SOLVING LOCATION PROBLEM FOR VEHICLE IDENTIFICATION SENSORS TO OBSERVE AND ESTIMATE PATH FLOWS IN LARGE-SCALE NETWORKS

Pegah T. Yazdi and Yousef Shafahi
Department of Civil engineering
Sharif University of Technology
Azadi St., Tehran 11365-9313, Iran
E-mail: shafahi@sharif.edu

KEYWORDS
Location Problem, Vehicle Identification Sensor, Large-Scale Networks, Heuristic and Meta-Heuristic

ABSTRACT

Origin-Destination (OD) demand is one of the important requirements in transportation planning. Estimating OD demand could be an expensive and time consuming procedure. These days using vehicle identification sensors for OD estimation has become very common because of its low cost and high accuracy. In this paper, we focus on solving two location problems of these sensors: one to observe and one to estimate path flows. These problems have only been solved for small-scale networks until recently due to being computationally expensive. Therefore, we try to present a method to solve these models for large-scale networks. Due to resemblance of these models and set covering problem, we used heuristic and meta-heuristic methods based on set covering problem. For this purpose, we defined our new set covering matrix based on prime matrix. In order to determine which method is more appropriate, we chose a large-scale and six medium-scale networks. The results represent that through heuristic methods and meta-heuristic methods a greedy algorithm and a Tabu search are more appropriate respectively.

INTRODUCTION

One of the most important requirements in transportation planning is demand data of using networks. OD and path flows estimation is one of the approaches in this regard. Common methods of these estimation are in the form of household surveys, on-board vehicle surveys, using trip distribution, assignment models etc. which are mainly economically expensive and not always reliable. Therefore, these days using output information from ITS is also taken into consideration. This information can estimate OD and path flows with high accuracy and low cost. One type of ITS that is used for this purpose is the use of sensors. However due to high price of these sensors, solving location problem and reducing their number in network is very important.

Gentili and Mirchandani (2012) represented two major categories for sensor location models:

- Flow-observability model: These models locate and determine the optimum number of sensors in the network so that the target flows can be obtained uniquely by solving a linear system of equations associated with the optimum number of located sensors.

- Flow-estimation model: These models locate and determine limited number of sensors in the network due to budget constraints so that the best estimation of target flows can be obtained.

Here, target flow is a term for three type of flows: OD flow, path flow and arc flow. Also four categories of sensor types have been represented: counting sensors, path-ID sensors, image sensors and vehicle identification sensors. Gentili and Mirchandani (2012) In this paper, we focused on path flow observability and estimation based on vehicle identification sensors.

Through literature review, two location models have been selected for studying in this paper. First, a vehicle identification sensor location model for path flow observability represented by Castillo et al. (2008). This model finds the minimum number of arcs to locate vehicle identification sensors so that path flows can be observed uniquely. Second, a vehicle identification sensor location model for path flow estimation represented by Minguez et al. (2010). This model finds limited number of arcs for locating vehicle identification sensors (because of budget constraints) so that the quality of path flow estimation can be maximized with limited number of sensors. In both models the assumption is that the arc number where vehicle has been seen and the vehicle unique ID, from extracted information of vehicle identification sensors is available.

Up to now, these location problems were only solved for small-scale networks. But in practice it is in the large-scale networks that location and installation of sensors are more necessary. Due to volume of calculations and lack of solution we have to present a method to solve these models for large-scale networks.

Proceedings 32nd European Conference on Modelling and Simulation ©ECMS Lars Nolle, Alexandra Burger, Christoph Tholen, Jens Werner, Jens Wellhausen (Editors)
ISBN: 978-0-9932440-6-3/ ISBN: 978-0-9932440-7-0 (CD)

Literature review shows that Castillo et al. (2008)'s model is an instance of set covering problem. Due to resemblance of our two models we used set covering solutions for both of them. For this purpose, we adjust our model with set covering matrix. Finally, we solve our models for six medium-scale networks and a large-scale network. Our results present a method for solving location models for large-scale networks.

