

# Characteristics of optimal solutions to the sensor location problem

David R. Morrison · Susan E. Martonosi

Published online: 18 June 2014  
© Springer Science+Business Media New York 2014

**Abstract** The sensor location problem is that of locating the minimum number of traffic sensors at intersections of a road network such that the traffic flow on the entire network can be determined. In this paper, we provide a new necessary condition on the location of these sensors to enable the traffic flow throughout the network to be computed. This condition is not sufficient in general, but we show that for a large class of problem instances, the condition is sufficient. Many typical road networks are included in this category, and we show how our condition can be used to inform traffic sensor placement.

**Keywords** Traffic flows · Sensor location · Transportation network design · Network flow · Network optimization

## 1 Introduction

Traffic congestion is a significant problem in most major cities in the world. An important first step in mitigating road congestion is to know the distribution of cars on each road of the network. This can be achieved by using traffic sensors to count cars traveling into and out of an intersection. However, placing sensors on every intersection is not only prohibitively expensive, it is also inefficient: if some sensors were removed, traffic flow through those intersections might still be calculated by applying flow conservation laws and knowledge of

---

This research was supported in part by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the the Department of Defense or the United States Government.

---

D. R. Morrison  
University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave, Urbana, IL 61801, USA  
e-mail: drmorr0@gmail.com

S. E. Martonosi (✉)  
Harvey Mudd College, 301 Platt Blvd, Claremont, CA 91711, USA  
e-mail: martonosi@g.hmc.edu

the fraction of cars turning in each direction at each intersection. In fact, even a vertex cover is inefficient for these reasons. Thus, we want to locate the minimum number of sensors such that we can still determine the distribution of cars in the entire network. This problem is known as the sensor location problem (SLP).

Bianco et al. (2001) present a necessary and sufficient condition on the set  $M$  of monitored intersections such that the traffic flow on the entire network is calculable. We present a counterexample demonstrating that the condition, while necessary, is not sufficient. Using the insights provided by the counterexample, we develop a stronger necessary condition that, while not sufficient in general, is sufficient in a large class of networks in which a particular *unmonitored subgraph*, to be defined in this paper, is a tree. We present several examples of road networks, including the standard grid network, to which this sufficient condition can be used to confirm that the flow can be completely specified. Moreover we present examples for which the condition is *not* sufficient, but where the failure of the necessary condition also provides useful information about the network.

In Sect. 2, we present some related work to this problem. Section 3 gives some definitions and notation, and formally defines the SLP. We present our counterexample to Bianco et al. (2001) in Sect. 4, and develop a matrix representation for the problem in Sect. 5. Section 6 derives a graph-theoretic necessary condition for flow computability, and Sect. 7 demonstrates that this condition is sufficient in the case when each unmonitored subgraph is a tree. In Sect. 8 we provide examples of how this new condition could be used for decision support by traffic engineers. We offer concluding remarks in Sect. 9.

## 2 Related work

The problem of locating sensors in a network has been widely-studied across many different disciplines. Lam and Lo (1990) provide one of the first formal treatments of the problem with respect to traffic flow, by considering how to estimate origin-destination matrices from traffic counts; this work is extended in Yang and Zhou (1998), who provide four heuristic rules for locating sensors in a network to provide the best estimation of the origin-destination matrix.

Bianco et al. (2001, 2006) extend this problem by studying conditions on the placement of sensors at vertices which enable the flow in the network to be uniquely determined. They prove that determining the optimal placement of sensors on vertices is NP-hard, and present a number of special cases in which the problem is more tractable.

A related problem is studied by Gu and Jia (2005) and Chin et al. (2009), in which sensors are not located on vertices but on the network arcs. Gu and Jia (2005) show that for any strongly-connected directed graph with  $m$  edges and  $n$  vertices,  $m - n + 1$  sensors are needed to uniquely determine the flow value. Chin et al. (2009) study the problem of placing edge sensors on an undirected graph; in this setting as well, the problem is shown to be NP-hard. Chin et al. (2009) then present two different approximation algorithms for the problem that run in polynomial time.

A number of heuristic approaches have also been employed to solve the SLP. Zhang et al (2007) propose a hybrid genetic algorithm with simulated annealing approach, where they examine the problem from the perspective of monitoring network traffic instead of physical transportation flows. Zhang et al (2007) extend work done by Suh et al. (2005), who propose a number of greedy heuristics, and test them on real network topologies.

In the next section we formally define the version of the SLP that we consider in the remainder of this paper.

### 3 Definitions

Let the road network be represented by a directed graph  $G = (V, A)$ , where  $V$  is a set of intersections and  $A$  is a set of “two-way” directed arcs (roads). That is, if  $u, v \in V$  and  $uv \in A$ , then  $vu \in A$ , but the traffic flow on arc  $uv$  need not equal that on arc  $vu$ . We represent the traffic flowing over the roads by a network flow function  $f : A \rightarrow \mathbb{R}$  that satisfies the flow conservation law at each vertex  $v \in V$ :

$$\sum_{e \in v^-} f_e - \sum_{e \in v^+} f_e + S_v = 0, \tag{1}$$

where  $v^-$  is the set of arcs with head at  $v$ ,  $v^+$  is the set of arcs with tail at  $v$ , and  $S_v$  is the **balancing flow** at vertex  $v$ . The sources and sinks of traffic, called **centroids**, are the vertices with non-zero balancing flows; the set of all such centroids we denote  $B$ . Because flow is conserved at each vertex, we have  $\sum_{v \in V} S_v = 0$ . We assume that while the set  $B$  is known, the values of the balancing flows for vertices in  $B$  are unknown.

To determine the network flow function  $f$ , sensors are placed at various intersections in the road network. We denote the set of **monitored vertices** by  $M$ . If an intersection is monitored, then the number of cars entering and leaving the intersection along each road connected to the intersection is revealed. We denote the set of vertices directly adjacent to vertices in  $M$  via an arc in  $A$  as  $A(M)$ .

We finally assume knowledge of the **turning ratios** at every intersection in the network. The turning ratio  $c_{vu}$  for arc  $vu$  at vertex  $v$  is simply the percent of net incoming traffic to  $v$  that leaves along arc  $vu$ . That is,

$$f_{vu} = c_{vu} \sum_{e \in v^-} f_e. \tag{2}$$

Define the **turning factor** of arc  $vu$  with respect to given reference arc  $vw$  to be the ratio of their turning ratios:

$$\alpha_{vu} = \frac{c_{vu}}{c_{vw}}. \tag{3}$$

Then we can write the flow  $f_{vu}$  of any outgoing arc  $vu$  from  $v$  in terms of  $f_{vw}$  as

$$f_{vu} = \alpha_{vu} f_{vw}. \tag{4}$$

The values for the turning ratios can be obtained from historical data about traffic patterns if available, or can be determined easily by monitoring existing traffic patterns for a short time.

