The origin-destination (OD) matrix is the important input in many transportation planning problems. Recently various traffic simulation softwares has become available, which made it possible to perform the assignment of OD matrix to transportation network in more efficient way. Use of simulation brought up a need for efficient simulation optimization techniques, that can be applied to the OD matrix estimation problem. Recent research showed some inspiring results on the performance of Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm. The aim of this research is to justify the use of SPSA for OD estimation, test the performance of the algorithm and develop the modification that can improve it.

1 Introduction

The origin-destination (OD) matrix is the important input in many transportation analysis problems. It describes the demand for the transportation network in terms of vehicle flows from all possible origins to all possible destinations (OD pairs) in the network.

The direct information on the OD flows can be gathered though household sample surveys, though it is very expensive, and often not very efficient.

Other sources of information, so called indirect measures, are used to obtain more exact estimates on the real OD flows. Commonly, the information used to estimate OD flows is collected through sensors. They can record number of passing vehicles, their speeds or travel time between two sensor locations. Most of the research is based on only flow measurements for OD estimation. While for uncongested networks it can be enough for accurate OD estimation to use just traffic counts, in the congested case it is preferable to use other available information on the traffic conditions.

More and more often simulation is used as a part of the OD estimation process. Simulation software is used to map OD flows to the network. While it helps in creating some realistic assignment of flows to the network, it complicates the estimation process since simulation includes stochastic elements. Moreover, each simulation run can be computationally expensive, and use of algorithms which require many simulation runs can be unfeasible it terms of time.

The objective of the research is to develop methods, that are practical, computationally efficient so that they can be used for large networks and for a variety of
applications (both off-line and real time), and general so that they can be utilized with a number of existing simulation and other tools. Furthermore, the developed methods will be flexible in the use of general traffic data (ranging from counts and spot speeds to travel times). The research will examine methods which have the potential to work even under congested and oversaturated conditions.

2 Problem Description

2.1 Network description

Consider the transportation network, which can be described by graph \(G = (N, L)\), where \(N\) is a set of nodes in the network, and \(L\) is a set of directed links that connect nodes \(N\).

\[ G = (N, L) - \text{network}, \]

\[ N - \text{nodes}, \]

\[ L - \text{directed links}. \]

Consider set of origins \(O\) and set of destinations \(D\), \(O \subseteq N, D \subseteq N\). Some nodes can be both origins and destinations, and therefore belong to both sets. It might be possible that not every origin is connected (or can be paired) with every destination.

Some links in the network are equipped with sensors that collect the data on passing vehicles. We define such subset of links as \(\tilde{L} \subseteq L\). Most commonly these sensors collect the data on passing number of vehicles and their speeds.

Now additionally to the network described above, \(R = \{r\}\) will present the set of departure time periods for OD flows (during which the departures are counted), and \(T = \{t\}\) will present the measurement periods (during which measurements are collected).

For more general formulation we also assume that possible combination of origins and destination may change from one time period to another. Then the set of all possible OD pairs will be defined as following:

\[ N_{OD} = \{(o, d, r), o \in O, d \in D, r \in R\} \]

Similarly, information from sensors might be collected at different time periods, and measurements set will become:

\[ L' = \{(l, t), l \in \tilde{L}, t \in T\} \]

2.2 Parameters description

The counts collected from the sensors will be presented by vector of the size same as the size of the subset \(L'\):

\[ \tilde{f} = (\tilde{f}_l, l \in L') \]

Same is for speeds:

\[ \tilde{s} = (\tilde{s}_l, l \in L') \]
Also some historical estimation of OD flows will be available in following form:

\[ x^H = (x_n^H, n \in N_{OD}) \]

For modelling purposes we also need to define parameters, which will present reliability of available data, and also take into account some other information, which is relevant for each particular approach:

- \( \omega_{OD} \) – weight related to OD matrix
- \( \omega_f \) – weight related to counts
- \( \omega_s \) – weight related to speeds.

### 2.3 Variables description

3 types of variables are estimated during OD estimation procedure:

- \( x = (x_n, n \in N_{OD}) \) - vector of estimated flows between OD pairs \( N_{OD} \),
- \( f = (f_l, l \in L') \) - vector of estimated counts on the links \( L' \) produced by estimated OD flows,
- \( s = (s_l, l \in L') \) - vector of estimated speeds on the links \( L' \) produced by estimated OD flows.

### 2.4 Formulation

So the general formulation of the OD estimation problem will be the following:

\[
\min_{x \geq 0} (\omega_{OD}F(x, x^H) + \omega_f F(f, \tilde{f}) + \omega_s F(s, \tilde{s}))
\]

Subject to

\[
(f, s) = assign(x)
\]

where \( F(\cdot, \cdot) \) is a function, which commonly represents a measure applicable for a vector space.

### 3 Simulation Optimization

One of the approaches to solve simulation optimization problem is Simultaneous Perturbation Stochastic Approximation (SPSA) method, proposed by Spall (1998) [4, 5]. The SPSA algorithm is an optimization algorithm based on an approximation of the gradient of the objective function that requires only two function evaluations.

SPSA allows to solve the problem in following form:

\[
\min_{\theta \in \Theta} z(\theta)
\]

where \( \theta \) is a \( K \) – dimensional parameter vector to be calibrated, \( \Theta \) is a set of possible values for parameter \( \theta \). The estimate of the gradient is used to determine the parameter update.
\[ \theta^{i+1} = \theta^i - a^i \cdot \hat{g}(\theta^i) \]  

where \( \hat{g}(\theta^i) \) is a current gradient estimation, which is calculated as

\[
\hat{g}(\theta^i) = \frac{z(\theta^i + c^i \Delta_i) + z(\theta^i - c^i \Delta_i)}{2c^2} \cdot \begin{bmatrix} \Delta_{i1}^{-1} \\ \vdots \\ \Delta_{iK}^{-1} \end{bmatrix}
\]

\[ (5) \]

\( a^i \) and \( c^i \) are the parameters defining the gain sequence of step size and perturbation size respectively.

SPSA was applied to offline calibration of DTA models by Balakrishna (2005) [3], and it was shown that while on the small network SPSA algorithms showed comparable results with other algorithm, on the larger networks SPSA required much less computational time.

Another research, performed by Ma et al. (2007) [2], also compared SPSA to other heuristic optimization methods, and showed also similar results: SPSA, compared to genetic algorithm, can generally obtain an acceptable set of parameters in much less time.

4 Implementation and Results

SPSA algorithm has been implemented and tested on a small network with 40 nodes, 136 links (20 links with sensors), 219 OD pairs, and one time period. Function \( F(\cdot, \cdot) \) was set to be square error and the assignment of OD matrix to the network is performed by the simulation software Mezzo [1].

The tests has demonstrated that algorithm is very sensitive to the choice of parameters, such as gain sequences of step size and perturbation size.

References