The observability model

In this model, Castillo et al. (2008) allocated the minimum number of vehicle identification sensors in the network required for unique observation of all path flows. For this purpose, they appointed a set of arcs to a vehicle. Each set shows the arcs which vehicle has been seen in them. Thus, The path that each vehicle has taken becomes known. They formulated the model as follows:

A: Minimize (1) $N_s = \sum_{a \in A} J_a$

Subject to: (2) $J_a \in \{0,1\}$, $\forall a \in A$

(3) $\sum_{a \in A} J_a \cdot d(r_1, r_2, a) \geq 1$, $\forall r_1, r_2 \in R | r_1 \neq r_2$

(4) $\sum_{a \in A} J_a \cdot \delta_{ra} \geq 1$, $\forall r \in R$

Where J_a is a binary variable and it takes value (of) "1" if a sensor is located on link "a" and it's "0" otherwise. A is the set of all links and "R" is the set of all paths of the network. Parameter δ_{ra} equals "1" if link "a" exists in path "r" otherwise it equals "0". $d(r_1, r_2, a)$ is a binary parameter too. It takes value of "1" if link "a" exists in only one of the paths "r_1" and "r_2" and it's "0" otherwise.

In this model, with the objective function (1) of minimizing number of sensors which should install in the network, constraint (4) ensures that each path gets at least one sensor and constraint (3) guarantees that there is at least one uncommon link between every tow paths that get a sensor. With these two constraints we ensure that path flow could be observed uniquely.

The estimation model

Minguez et al. (2010) developed an estimation model based on previous model. Due to consideration of budget constraints, this model is more practical. Their model with the assumption that no prior OD matrix information is available, was written as follows:

B: Maximize (5) $N_r = \sum_{r \in R} x_r$

Subject to: (6) $J_a \in \{0,1\}$, $\forall a \in A$

(7) $x_r \in \{0,1\}$, $\forall r \in R$

(8) $\sum_{a \in A} J_a \cdot d(r, r_1, a) \geq x_r$, $\forall r, r_1 \in R | r \neq r_1$

(9) $\sum_{a \in A} J_a \cdot \delta_{ra} \geq x_r$, $\forall r \in R$

(10) $\sum_{a \in A} C \cdot J_a \leq B$, $\forall r \in R$

Where C is cost of buying and installing one sensor and B is available financial resource. x_r is a binary variable and when flow of path "r" could be observed it is equal to "1", otherwise it is "0".

In this model, with the objective function (5) of maximizing number of paths that their flow could be observed uniquely, constraint (10) controls the number of sensors that could be afforded with available budget. Other constraints and parameters are same as Castillo et al. (2008)'s model but the right part of the constraints (8) and (9) are x_r. It means that this model tries to maximize the number of paths that get at least one sensor and have at least one uncommon sensored arc with each other paths (flow observable path).

Set covering problem

Zangui et al. (2015) proved that model A is an instance of set covering problem. Thus, solving model A with the solutions of such problem can represent acceptable answers. Due to resemblance of model A and B we use same solutions for model B.
Set covering problem is a NP-complete problem of covering a $m \times n$ zero-one matrix by selecting columns at minimum cost. Let $A = (a_{ij})$ be a $m \times n$ zero-one matrix with $M = \{1.2. \dots . m\} . N = \{1.2. \dots . n\}$ and $C = (c_1 . c_2 . \dots . c_n)$ denoting respectively the sets of rows and columns and cost of each column of A. This is said that column $j \in N$ covers row $i \in M$ if $a_{ij} = 1$. The target in this problem is to find a set of columns $(S \subseteq N)$ with the minimum cost where each row $i \in M$ is covered by at least one column $j \in S$. The classical mathematical formulation for Set covering problem is:

Minimize (11) $V = \sum_{j \in N} c_j x_j$

Subject to: (12) $\sum_{j \in N} a_{ij} . x_j \geq 1$, $\forall i \in M$

(13) $x_j \in \{0,1\}$, $\forall j \in N$

Where x_j is the binary decision variable whose value is "1" if column j is in the solution(S) and "0" otherwise. Constraint (12) ensures that every row is covered by at least one column and constraint (13) is written for definition of x_j.

As previously mentioned $C = (c_j)$ is a vector with positive value as the cost of columns. In a version of set covering problem known as unicost set covering problem all c_j for each column j ∈ N are the same. Thus, we can put the value "1" instead of c_j in objective function. In this version, the target is to find the minimum number of columns where each row is covered by at least one column. It could be demonstrated that model A is an unicost set covering problem.