We are now ready to define the SLP:

**Definition 1** (*Sensor Location Problem*) Given a two-way directed graph  $G = (V, A)$ , a network flow function  $f$  and a set of centroids  $B$ , what is the smallest set  $M$  of monitored vertices such that knowledge of all turning ratios, the values of  $f$  on incoming and outgoing arcs of  $M$  and balancing flows  $S_v$  on  $M$  uniquely determines  $f$  and the balancing flows  $S_v$  everywhere on  $G$ ?

We focus on the verification version of SLP and seek a condition to verify that a proposed set  $M$  uniquely determines  $f$  and the balancing flows. In the next section, we demonstrate that a previous condition for verification proposed by Bianco et al. (2001) is incorrect in certain cases.

### 4 A proposed condition and counterexample

When a set  $M$  of vertices is monitored, the flow on all arcs between vertices in  $M$  and between  $M$  and  $A(M)$  are known, as well as the balancing flows at each centroid in  $M$ . Applying the turning ratios, we also know the flow on all arcs between vertices in  $A(M)$ . We call the set of arcs connecting vertices in  $M$  and  $A(M)$ , on which the flow can be computed directly from monitoring  $M$  and applying turning ratios, the **combined cutset** of  $M$ :

**Definition 2** (Bianco et al. 2001) The **combined cutset** of  $M$ ,  $C_M$ , is the set of arcs in the subgraph of  $G$  induced by  $M \cup A(M)$ .

We see from the definition of the combined cutset of  $M$  that these are arcs over which the problem of determining the flow has already been solved directly from monitoring. Thus, we can remove  $C_M$  from the graph, and try to use the remaining flow coming out of  $A(M)$ , turning ratios and flow balance equations to determine the flow everywhere else in the graph. We therefore define the **unmonitored subgraph** of  $G$  to be the subgraph  $G'$  that remains when  $C_M$  has been removed from the graph:  $G' = (V - M, A - C_M)$ . This subgraph contains all arcs over which the flow is not completely determined by monitoring.

The unmonitored subgraph is often, but not always, disconnected. We call the  $i$ th connected component of the unmonitored subgraph the  $i$ th **unmonitored component** and label it  $G'_i$ . We label the set of centroids in that component  $B_i$ , and the set of (originally) adjacent vertices in that component  $A_i(M)$ . Bianco et al. (2001) present a proof of the following condition on the set  $M$  in order for the flow function  $f$  to be uniquely determined. While this is a necessary condition, we present an example that demonstrates it is not actually sufficient in general.

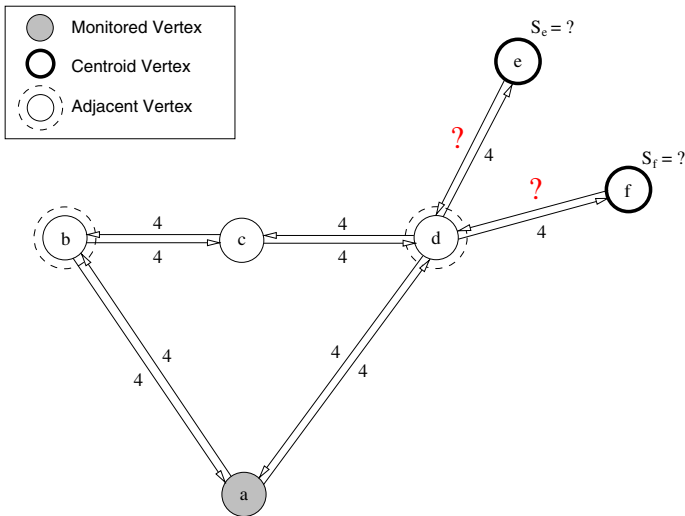
**Theorem 1** (Bianco et al. 2001) *Given a set of monitored vertices  $M$ , the flow on a digraph  $G$  can be uniquely determined everywhere if and only if for every unmonitored component  $G_i$  of  $G$ ,*

$$|B_i| \leq |A_i(M)|.$$

In their proof of this theorem, the authors compare the number of unknown arc and balancing flow variables to the number of flow balance and turning ratio equations when this condition holds. They argue (correctly) that the number of equations must be at least the number of unknowns, which happens only if  $|B_i| \leq |A_i(M)|$ . However, in their argument that the condition is sufficient, they neglect the possibility that some of the resulting equations might be linearly dependent, and thus the solution will not be unique.

To see this, consider the following example (shown in Fig. 1). Let  $\delta^+(u)$  be the outgoing degree of vertex  $u$ , and suppose that the turning ratios  $c_{uv} = 1/\delta^+(u)$  for all arcs  $uv$  (the flows on all outgoing arcs from vertex  $u$  are equal). By monitoring vertex  $a$ , the unmonitored subgraph  $G'$  induced by removing the combined cutset has only a single connected component, consisting of the vertices  $b, c, d, e, f$  and the arcs between them.  $A(M)$  in this component is  $\{b, d\}$ , and  $B - M = \{e, f\}$ . Thus  $|A(M)| = |B - M| = 2$ , and by Theorem 1, we should be able to determine  $f$  and the vector  $S$  of balancing flows uniquely.

However, suppose we observe 4 units of flow along arcs  $ab, ba, ad$ , and  $da$ . We apply the flow balance equation and knowledge of the turning ratios sequentially at each vertex until we get stuck. Consider vertex  $b$ . It is not a centroid, so  $S_b = 0$ , and since flows on all outgoing arcs are equal,  $f_{bc} = f_{ba} = 4$ . To preserve balance of flow,  $f_{cb} = 4$  as well. By a similar logic, we obtain  $f_{cd} = f_{dc} = 4$  at vertex  $c$  and  $f_{de} = f_{df} = 4$  at vertex  $d$ . We cannot determine  $f_{ed}$  and  $f_{fd}$  because both  $e$  and  $f$  are centroids, and their balancing flows are



**Fig. 1** A counterexample to the flow calculation theorem (Theorem 1). If we monitor vertex  $a$  in the above graph, the graph with cutset  $C_M = \{ab, ba, ad, da\}$  removed satisfies the conditions in Theorem 1. However, we cannot calculate  $f_{ed}$  or  $f_{fd}$  from the known information

unknown. Balancing flows in the network must sum to zero, so  $S_f = -S_e$ , leaving us with the following system of equations having three unknowns and three equations, as predicted by Theorem 1.

$$\begin{aligned}
 f_{ed} + f_{fd} &= 8 \\
 f_{ed} - S_e &= 4 \\
 f_{fd} + S_e &= 4
 \end{aligned}
 \tag{5}$$

Notice, however, that these equations are linearly *dependent* and thus fail to admit a unique solution. Therefore, the condition provided in Theorem 1 is not sufficient.

