. The goal is to strike a balance between using as few as possible traffic counters for economical efficiency and deploying more counters which could collect more data. By using shortest path algorithms to determine the paths between the centroids of statistical divisions, we derive from origin-destination matrices which traffic is flowing from where to where over which links in a multi-modal network. Using centrality measures such as betweenness, we determine the links in the transportation networks that capture the most useful traffic in terms of as much unique traffic as possible. Next we look at ways to implement additional criteria in the selection of locations: those that are permanently covered, locations that were used for previous studies in prior years for which historical analyses can be made, and locations that capture more than one modality for vehicles, bicycles and pedestrians. Finally we study groups of traffic counters organized in screen-lines.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: traffic counter, screen-line, origin-destination matrix, multi-modal networks, bicycling, pedestrians

1. Introduction

For the city of Amsterdam, we have been investigating methods how to optimize the planning of where to put traffic counters that measure the traffic circulating within the jurisdiction of the city. In prior years, the planning for the placement and number of traffic counters was determined manually based on intuition and operational experience; the city was interested if a data-driven approach could lead to improved deployment of the counters.

An optimal planning for traffic measurement is a balance between deploying as few counters as possible while not significantly decreasing the coverage of travel movements within the city. For example we might prefer to catch less traffic with more measurement points because we prefer to organize the counters within screen-lines. With screen-line analysis all movements with an origin of travel at one side of a screen line and the destination on the other side are intercepted[8]. The benefits of screen-line analysis is that it enables a comparison between the results of traffic

* Corresponding author. Tel.: +31 20-592-4132
E-mail address: koch@cwi.nl

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.041

assignment and traffic count data along a screen-line; by comparing the sum of traffic volumes in the assignment with the traffic counts, the ratio can be computed[1]. Other reasons to prefer certain locations could be policy reasons such as a high number of traffic complaints in a certain area, future developments in a region or certain spots that are already often permanently covered by counters such as tunnels, bridges and intercity highways.

To conduct the research we had access to multiple sources of data, such as origin-destination matrices sourced from Google and the city council, the measurements of the traffic counters in previous years and speed measurements sourced from Google and city traffic control cameras. To model the street network we used geographical material from the national government (NWB), OpenStreetMaps and the GIS data used within the official traffic model of the city of Amsterdam.

2. Related research

A number of studies have been done in this area. In 1998[7] Yang and Zhou derived four rules to locate traffic counting points based on the theory of maximal possible relative error in origin-destination (O-D) matrix estimation: 1) the O-D covering rule, 2) the maximal flow fraction rule, 3) the maximal flow-intercepting rule, and 4) the link independence rule. The theory of maximal possible relative error "represents the maximum possible relative deviation of the estimated O-D matrix from the true one or the upper bound of the real relative error for a particular fitted O-D matrix"[7]. The O-D covering rule states that traffic counters should observe at least a certain portion of traffic between O-D pairs at at least one counter. The maximal flow fraction rule states that for an O-D pair the traffic counters should be placed on links where we find the largest flow fraction between that O-D pair and all flows on that link. The maximum flow-intercepting rule states that under a minimum number of links to be observed, these links should intercept the highest amount of traffic flow possible. The link independence rule states that the traffic counters should be located on the network where the resultant traffic counts on the chosen links are not linearly dependent. Yang and Zhou[7] formulate the problem of locating traffic counters as a mathematical problem, where the O-D covering rule and link independence rule are applied as constraints in the attempt to maximize the total traffic flow observed, solved with a heuristic greedy algorithm.

In 2004 Chootinan, Chen, Yang[4] published a distance-based genetic algorithm. They investigated two methods to solve the problem: a weighed-sum method and a distance-based method. With the weighted-sum method they combine two objectives, covering as much traffic as possible while having the lowest number of counting points, into a single objective. By adjusting the weights different solutions will be found, switching between covering more traffic or having more points. In the distance-based method their main idea is to evolve the genetic search to get closer to Pareto optimal solutions. Their results indicate that the distance-based method is able to provide a better description of the quality-cost trade-off than the weighted-sum method. Additionally it can generate non-dominant solutions in the duality gap that can not be represented in a weighted-based search.