We use four algorithms to solve model A and B:

- Greedy Algorithm – Heuristic method (Johnson 1974)
- Greedy algorithm (four selected row knowledge function)– Heuristic method (Vasco et al. 2016)
  (1) $1/\sqrt{L_i}$ (L_i defined as sum of each row)
  (2) $1/\sqrt{(L_i - 1)}$
  (3) $1/\sqrt{(L_i)^2}$
  (4) $1/\sqrt{(L_i - 1)^2}$
- Tabu Search Algorithm – Meta-Heuristic method (Cerrone et al. 2015)
- Meta-RAPS Algorithm – Meta-Heuristic method (Lan et al. 2007)

We chose two heuristic and two meta-heuristic methods. Heuristics for lower run time and meta-heuristic for better answers. Johnson (1974)'s greedy algorithm is the basis for most of algorithms after that. Vasco (2016)'s greedy algorithm considers rows priority in selecting best column in each iteration. Tabu search and Meta-RAPS algorithms both are based on Johnson (1974)'s greedy algorithm.
We chose Johnson (1974)'s greedy algorithm to compare our methods with the basis. And we chose Vasco (2016)'s greedy algorithm because in our problem number of rows is more than columns and it seemed that considering rows priority could help reaching better answers. Also reason of choosing Cerrone et al. (2015)'s Tabu search was using it in solving some similar models by Cerrone et al. (2015). We chose Meta-RAPS algorithm because of high randomness in choosing columns to see the effect of that in this type of models.
Tabu search and Meta-RAPS algorithms have some parameter which need to be determined. We determined these parameter by examining different values of parameters for solving our medium-scaled networks which their exact answers are in hand. The value of parameters which reached the closest to the exact answers had been chosen.

DEFINING OUR NEW SET COVERING MATRIX

In this section, we describe how the models adjust with the set covering problem. Firstly, we should define our rows, columns and elements of set covering problem matrix. Then determining the problem for each model is necessary.

We set our columns as arcs of the network that we should select them as the place of installation of each sensor. Every row of set covering matrix stands for one of these two subjects: paths of network and couple-paths of network. The definition of elements which are equal to 1 is different for rows that stand for paths of network and rows that stand for couple-paths of network. The element a_{ij} for paths of network equals 1 if arc j would exist in the related row and the element a_{ij} for couple-paths of network equals 1 if arc j would exist in just one of the related couple-rows.

Let assume that $R_1 = (a_1, a_4)$ and $R_2 = (a_1, a_3)$ are two paths of a network and $A = (a_1, a_2, a_3, a_4)$ is the set of all arcs in network. Set covering matrix of this network is written as below:

	a_1	a_2	a_3	a_4
R_1	1	0	0	1
R_2	1	0	1	0
R_1-R_2	0	0	1	1

Figure 1 An Example of a Set Covering Matrix

Observability models should find the minimum number of arcs for installation of sensors, therefore the target in solving this problem is to find the minimum number of columns that cover all rows.

Estimation models should find exact number of arcs for installation of sensors with which target flows could be estimated with the highest quality. The definition of quality in model B is the number of paths that their flow can be observed uniquely. We changed this definition with the aim of simplification in order to solve this model with set covering solutions. We define the quality of flow estimation as the number of paths that at least have one sensor plus the number of couple paths that have at least one uncommon sensor. The objective function of this new definition has been written as below:

$$\text{Maximize:} \quad N_{rr'} = \sum_{r,r' \in R} x_{rr'}$$

Therefore, the target in solving this problem is to find exact number of columns that cover maximum number of rows.

PROBLEM REDUCTION

As previously mentioned, the goal of this paper is to solve models for large-scale networks. Due to volume of calculations, we looked for approaches to reduce this volume and time of calculations. There are various approaches that reduce size of set covering matrix. These reduction approaches are either related to the columns (variables) or to the rows (constraints). As we saw in our models, the number of constraints is larger than the number of variables, thus we choose row reduction to diminish set covering matrix.

Beasley (1987) reviewed the literature and presented an effective row reduction procedure. In this procedure redundant rows are being deleted. Beasley defines a redundant row as: "If there is a subset for row i in set covering matrix, row i is redundant."

It is worth mentioning that we can't use row reduction for our estimation model (model B with new objective function) because the target is covering maximum number of rows so existence of every row would be necessary.