Unfortunately, there are many such counterexamples, including cases when the graph is a tree or when the inequality in the theorem is strict. Fortunately, the subsequent work of Bianco et al. (2006) is correct despite the erroneous Theorem 1. Nonetheless, it is still valuable to understand why the theorem is incorrect and to formulate a new theorem that guarantees the computability of traffic flows on a monitored graph. To better understand the circumstances under which Theorem 1 fails, we next examine the problem via the graph’s incidence matrix.

### 5 SLP and invertible matrices

Let  $\mathbf{E}$  be the  $|V| \times |A|$  incidence matrix where the  $(u, e)^{th}$  entry is  $-1$  if vertex  $u$  is the tail of arc  $e$ ,  $1$  if it is  $e$ ’s head, and  $0$  if  $e$  is not incident to  $u$ . Let  $\mathbf{f}$  be the  $|A|$ -length vector of unknown arc flows and  $\mathbf{S}$  the  $|V|$ -length vector of balancing flows. The system of linear flow conservation constraints at each vertex then takes the form

$$\mathbf{E}\mathbf{f} + \mathbf{S} = \mathbf{x},
 \tag{6}$$

where  $\mathbf{x} = \mathbf{0}$ . Notice that the sum of these equations yields the balancing flow constraint  $\sum_{u \in V} S_u = 0$ , so we do not need to add this constraint to system (6).

This system does not include the known turning ratios or the observed flow along monitored arcs, and thus contains more unknown variables than are necessary. We can therefore reduce this system of equations to a more compact representation, as follows:

1. For each vertex  $u \in V$ , we designate an arbitrary outgoing arc  $e_u$  to be the **canonical arc** for vertex  $u$ . Since we know the turning ratios of the graph, the flow over any arc  $uv$  is  $f_{uv} = \alpha_{uv} f_{e_u}$ . This reduces the number of flow variables from  $|A|$  to  $|V|$ , and we can modify the unknown flow vector  $\mathbf{f}$  to include only the  $|V|$  canonical arcs.
2. Having expressed the flow on any arc  $uv$  in terms of the flow on  $e_u$ , the flow balance matrix  $\mathbf{E}$  collapses into a square matrix  $\hat{\mathbf{E}}$ , where row  $u$  still corresponds to the balance equation at vertex  $u$ , and column  $v$  corresponds to the canonical arc for vertex  $v$ ,  $e_v$ . The  $(u, v)^{th}$  entry of  $\hat{\mathbf{E}}$  is given by

$$\hat{\mathbf{E}}_{u,v} = \begin{cases} \alpha_{vu} & \text{if } u \text{ and } v \text{ are connected} \\ -\sum_w \text{ adjacent to } u \alpha_{uw} & \text{if } u = v \\ 0 & \text{if } u \text{ and } v \text{ are not connected} \end{cases}$$

3. We also augment  $\hat{\mathbf{E}}$  with  $|B|$  columns for the unknown balancing flows at the centroids. The column corresponding to the centroid at vertex  $u$  has a 1 in the  $u$ th row and 0's everywhere else. Likewise, we create a single  $(|V| + |B|)$ -length vector  $\mathbf{g} = \begin{bmatrix} \mathbf{f} \\ \mathbf{S} \end{bmatrix}$  of unknown canonical arc and balancing flows. Equation (6) then becomes

$$\hat{\mathbf{E}}\mathbf{g} = \mathbf{x}, \tag{7}$$

where  $\mathbf{x}$  is still the zero vector.

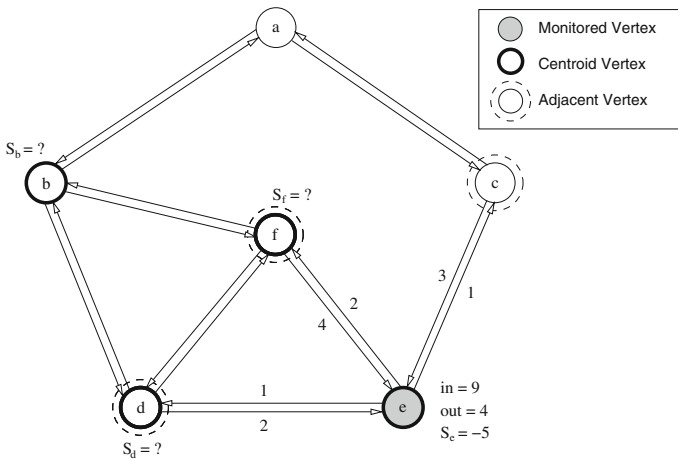
We next incorporate the known flow values obtained by monitoring vertices in  $M$ .

4. For each vertex  $m \in M$ , the flow along  $m$ 's canonical arc and the balancing flow (if  $m$  is a centroid) are known. We can remove row  $m$  from the matrix  $\hat{\mathbf{E}}$ . We also remove column  $m$ , corresponding to vertex  $m$ 's canonical arc. Next, we update the right-hand side vector  $\mathbf{x}$  with the known flow values by subtracting  $f_{e_m}$  times the removed  $m^{th}$  column from  $\mathbf{x}$ . This is equivalent to subtracting  $\alpha_{mu} f_{e_m}$  from the  $u^{th}$  entry of  $\mathbf{x}$  for each vertex  $u$  adjacent to  $m$ . If  $m$  is a centroid, we also remove the column of  $\hat{\mathbf{E}}$  corresponding to its balancing flow. We likewise remove the entry from  $\mathbf{g}$  corresponding to  $f_{e_m}$  (and  $S_m$  if  $m$  is a centroid), and remove the  $m^{th}$  entry from  $\mathbf{x}$ .
5. For each vertex  $a \in A(M)$ , the outgoing flow from  $a$  to any vertex  $m \in M$  is monitored, so by turning ratios, we can deduce the flow over  $a$ 's canonical arc. We therefore remove column  $a$  from  $\hat{\mathbf{E}}$  and subtract  $f_{e_a} = \frac{1}{\alpha_{am}} f_{am}$  times column  $a$  from the right-hand side vector  $\mathbf{x}$ . This is equivalent to subtracting  $\alpha_{au} f_{e_a}$  from the  $u^{th}$  entry of  $\mathbf{x}$  for each vertex  $u$  adjacent to  $a$  and adding  $\sum_w \text{ adjacent to } a \alpha_{aw} f_{e_a}$  to the  $a^{th}$  entry of  $\mathbf{x}$ . We remove the entry from  $\mathbf{g}$  corresponding to  $f_{e_a}$ . We name the resulting coefficient matrix for the system of equations the **flow calculation matrix**  $\mathbf{F}$  and rewrite for the last time our original system of equations

$$\mathbf{F}\mathbf{g} = \mathbf{x}. \tag{8}$$

If Eq. (8) has a unique solution (which occurs when the columns of  $\mathbf{F}$  are linearly independent), then we can uniquely determine the flow everywhere on the graph.