Ehlert, Bell and Grosso[5] build a software application based on the heuristics published by Yang and Zhou[7] and applied it on the network in the district of Gateshead in Northeast England, with 1980 O-D pairs on a network of 240 kilometers of classified roads in 1414 directed links within an area of approximately 142 square kilometers. Additionally[5] Ehlert, Bell and Grosso proposed two extensions to take budget restrictions into account. Their first extension takes into account locations of existing detectors by using the second-best formulation of the optimization problem and then using these locations to define new link choice proportions. Their second extension prioritizes O-D pairs based on the average information content of an O-D movement taken from previously collected O-D matrices.

Yang and Liu[6] proposed an enhanced genetic algorithm to solve two traffic counting location problems. Firstly they address the problem of determining the optimal number of locations of traffic counts that will cover all the O-D pairs in the network. Secondly they determine the maximum number of covered O-D pairs with a defined number of locations. The enhancement of the genetic algorithm is due to the selection, mutation and partial elitism policy leading to better and faster results.

Chen and Pravinvongvuth[3] developed strategies to select additional locations beyond an initial set of already existing traffic counters in order to improve origin-destination trip table estimation. Using these counts the O-D trip table is estimated using a modified flow path estimator that is capable of internally consistent handling of the traffic count. To solve this NP-hard combinatorial problem, they developed another genetic algorithm embedded within a shortest path algorithm.

Barcelo, Guillieron, Linares, Serch, Montero[2] proposes a modified set that formulates the link detection layout with side constraints. Additionally it presents a new meta-heuristic tabu search algorithm with a high computation efficiency. Their solution focuses on sensors that can follow vehicles using the electronic signatures of phones with bluetooth, allowing the placement of sensors at intersections. Specifically Barcelo et al.[2] proposed a new formulation in terms of a node covering problem with side constraints that can be efficiently solved by solving software.

3. Network

The road network of Amsterdam within the boundary of the city contains 18,438 links. The VMA ('Verkeersmodel Amsterdam' / Amsterdam Traffic Model) has 4,418 traffic analysis zones that represent origin and destinations for journeys between Amsterdam and any region up to neighboring countries. The city itself has a ring road A10 and a radial secondary network in the form of S-roads that connect the neighbors to and from the ring-road. A water body (the 'IJ') separates North and South districts of the city, which are only connected via two tunnels on the A10 ring road and two S-road tunnels.

To create a graph to represent the network, you would ideally use geographical material with only one edge between each intersection as multiple edges would dilute the intensity of the traffic. Geographic material from OpenStreetMap often uses 3 separate edges to represent one street with cycle-lanes, motorized vehicle lanes and a tramway in the middle, making it a poor choice to use in our computation.

4. Calculating paths between areas

To gain insight which and how much traffic is flowing from and to where, we calculated the fastest paths between each origin and destination (O-D) pair. To do that we used the open source routing library pgRouting to calculate all the links that are on the fastest path using the A* algorithm. As the cost-function we used the travel-time for each direction on the link.

For each link we store a tuple $(ID_{origin}, ID_{destination}, ID_e dge, forward, backward)$, denoting the identifiers of the origin and destination, the link number and whether the path is following the direction of the 'LineString' geometry on the link.

The full collection of these tuples represents an all-or-nothing assignment than can be used to distribute O-D matrices along the network. Using this collection and an O-D matrix it is possible to determine per link how much traffic flows along both directions of the link. Additionally it is possible to calculate the traffic between a subset of the origin-destination pairs along a link, for example to exclude traffic already captured along other links.

4.1. Sorting links by amount of traffic

Using the traffic assignment and an O-D matrix it is possible to sort links by the amount of traffic flowing over them. This is a centrality measure called betweenness, the ratio of how often a vertex or link is part of a shortest path between one of the other nodes in the graph. This is given by $\sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} references the number of shortest paths from s to t , and $\sigma_{st}(v)$ the number of shortest paths from s to t that pass through v.

To avoid catching the same traffic at the links more than once, only traffic between O-D pairs not covered yet is counted. To do this it is necessary to keep a record of which O-D pairs are already covered.
By using the PostgreSQL database server, we keep the table with the route assignment and a table with the O-D matrix. To query the link with the highest number of journeys, we join these two tables, and pick the link with the highest aggregated count journeys per edge. To exclude traffic already captured in this query, we keep a third table with origin-destination pairs already covered. The moment we pick a link, we insert all origin-destinations that travel over that link. We continue this process until all origin-destination pairs are separated by at least one link.

The process can be customized, for any reason such as links that are permanently covered or areas that require additional scrutiny. By initializing the process with pre-defined links, the algorithm will seek an optimal solution with the remaining links.