RESULTS

To determine how well our selected set covering solutions is doing, we choose six path collections of Sioux Falls medium-scale networks and one path collection of Mashhad's large-scale network. The characteristics of these networks and their set covering matrix are given in table 1. Percent of remaining rows show that row reduction for this problem could delete 85% to 99% of set covering matrix rows.

On each instance, we compare the results returned by Johnson, Vasco et.al (four different row knowledge function), Cerrone et al.(2015)'s Tabu search and Meta-RAPS algorithms for each of our two problems. All algorithms were coded in MATLAB on a 2.5 GHz Intel i5 processor and 6.00 GB RAM.

Table 1: Characteristics of Instances

Network	nodes	OD	arcs	paths	Paths + Couple-Paths (rows)	Rows after reduction	Percent of remaining rows
Sioux Falls 1	24	12	76	140	9'870	1'490	15.963 %
Sioux Falls 2	24	12	76	225	25'425	3'282	12.908 %
Sioux Falls 3	24	12	76	305	46'665	4'269	9.148 %
Sioux Falls 4	24	30	76	245	30'135	91	0.302 %
Sioux Falls 5	24	30	76	448	100'576	216	0.215 %
Sioux Falls 6	24	30	76	837	350'703	399	0.114 %
Mashhad	917	7'157	2'092	44'210	977'284'155	1'468'803	0.150 %

Table 2 report the results of solving model A with four selected algorithms. Values in this table are values of objective function (number of necessary sensors to observe all path flows in network). The exact answers for Sioux falls networks were obtained by GAMS. As it's clear in table 2, Tabu search could reach to better answers in 100% of instances in average of 32.47 seconds for Sioux falls networks and 1'690'824 seconds for Mashhad. Vasco4 (row knowledge function: $1/\sqrt{(L_i - 1)^2}$) is the best algorithm among heuristics and it reached better answers in 16'765.23 seconds for our large-scale network.

Table 2: The Value of Objective Function after Solving Model A

Network	Exact answer	Johnson	Vasco1	Vasco2	Vasco3	Vasco4	Tabu search	Meta-RAPS
Sioux Falls 1	24	27	27	28	26	25	24	24
Sioux Falls 2	26	31	30	29	27	29	26	26
Sioux Falls 3	27	31	31	30	29	29	27	27
Sioux Falls 4	31	31	31	31	31	31	31	31
Sioux Falls 5	35	36	36	36	35	36	35	35
Sioux Falls 6	35	39	38	38	35	36	35	35
Mashhad	UK	679	678	671	668	659	651	655

Table 3-5 show the results of solving model B with three levels of budget constraint. We define budget constraint as a percent of observability model's answer (exact number of sensors). For number of sensors we chose 25%, 50% and 75% of exact answers for Sioux falls and 25%, 50% and 75% of Tabu search's answer for Mashhad. Values in these tables are number of uncovered rows, meaning number of rows minus value of objective function.

Thus, bigger values of objective function cause lower number of uncovered rows. As previous, Tabu search could reach to better answers in 100% of instances in three levels of budget constraint. But among heuristic methods that their run time is fewer, Vasco1 (row knowledge function: $1/\sqrt{L_i}$) could reach to better answers.

Table 3: Number of Uncovered Rows after Solving Model B with New Objective Function (Budget Constraint: 25%)

Network	Johnson	Vasco1	Vasco2	Vasco3	Vasco4	Tabu search	Meta-RAPS
Sioux Falls 1	912	859	859	1'034	1'159	841	841
Sioux Falls 2	1'443	1'482	1'482	1'833	2'246	1'384	1'384
Sioux Falls 3	2'127	2'115	2'115	2'468	3'082	2'115	2'115
Sioux Falls 4	2'029	2'029	3'289	2'639	3'289	2'023	2'023
Sioux Falls 5	4'580	4'795	8'897	7'212	8'897	4'432	4'555
Sioux Falls 6	11'774	11'784	24'912	15'104	24'912	10'891	10'891
Mashhad	471'907	387'980	951'983	464'745	977'329	329'500	342'074

Table 4: Number of Uncovered Rows after Solving Model B with New Objective Function (Budget Constraint: 50%)