For example, consider the graph in Fig. 2, with  $M = \{e\}$  and flows on monitored arcs as indicated in the figure. We choose arcs  $ab, ba, ca, db, ed$  and  $fb$  to be our canonical



**Fig. 2** A network in which the set of centroids is  $B = \{b, d, e, f\}$ , and vertex  $e$  is monitored, revealing the flows indicated on the arcs into and out of  $e$ . In this case, we can calculate the flow everywhere on the graph, as demonstrated by Eq. (9) having a unique solution

representatives for each vertex. We also assume that all turning ratios are equal except at vertex  $e$  (so  $\alpha_{uv} = 1$  for all  $uv \neq e$ ), where monitoring has revealed the turning factors to be  $\alpha_{ef} = 2$  and  $\alpha_{ec} = 1$ . The corresponding reduced system of equations is:

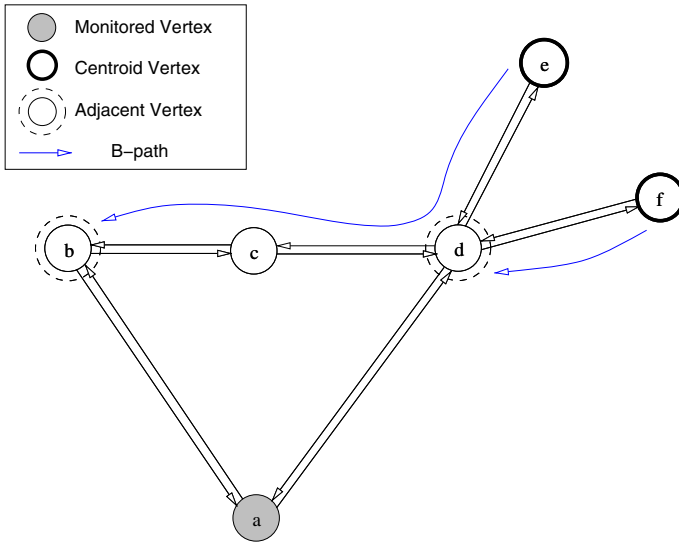
$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_{ab} \\ f_{ba} \\ S_b \\ S_d \\ S_f \end{pmatrix} = \begin{pmatrix} -3 \\ -6 \\ 5 \\ 1 \\ 8 \end{pmatrix} \tag{9}$$

It is easy to check that  $\text{rank}(\mathbf{F}) = 5$ , and thus the columns are linearly independent; this implies that Eq. (8) is solvable for the graph in Fig. 2.

### 6 A new necessary condition

The flow is uniquely calculable if and only if the matrix  $\mathbf{F}$  has full column rank. An obvious necessary (but insufficient) condition is for  $\mathbf{F}$  to have at least as many rows as columns.  $\mathbf{F}$  has  $|V| - |M| - |A(M)| + |B - M|$  columns and  $|V| - |M|$  rows. Therefore, we require  $|B - M| \leq |A(M)|$ . The necessary condition proved in Bianco et al. (2001) is stronger:  $|(B - M)_i| \leq |A(M)_i|$  for all connected components  $i$  in the unmonitored subgraph induced by removing the arcs in the combined cutset. In fact, we can prove an even stronger necessary condition that relies solely on the topology of the graph. This condition is motivated by studying the example in Fig. 1.

Our difficulty in calculating the flow in this example arose when we reached vertex  $d$ . Although we knew the flow exiting vertex  $d$  along the arcs toward  $e$  and  $f$ , we were unable to determine the flow entering  $d$  because both  $e$  and  $f$  were centroids, contributing unknown balancing flows. Traffic originating or terminating at vertices  $e$  and  $f$  got “mixed up” at vertex  $d$  and could not be uniquely differentiated.



**Fig. 3** The graph from Fig. 1, together with a set of  $B$ -paths. However, any two  $B$ -paths must pass through vertex  $d$ , so there is no set of  $|B - M|$  disjoint  $B$ -paths associated with  $M$

This observation motivates the following definition:

**Definition 3** A  **$B$ -path** is a path starting at a centroid and ending at a vertex in  $A(M)$ .

Using this definition, we present the following theorem, which provides a stronger necessary condition for flow computability:

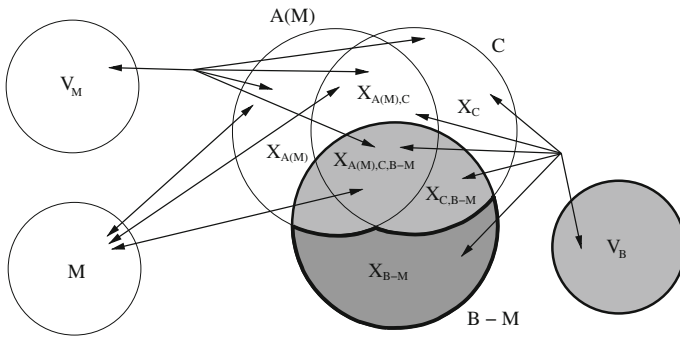
**Theorem 2** (Statement A) *Let  $G = (V, A)$  be a two-way directed graph with centroid set  $B$ , and let  $M$  be a set of monitored vertices. The flow on arcs in  $G$  and the balancing flow at the vertices in  $B$  can be uniquely determined everywhere only if there exists a set  $\mathcal{P}$  of  $|B - M|$  vertex disjoint  $B$ -paths.*

This is a stronger necessary condition than that given in Theorem 1 because it is not satisfied by our counterexample in Fig. 1. We see in Fig. 3 that any set of two  $B$ -paths will be forced to intersect at vertex  $d$ . Thus, the number of disjoint  $B$ -paths is smaller than  $|B - M|$  and we are unable to calculate the flow.

To prove Theorem 2, we must translate its statement related to the topological structure of the network into our algebraic framework described earlier. We note first that the number of vertex-disjoint  $B$ -paths cannot be larger than the size of a minimum disconnecting set  $C$  between  $B - M$  and  $A(M)$ , by Menger’s theorem. Thus, we require  $|C| \geq |B - M|$ . (In fact, the size of the minimum disconnecting set will never strictly exceed  $|B - M|$ ).