4.2. Sorting screen-lines by amount of traffic

An alternative strategy instead of placing traffic counters on links to capture the highest share of traffic might deliver more useful data. For example it can also be worthwhile to organize the counters within screen-lines. Screen-lines are groups of segments separating traffic between the origin and the destination. Screen-lines can help the validation of Origin-Destination matrix estimations using traffic counts. In our experiment we used previously determined screen-lines that were used in earlier transportation studies by the city. Using the same screen-lines allowed for a better comparison with the data collected in previous years.

To sort screen-lines by the amount of unique traffic we can apply a similar process. For each screen-line, we sum the weight for each O-D pair that covers more than one link that is part of the screen-line. To exclude traffic already covered, we exclude origin-destination pairs separated by at least one link.

In order to weight screen-lines, we use two criteria: the amount of traffic covered and the number of counters within a screen-lines. With a simple weighting function: $intensity/n_{points}$ we observed a disproportionate avoidance of screen-lines with more traffic-counters. As a screen line with more counters results in more useful data than a screen-line with only two counters, we used instead the intensity per eighth-root of the number of counters: $intensity/\sqrt[8]{n_{points}}$. This weighting function was used to pick the best screen-line each time and then look at subsequent screen-lines based on traffic not yet covered.

Table 1: Amsterdam screen-lines sorted by unique traffic coverage

screenline	Journeys covered	$n_{counters}$	$Coverage/n_{point}$	$Coverage/\sqrt[8]{n_{point}}$
Zuidas	169247647	4	42311911	142319739.652
Zuidoost	158309305	6	26384884	126542828.021
Amstel	135445001	5	27089000	110762240.020
Riekerpolder	113804400	2	56902200	104359094.934
Schinkel	104172820	4	26043205	87598550.905
Noord-kanaal-oost	99599079	4	24899769	83752508.494
Amstelveen	86381721	5	17276344	70639985.560
Westpoort/Haarlemmerweg	76489739	4	19122434	64319947.329
West	72324022	4	18081005	60817010.837
Singel	82659889	13	6358453	59986275.630
Noord-vanaf ring noordwest	54713890	3	18237963	47693306.907
Vondelpark	31482064	3	10494021	27442460.049
Gooiseweg	27784370	2	13892185	25478379.628
West(noord/zuid)	21690791	2	10845395	19890543.047
Boerenwetering	23159533	5	4631906	18939065.554
Centrum	15916517	6	2652752	12722695.444
Stadhouderskade	4391555	5	878311	3591261.880

4.3. Cycling and walking modalities

Besides cars and other vehicle traffic, the city of Amsterdam also deploys sensors to count the number of cyclists and pedestrians. Since the city itself has no complete O-D matrix for these modalities we had to use different sources. For cyclists we had access to data collected by cyclists who volunteered to log their activities for a week with an application that uses the GPS and sensors on smart-phones. Additionally we had access to an O-D matrix supplied by Google, which contained the movements between areas of pedestrians, cyclists and vehicle-passengers in June of 2016. As geographical material we used OpenStreetMap which has a complete coverage for the cycleways in Amsterdam.

The difference between fast-traffic and slow traffic is that journeys are shorter on average and that there are much more possible paths. This results in less prominent locations to place counters. Since there is an economical advantage to also use the counting locations for bicycle where possible, we used the top-100 locations as the starting point for this computation. Each segment allowing bicyclists parallel to the location covered by a car counting location was selected as a counting location for bicyclists. The remaining locations were then primarily segments that only allowed slow traffic.