Network	Johnson	Vasco1	Vasco2	Vasco3	Vasco4	Tabu search	Meta-RAPS
Sioux Falls 1	125	116	116	158	158	107	107
Sioux Falls 2	253	242	242	291	286	214	221
Sioux Falls 3	364	401	389	511	413	294	320
Sioux Falls 4	240	251	317	317	317	216	233
Sioux Falls 5	437	481	724	735	749	414	422
Sioux Falls 6	1'073	981	1'742	1'462	1'911	909	971
Mashhad	32'000	25'596	32'533	36'321	45'660	20'287	21'895

Table 5: Number of Uncovered Rows after Solving Model B with New Objective Function (Budget Constraint: 75%)

Network	Johnson	Vasco1	Vasco2	Vasco3	Vasco4	Tabu search	Meta-RAPS
Sioux Falls 1	24	31	25	33	30	17	20
Sioux Falls 2	43	39	39	45	45	25	30
Sioux Falls 3	62	57	61	60	77	36	36
Sioux Falls 4	50	43	36	50	66	31	31
Sioux Falls 5	71	90	67	62	72	52	60
Sioux Falls 6	103	124	139	175	192	97	103
Mashhad	1'596	1'592	1'535	3'006	2'868	1'026	1'339

CONCLUSION

In this paper, we adjusted two allocation models with set covering problem in order to solve our models for large-scale networks. Two heuristic algorithms from Johnson (1973) and Vasco et al. (2016) and two meta-heuristic algorithms from Cerrone et al. (2015) and Lan et al. (2007) were used. The results of solving the models with these algorithms in the medium-scale and large-scale networks show that Tabu search algorithm which was presented by Cerrone et al. is better than the others. Even though the calculation time of location models are not an important constraint (because these models are only solved once), Tabu search's run-time is much higher than heuristic algorithms. Nevertheless, the greedy algorithm of Vasco et al. (2016) solves our models in shorter time and it's more suited for heuristic methods.

REFERENCES

Beasley, J.E. 1987 “An algorithm for set covering problem”. *European Journal of Operational Research* 31, No.1 85-93.

Castillo, E.; J.M. Menéndez; and P. Jiménez. 2008. “Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations”. *Transportation Research Part B: Methodological* 42, No.5, 455-481.

Cerrone, C.; R. Cerulli; and M. Gentili. 2015. “Vehicle-id sensor location for route flow recognition: Models and algorithms”. *European Journal of Operational Research* 247, No.2, 618-629.

Gentili, M.; and P.B. Mirchandani. 2012. “Locating sensors on traffic networks: Models, challenges and research opportunities”. *Transportation research part C: emerging technologies* 24, 227-255.

Johnson, D.S. 1974. “Approximation algorithms for combinatorial problems”. *Journal of computer and system sciences* 9, No.3, 256-278.

Lan, G.; G.W. DePuy; and G.E. Whitehouse. 2007. “An effective and simple heuristic for the set covering problem”. *European journal of operational research* 176, No.3, 1387-1403.

Mínguez, R.; S. Sánchez-Cambronero; E. Castillo; and P. Jiménez. 2010. “Optimal traffic plate scanning location for OD trip matrix and route estimation in road networks”. *Transportation Research Part B: Methodological* 44, No.2, 282-298.

Vasko, F.J.; Y. Lu; and K. Zyma. 2016. “What is the best greedy-like heuristic for the weighted set covering problem?”. *Operations Research Letters* 44, No.3, 366-369.

Zangui, M.; Y. Yin; and S. Lawphongpanich. 2015. “Sensor location problems in path-differentiated congestion pricing”. *Transportation Research Part C: Emerging Technologies* 55, 217-230.

AUTHOR BIOGRAPHIES

PEGAH T. YAZDI was born in Tehran, Iran in 1992. She went to the University of Tehran, where she studied civil engineering and obtained her bachelor degree in 2015. After that she received her master of science in transportation planning and engineering in 2017 from Sharif university of technology. Her e-mail address is: pghtlbn@gmail.com

YOUSEF SHAFAHI received his Ph.D. in Civil Engineering from University of Maryland, USA in 1997. He joined the Department of Civil Engineering in Sharif University of Technology as an Assistant Professor in 1998 and became an Associated Professor in 2009. His main research interest is operation research in Transportation Planning. His email address is shafahi@sharif.edu and his web-page can be found at http://sharif.edu/~shafahi