Next, we partition the graph  $G$  into its unmonitored components by removing the combined cutset. If  $u$  and  $v$  are in different partitions in the graph, then there was no path from  $u$  to  $v$  in  $G$  except through  $M$  or along an  $a_1 a_2$  edge for some  $a_1$  and  $a_2 \in A(M)$ . Because all rows and columns corresponding to  $M$  and all columns corresponding to  $A(M)$  have been removed from the matrix, vertex  $u$ ’s flow balance equation will not include any  $e_v$  or  $S_v$  terms, and  $e_u$  and  $S_u$  will not appear in vertex  $v$ ’s flow balance equation. Thus, we can rearrange the flow calculation matrix  $\mathbf{F}$  into block form by collecting rows and columns corresponding to vertices in each unmonitored component, and prove the theorem for each component independently. We rephrase our original theorem accordingly:





**Fig. 4** The partition of the vertex set for Theorem 2.  $V_M$  is the set of unaccounted-for vertices on the  $M$  side of the cut, and  $V_B$  is the set of unaccounted-for vertices on the  $B - M$  side of the cut. Bold arrows indicate possible connections between sets. Some of these sets may be empty—in particular, note that by definition, there can be no vertices in  $((B - M) \cap A(M)) \setminus C$ , since  $C$  must separate  $B - M$  and  $A(M)$ . The shaded-in regions correspond to the columns included in the submatrix  $\mathbf{F}^*$

**Theorem 2** (Statement B) *Let  $G$ ,  $B$ , and  $M$  be as in Theorem 2 (Statement A), with the graph partitioned into unmonitored components and the flow calculation matrix partitioned into blocks as described. For each unmonitored component  $i$ , let  $C_i$  be the minimum vertex cut between  $(B - M)_i$  and  $A(M)_i$ . (If  $(B - M)_i$  is empty, then let  $C = \emptyset$ ).  $\text{rank}(\mathbf{F}^i) = \#\{\text{columns of } \mathbf{F}^i\}$  (and hence the flow on  $G_i$  is calculable) only if  $|C_i| = |(B - M)_i|$ .*

*Proof* For ease of notation, we drop the subscript  $i$  and henceforth refer to all sets in the context of a given unmonitored component  $i$ . We assume the component contains at least one centroid, otherwise the theorem is true trivially because both  $B - M$  and  $C$  are empty. Within component  $i$ , we call  $V_M$  the set of vertices that are not in  $M$  or  $A(M)$  and are connected to  $M$  by some path that does not pass through  $C$  (i.e. they are on the  $M$  side of the cut  $C$ ). Similarly, we call  $V_B$  the set of vertices not in  $B - M$  that are on the  $B - M$  side of the cut. Note that  $C$ ,  $A(M)$ , and  $B - M$  could all overlap, as shown in Fig. 4; we label these intersections as shown, where  $X_{A(M),C} = (A(M) \cap C) \setminus (B - M)$ ,  $X_{A(M)} = A(M) \setminus (C \cup (B - M))$ , etc. Note that since  $C$  is by definition a vertex cut between  $B - M$  and  $A(M)$ , the set  $X_{A(M),B-M} = (A(M) \cap (B - M)) \setminus C$  is empty.

Let us consider a submatrix  $\mathbf{F}^*$  of  $\mathbf{F}$  that contains only the columns corresponding to canonical arcs for vertices in  $X_{B-M}$  and  $V_B$  and to balancing flows at vertices in  $X_{B-M}$ ,  $X_{C,B-M}$  and  $X_{A(M),C,B-M}$ . These are the shaded regions of Fig. 4. Since  $\mathbf{F}$  has linearly independent columns,  $\mathbf{F}^*$  has full column rank, and

$$K \leq R - Z, \tag{10}$$

where  $K$  is the number of columns of  $\mathbf{F}^*$ ,  $R$  the number of rows and  $Z$  the number of zero rows. By construction,

$$K = |X_{B-M}| + |V_B| + |B - M|$$

and

$$R = |X_{A(M)}| + |X_{A(M),C}| + |X_{A(M),C,B-M}| + |X_{B-M}| + |X_C| + |X_{C,B-M}| + |V_M| + |V_B|.$$

Next we determine  $Z$ . As we see in Fig. 4, there are no arcs from vertices in  $V_M$  or  $X_{A(M)}$  to vertices in  $X_{B-M}$  or  $V_B$  by definition of the cut  $C$ . Moreover, vertices in  $V_M$  and  $X_{A(M)}$

are not centroids. Therefore, the rows in  $\mathbf{F}^*$  corresponding to vertices in  $V_M$  or  $X_{A(M)}$  are all zero, and  $Z \geq |V_M| + |X_{A(M)}|$ . Applying inequality (10) and canceling common terms, we see that  $|B - M| \leq |X_C| + |X_{A(M),C}| + |X_{C,B-M}| + |X_{A(M),C,B-M}| = |C|$ . Because  $|C|$  can never exceed  $|B - M|$ , we have  $|C| = |B - M|$ .  $\square$

As an example, we walk through the construction of the  $\mathbf{F}^*$ -matrix for our original counterexample of Fig. 1. We start with the flow calculation matrix  $\mathbf{F}$  in the system of equations  $\mathbf{F}\mathbf{g} = \mathbf{x}$ :

$$\begin{aligned}
 b: & \left( \begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{array} \right) \begin{pmatrix} f_{cb} \\ f_{ed} \\ f_{fd} \\ S_e \\ S_f \end{pmatrix} = \begin{pmatrix} 4 \\ -8 \\ 12 \\ -4 \\ -4 \end{pmatrix} \tag{11}
 \end{aligned}$$

When we remove the cutset associated with monitored vertex  $a$ , the minimum vertex cut between  $A(M) = \{b, d\}$  and  $B - M = \{e, f\}$  is  $C = \{d\}$ .  $|C| \neq |B - M|$ , so we should not be able to calculate the flow. We generate the  $\mathbf{F}^*$  submatrix using the sets  $X_{A(M)} = \{b\}$ ,  $V_M = \{c\}$ ,  $X_{A(M),C} = \{d\}$ , and  $X_{B-M} = \{e, f\}$ :

$$\mathbf{F}^* = \begin{matrix} & \begin{matrix} ed & fd & S_e & S_f \end{matrix} \\ \begin{matrix} b: \\ c: \\ d: \\ e: \\ f: \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \end{matrix} \tag{12}$$

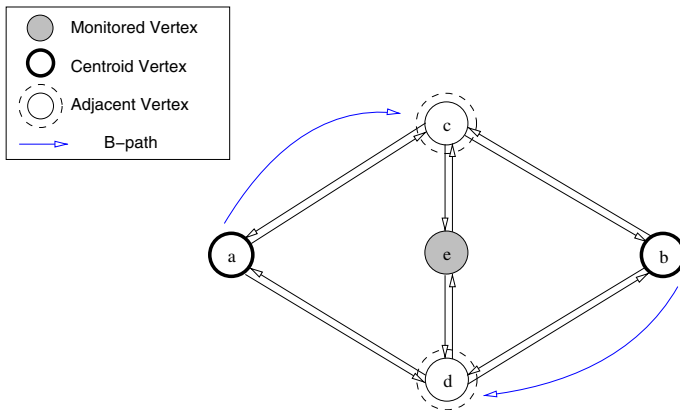
Notice that the first two rows of the matrix are 0, which means that the rank of this submatrix can't be any higher than 3. This implies that the rank of  $\mathbf{F}$  cannot equal the number of columns, and thus the flow on the graph cannot be calculated.