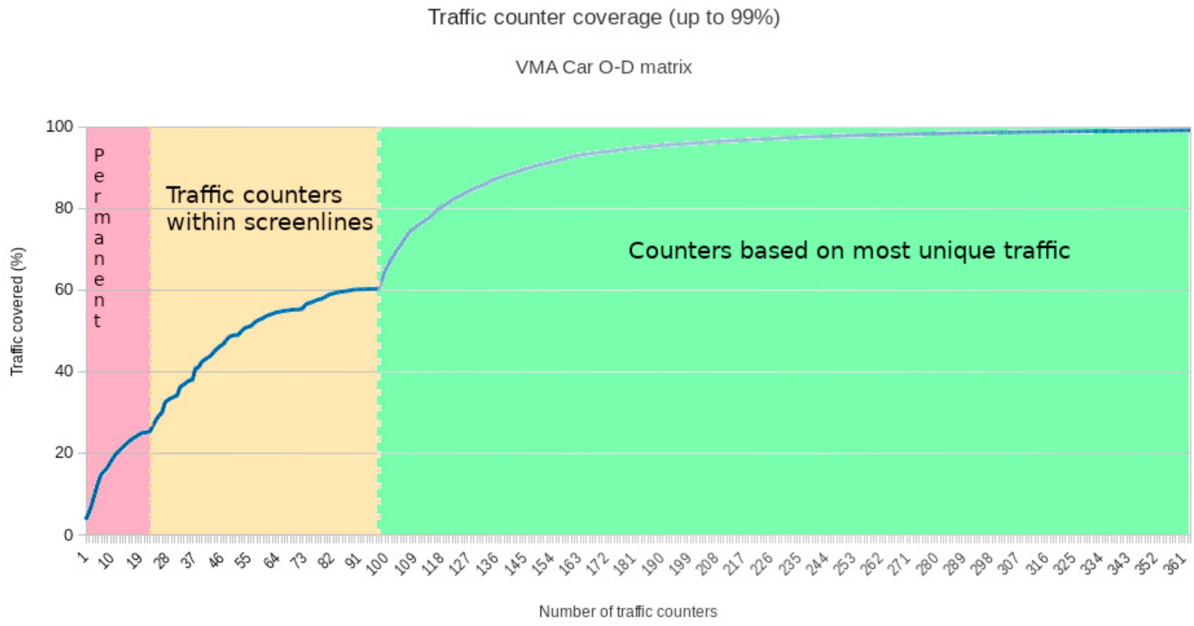

Fig. 1: Chart of unique traffic of cars covered. The first section includes coverage of all permanently covered locations and the second section is grouped within screen-lines. Note the step-like behavior. Each counter within each screenline is included in the order of the most unique traffic per counter per screenline. The third section contains the most remaining unique traffic.

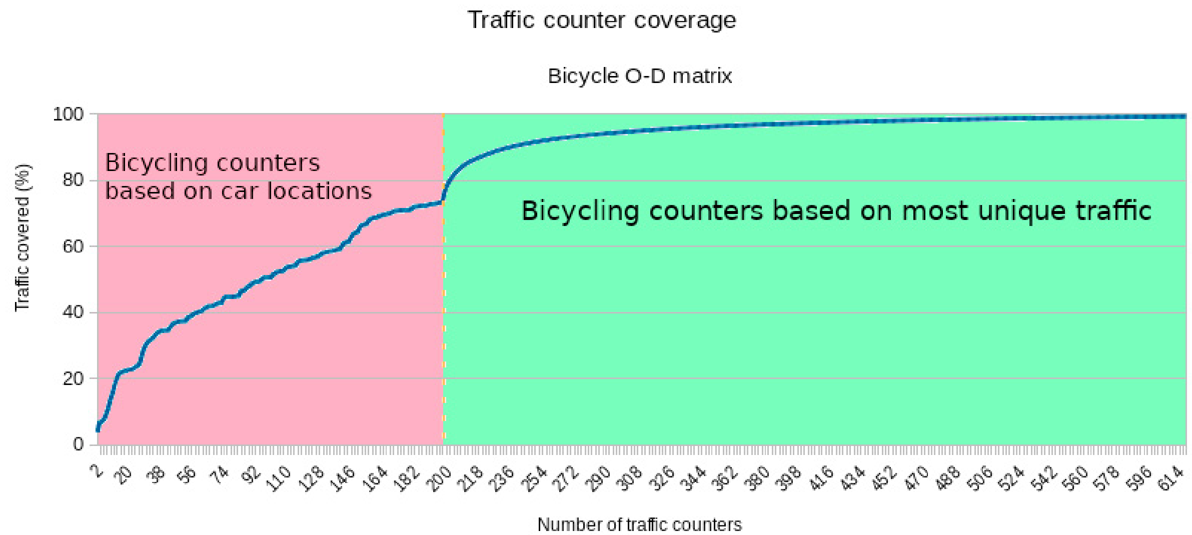

Fig. 2: Chart of unique bicycle traffic covered. The first 200 points are based on the top 75 locations with the most unique car traffic. The difference in points is due to the fact that for proper coverage we include all links parallel to the car locations.