### 7 A sufficient condition for trees

Next we turn to the question of sufficiency. Unfortunately, the condition is not sufficient for graphs in general, but is sufficient in the case of networks whose unmonitored components are all trees. Figure 5 provides an example of a general graph in which there are  $|B - M|$  vertex-disjoint  $B$ -paths, but the matrix  $E_M$  still does not have linearly independent columns and the flow on the graph cannot be calculated.

We see that the unmonitored subgraph of this example, which we obtain by removing  $M$ 's combined cutset (in this case, the arcs  $ce, ec, de,$  and  $ed$ ), is not a tree. However, the following theorem states that as long as the unmonitored components of a graph are all trees, our condition is sufficient to guarantee the calculability of traffic flow.

**Theorem 3** *Let  $G, B,$  and  $M$  be as in Theorem 2, with the flow calculation matrix partitioned into blocks as described. For each unmonitored component  $i$ , let  $C_i$  be the minimum vertex cut between  $(B - M)_i$  and  $A(M)_i$ . If the  $i^{\text{th}}$  component is a tree, then  $\text{rank}(\mathbf{F}^i) = \#\{\text{columns of } \mathbf{F}^i\}$  (that is, the flow on block  $i$  is calculable) if and only if  $|C_i| = |(B - M)_i|$ .*



**Fig. 5** In this graph, monitoring vertex  $e$  creates two disjoint  $B$ -paths, satisfying the conditions of Theorem 2. However, the  $F$  matrix does not have linearly independent columns, and thus we cannot calculate the flow on the graph

Prior to proving this theorem, we first prove the following lemma:

**Lemma 1** *Let  $G$  be a two-way directed tree with known turning ratios, containing no centroids and having root vertex  $r$ . Suppose that  $G$  is attached at  $r$  to a graph  $\hat{G}$  at vertex  $v$  with known flow value  $f_{vr}$ . Then  $f_{rv} = f_{vr}$  and the flow on  $G$  can be determined.*

*Proof* We prove this by induction on the number  $n$  of vertices in  $G$ . As a base case, suppose  $n = 1$ , then  $G$  contains only the leaf node  $r$ . Because  $r$  is not a centroid, its balancing flow is zero, so  $f_{rv} = f_{vr}$ , and the flow on  $G$  has been determined.

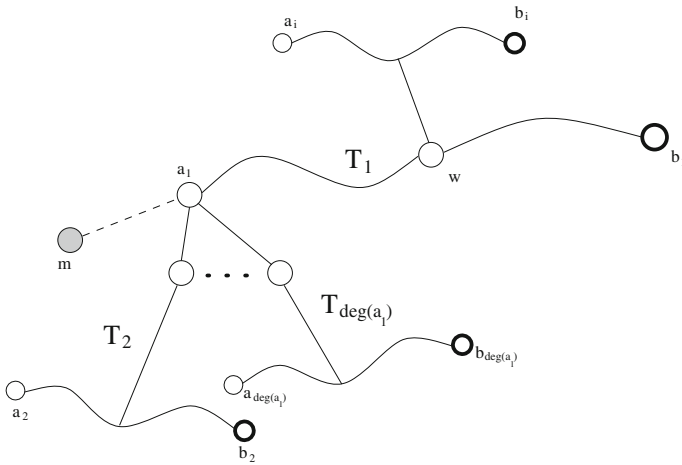
Suppose the statement is true for any tree of size strictly less than  $n$  and let  $G$  have size  $n$ . Consider the vertex  $r \in G$  which is attached to graph  $\hat{G}$  at vertex  $v$ . Because  $G$  is a tree with no centroids, flow is conserved, so  $f_{rv} = f_{vr}$ , and all other outgoing flows of  $r$  can be determined using turning ratios. Moreover,  $r$  is connected to  $\text{deg}(r)$  subtrees of  $G$  each of size strictly less than  $n$  and having root vertices  $v_i, i = 1 \dots \text{deg}(r)$  with known incoming flow values  $f_{rv_i}$ . So the flow on each subtree can also be determined, and  $f_{v_i r} = f_{rv_i}$ . We have therefore found the flow on the entire graph.  $\square$

Thus, on a subtree with no centroids, the flow can be computed knowing only a single incoming arc to the tree. We now prove Theorem 3.

*Proof* Theorem 2 handles the necessary condition. Let  $n = |(B - M)_i|$ . Because  $|C_i| = |(B - M)_i|$ ,  $|A(M)_i|$  must also be at least  $n$ , and there exists a pairing between a subset of  $A(M)_i$  and the vertices in  $(B - M)_i$  such that the set of paths between all pairs  $a_i \in A(M)_i$  and  $b_i \in (B - M)_i$  are vertex disjoint. (If  $|A(M)_i| > n$ , then the “extra” vertices in  $A(M)_i$  will act like centroids with known balancing flow equal to the incoming flow from  $M$  and can be treated like any other non-centroid). We will induct on  $n$  to show how to propagate the flow calculation through the graph.

If  $n = 0$ , then our partition consists of a tree having no centroids that, in the original graph, is connected to a vertex in  $M$ . The flow along an incoming arc to  $G'_i$  from  $M$  is known due to monitoring, so our Lemma 1 tells us that the flow along every arc in  $G'_i$  can be determined.

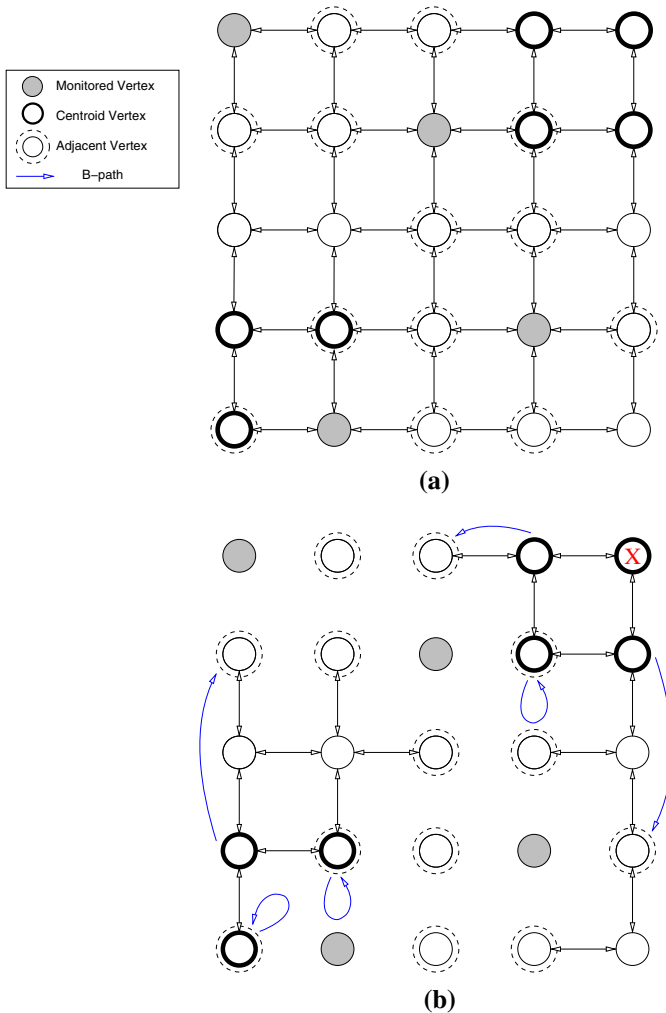
Suppose the theorem is true for any partition having  $|(B - M)_i| < n$  unmonitored centroids, and consider a partition having  $|(B - M)_i| = n$ . Without loss of generality, consider



**Fig. 6** The tree structure induced by the pairing of centroids and adjacent vertices in Theorem 3. Note that no path from a vertex  $a_j \in A(M)$  to its pair  $b_j \in B - M$  can cross any other vertex in  $A(M)$ . This allows us to treat each subtree separately when calculating the flow along it

centroid  $b_1$  and its matching adjacent vertex  $a_1$ . Let  $T_1, \dots, T_{deg(a_1)}$  be the subtrees of  $T$  rooted at  $a_1$ 's neighbors  $t_1, \dots, t_{deg(a_1)}$ , and assume that  $T_1 \cup \{a_1\}$  is the tree containing the  $(a_1, b_1)$  pair (see Fig. 6). Every subtree maintains the original pairing of vertices in  $A(M)_i$  and  $(B - M)_i$  because these pairings corresponded to vertex-disjoint paths which could not pass through  $a_1$ . Moreover, each subtree  $T_j, j \neq 1$  has strictly fewer than  $n$  such pairings and satisfies the induction hypothesis. The flow on these subtrees can be calculated. It remains to determine the flow on the edges between  $a_1$  and  $t_j$ :  $f_{a_1 t_j}$  is an outgoing flow from  $a_1$  which is known by monitoring and application of turning ratios.  $f_{t_j a_1}$  can be expressed in terms of the flow on  $t_j$ 's canonical edge. Thus, there is still a unique solution for the flow on  $T_j \cup \{a_1\}$ , and all incoming flow to  $a_1$  from trees  $T_2, \dots, T_{deg(a_1)}$  has been determined.

Next we consider the tree  $T_1 \cup \{a_1\}$ , by propagating flow calculations along the path from  $a_1$  to  $b_1$ . The outgoing flow from  $a_1$  along the path is known by applying turning ratios. If  $a_1 = b_1$ , then in fact  $T_1$  is empty.  $a_1$ 's incoming and outgoing flows can be used to find its balancing flow, and the flow on  $G_i$  has been completely determined. If  $a_1 \neq b_1$ , the incoming edge to  $a_1$  from the next vertex on the path to  $b_1$  is the only edge incident to  $a_1$  whose flow has not yet been determined; it can be determined by flow conservation. By a similar logic, we can propagate these flow calculations along the path from  $a_1$  to  $b_1$  until we reach the first vertex  $w$  having degree greater than 2. All outgoing flows of  $w$  are known by applying turning ratios to the known outgoing flow from  $w$  to the vertex preceding  $w$  in the path. Each branch of  $w$  that does not contain  $b_1$  is once again a subtree that maintains the (strictly fewer than  $n$ ) original  $(B - M)_i$  and  $A(M)_i$  pairings. By our induction hypothesis, the flow on this branch can be determined and by the same reasoning as above, the flow on the edges between  $w$  and the branch can also be determined. Flow conservation determines the incoming flow to  $w$  from the next vertex on the path to  $b_1$ . We continue this way until vertex  $b_1$  is reached. All outgoing flows from  $b_1$  are determined by applying turning ratios to the known outgoing flow from  $b_1$  into the vertex preceding  $b_1$  in the path. The flow on branches stemming from  $b_1$  can be determined by our induction hypothesis, and the balancing flow at  $b_1$  is simply the difference between all outgoing and incoming flows at  $b_1$ . The flow on the tree has been calculated. □