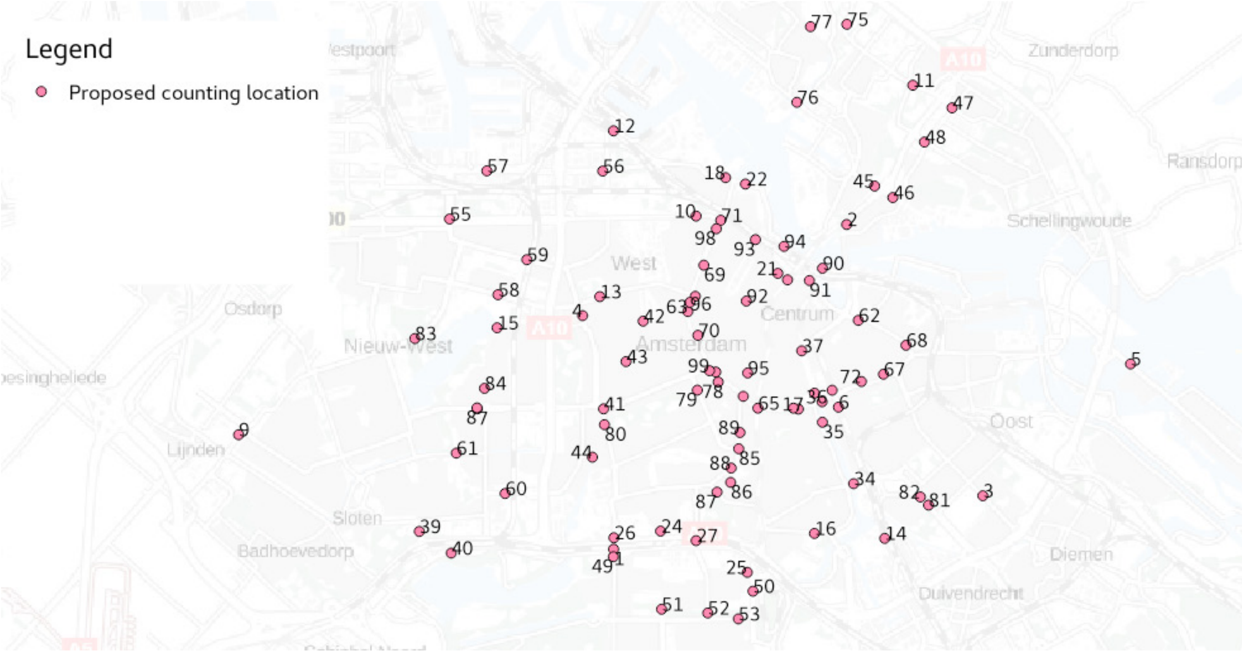

Fig. 3: Map with visualization of the top 100 traffic counting locations capturing the highest number of unique traffic. Each digit indicates the priority, firstly we prioritize all locations that are permanently covered, then the locations within screen-lines and finally a free-selection. Each point within a group is ordered by the number of unique traffic it captures.

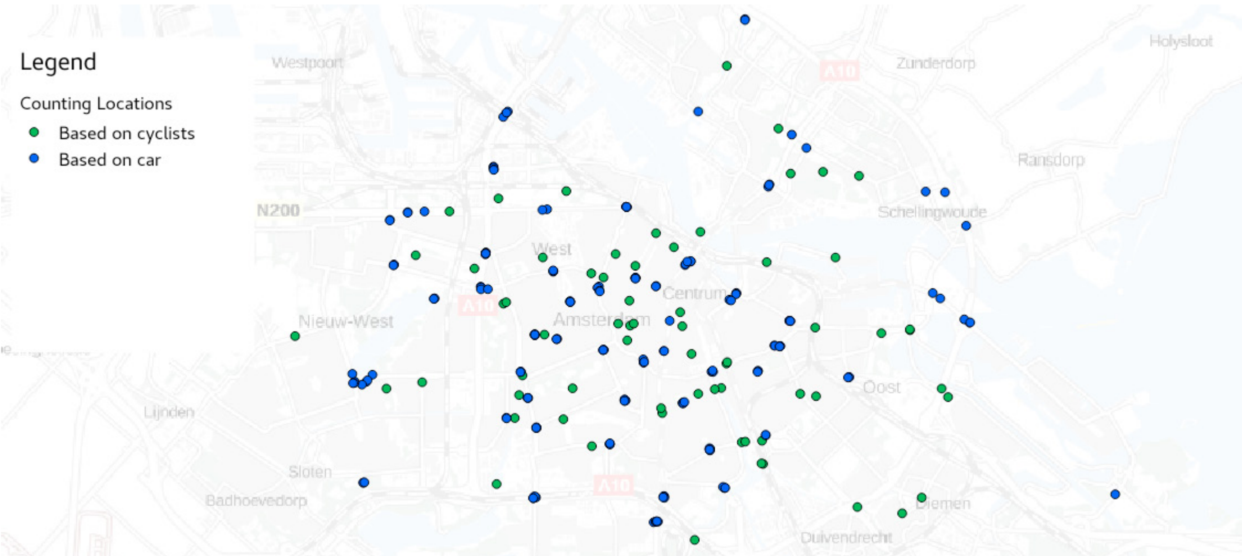

Fig. 4: Map with the resulting counting locations for bicycles. The green points mark locations added for just bicycling; the blue points indicate that they were chosen based on car traffic patterns.

4.4. Results

In the first step to obtain the best locations, we calculated the coverage based on 22 links in the network at locations such as tunnels that are permanently covered by loop counters; these 22 links are sorted based in descending order on the most remaining unique traffic captured. The next 77 edges were organized in 6 screen-lines. Originally the city of Amsterdam proposed 17 possible screen-lines, which were then sorted using the algorithm described; the results of this process are described in table 1. From this list the city selected the ten screen-lines covering the most traffic. The 24th to 54th counting point are selected within these screen-lines in a descending order of most unique traffic. The remaining counting points were selected using the unique traffic remaining.

The top 100 locations for cars were used in our calculations for bicyclists; this resulted in 197 ”locations” since OpenStreetMap had many segments parallel to each selected link in the car network.

5. Conclusions

Our idea to assign traffic from O-D matrices to a network to determine locations that capture the most unique traffic in descending order worked well. The flexibility we have in selecting locations allowed us to respond quickly to additional requirements that came later on in this study. Requirements such as organization of counting locations within screen-lines was solved by determining how much traffic each screen-line captures with taking into effect of screen-lines close to each other. Since any point can be arbitrarily inserted into the process, allowed to take into account the preference for historical locations for trend-lines and the preference to combine cycle with car locations all came in the end phase of this project.

For future research it would be interesting to also find the optimal screen-lines, or a weighting function to take other factors into account besides capturing more traffic. In this study we used the screen-lines that were used in previous traffic studies since they came with the added benefit of providing new data on historical trends, but it should be possible to find screen-lines that separate more traffic with fewer traffic counters, since we know for each segment where each vehicle is coming from and is heading to.

Acknowledgements

We would like to acknowledge the City of Amsterdam for their support, with special thanks to Bas Bussink, Eric de Kievit, Laurens Peijs and Stefan Verduin.

References

1. What is screenline analysis? URL `http://www.caliper.com/glossary/what-is-screenline-analysis.htm`. Accessed: 2017-07-05.
2. J Barceló, Fanny Gilliéron, M Linares, O Serch, and L Montero. Exploring link covering and node covering formulations of detection layout problem. *Transportation Research Record: Journal of the Transportation Research Board*, (2308):17–26, 2012.
3. Anthony Chen, Surachet Pravinvongvuth, Piya Chootinan, Ming Lee, and Will Recker. Strategies for selecting additional traffic counts for improving od trip table estimation. *Transportmetrica*, 3(3):191–211, 2007.
4. Piya Chootinan, Anthony Chen, and Hai Yang. A bi-objective traffic counting location problem for origin-destination trip table estimation. *Transportmetrica*, 1(1):65–80, 2005.
5. Anett Ehlert, Michael GH Bell, and Sergio Grosso. The optimisation of traffic count locations in road networks. *Transportation Research Part B: Methodological*, 40(6):460–479, 2006.
6. Chao Yang and Qi Liu. Enhanced genetic algorithm for selecting traffic counting location. In *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, pages 1–6. IEEE, 2009.
7. Hai Yang and Jing Zhou. Optimal traffic counting locations for origin–destination matrix estimation. *Transportation Research Part B: Methodological*, 32(2):109–126, 1998.
8. Hai Yang, Chao Yang, and Liping Gan. Models and algorithms for the screen line-based traffic-counting location problems. *Computers & Operations Research*, 33(3):836–858, 2006.