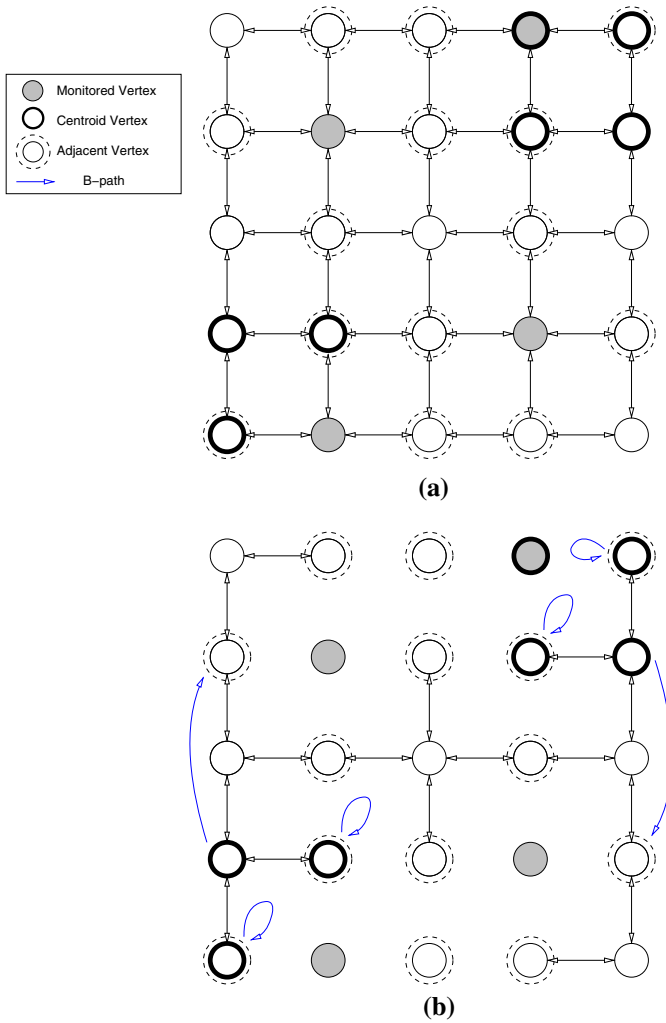


**Fig. 7** A road network with four monitored vertices (*shaded*) and seven centroid vertices (*bold*). The unmonitored subgraph has two components. The leftmost component has three centroids and five vertices in  $A(M)$ . All centroids have a corresponding B-path, but because this unmonitored component is not a tree, we cannot be certain that the flow is calculable on this region of the network. The rightmost component has four centroids and five vertices in  $A(M)$ ; However, the centroid labeled  $X$  does not have its own B-path in this unmonitored component, and hence the traffic flow is not calculable in this region of the network, by Theorem 2 (Statement A). **a** Original road network. **b** Unmonitored subgraph obtained by removing the combined cutset  $C_M$

An obvious corollary is the following:

**Corollary 1** *If  $G$  is a two-way directed tree with centroid set  $B$  and monitored vertex set  $M$ , the flow on  $G$  can be calculated if and only if there exist at least  $|B - M|$  vertex-disjoint paths between  $A(M)$  and  $B - M$ .*

This suggests that it is the presence of cycles in unmonitored subgraphs that can occasionally lead to difficulties in uniquely determining the flow.



**Fig. 8** A road network with four monitored vertices (*shaded*) and seven centroid vertices (*bold*). This network is identical to that of Fig. 7 except that two of the monitored vertices have changed position. In this modified graph, the unmonitored subgraph is a tree, and every centroid has its own B-path. Thus, Theorem 3 guarantees the traffic flow is fully calculable on this network. **a** Original road network. **b** Unmonitored subgraph obtained by removing the combined cutset  $C_M$

### 8 Applications to traffic sensor placement

While it might at first seem restrictive to require the unmonitored subgraph to be a tree in order to guarantee flow calculability, we show in this section that this is not the case. In fact, a broad collection of road networks can have trees as unmonitored subgraphs. Moreover, even for those networks whose unmonitored subgraphs contain cycles, our condition can still provide useful information about the placement of traffic sensors.

Consider the traffic network shown in Fig. 7a; this is a traditional grid network found in many cities, and it has twenty-five intersections, of which seven are considered to be centroids. A traffic planner might be interested in monitoring four intersections on the network to to

calculate the traffic flow throughout it. If she places the monitors on the four vertices shaded in Fig. 7a, then the unmonitored subgraph is as shown in Fig. 7b. The rightmost component violates the necessary condition of Theorem 2 (Statement A) because the centroid marked by  $X$  does not have its own B-path. Therefore, we are able to conclude that flow is not calculable in this region of the network. The leftmost unmonitored component satisfies our necessary condition, but because this component is not a tree, we are unable to conclude that the flow is calculable in this region strictly from analysis of the graph structure.

If the traffic planner simply rearranges two of the monitored vertices, as shown in Fig. 8, she is easily able to construct a network whose unmonitored subgraph is a tree. The sufficient condition of Theorem 3 is satisfied, and she can conclude that by placing the monitors in this orientation, the traffic flow on the network will be completely calculable.

It is worth pointing out that unmonitored subgraphs might be trees even on very large graphs with only a small fraction of monitored vertices. We considered an  $18 \times 18$  grid network (324 vertices) on which 72 vertices were monitored and all unmonitored subgraphs were trees satisfying the sufficient condition for flow calculability given by Theorem 3. More generally, it could be expanded to a  $3k \times 3k$  graph for any integer  $k$  having  $12k$  monitored vertices. Many more examples of large traffic networks can be found for which Theorem 3 applies. And as we have seen in Fig. 7, even when the unmonitored subgraph is not a tree, failure to satisfy the necessary condition of Theorem 2 signals the need either to increase the number of sensors, or to rearrange their positions until the flow is calculable.

## 9 Conclusions

In this paper, we study necessary conditions for the location of sensors in a flow network that will allow us to uniquely determine the rate of flow everywhere in the network. We corrected a slight error in an earlier theorem that addressed this issue, and presented a stronger necessary condition for this problem that is also sufficient for any unmonitored acyclic subgraph. An example was presented showing how traffic engineers may be able to use this result in a broad array of networks to determine good sensor placement.

Future directions for research would be to determine a sufficient condition for the placement of sensors in a general network; furthermore, as the Sensor Location Problem is NP-hard, it would be worthwhile to use the results in this paper to develop heuristics or approximation algorithms that efficiently produce near-optimal solutions.

**Acknowledgments** The authors would like to thank Dr. Lucio Bianco, Dr. Giuseppe Confessore, and Dr. Monica Gentili for communications and assistance regarding their previous work on the problem.

## References

- Bianco, L., Confessore, G., & Reverberi, P. (2001). A network based model for traffic sensor location with implications on O/D matrix estimates. *Transportation Science*, 35(1), 50–60.
- Bianco, L., Confessore, G., & Gentili, M. (2006). Combinatorial aspects of the sensor location problem. *Annals of Operations Research*, 144(1), 201–234.
- Chin, F., Chrobak, M., & Yan, L. (2009). Algorithms for placing monitors in a flow network. In *Algorithmic aspects in information and management*, Vol. 5564, Lecture Notes in Computer Science (pp. 114–128). Berlin, Heidelberg: Springer.
- Gu, W., & Jia, X. (2005). On a traffic control problem. In: *Proceedings of the 8th international symposium on parallel architectures, algorithms, and networks*.

- Lam, W., & Lo, H. (1990). Accuracy of O-D estimates from traffic counts. *Traffic Engineering and Control*, 31(6), 358–367.
- Suh, K., Guo, Y., Kurose, J., & Towsley, D. (2005). Locating network monitors: Complexity, heuristics, and coverage. In *Proceedings of the 24th annual joint conference of the IEEE computer and communications societies*, (Vol. 1, pp. 351–361).
- Yang, H., & Zhou, J. (1998). Optimal traffic counting locations for origin-destination matrix estimation. *Transportation Research Part B-Methodological*, 32(2), 109–126.
- Zhang, J., Zhang, X., Wu, J. (2007). Genetic simulated annealing algorithms for optimal deployment of flow monitors. In *Proceedings of the 3rd international conference on natural computation*, (Vol. 4, pp 398–402).