Auditing Sum Queries

Francesco M. Malvestuto[1], and Mauro Mezzini[2]

[1] Dipartimento di Informatica, Università "La Sapienza", Via Salaria 113, 00198 Roma, Italy
[2] Telecom Italia, Roma, Italy

Abstract. In an on-line statistical database, the query system should leave unanswered queries asking for sums that could lead to the disclosure of confidential data. To check that, every sum query and previously answered sum queries should be audited. We show that, under a suitable query-overlap restriction, an auditing procedure can be efficiently worked out using flow-network computation.

1 Introduction

Statistical databases raise concerns on the compromise of individual privacy, a statistical database [1] being an ordinary database which contains information about individuals (persons, companies, organisations etc.) but its users are only allowed to access sums of individual data provided that they do not lead to the disclosure of confidential data. Consider a relation scheme R = {`NAME`, `SSN`, `AGE`, `DEPARTMENT`, `SALARY`} and a statistical-query system which answers only queries such as: "What is the sum of salaries of the individuals qualified by the condition P" where P is a "category predicate", that is, a condition on the domain of the pair {`AGE`, `DEPARTMENT`} of "category" attributes such as `DEPARTMENT` ≠ `Direction` & `AGE` ≥ `40`. Such a query is called a (*categorical*) *sum query* on `SALARY`. Assume further that `SALARY` is a confidential attribute. Given a relation with scheme R, if the set T of tuples selected by P is a singleton, say $T = \{t\}$, then the response to the sum query above (if released) would allow the salary of the individual corresponding to the tuple t to be disclosed and, therefore, it should be denied. The response to such a sum query is called a *sensitive sum* [13, 14]. Indeed, more sophisticated sensitivity criteria exist which broaden the class of sum queries that should be left unanswered [13, 14]. A (memoryless) security measure that leaves unanswered only sum queries whose responses are sensitive sums is not adequate for it does not exclude the possibility of computing some sensitive sum by combining nonsensitive responses. What measures suffice to avoid the disclosure of sensitive sums? This is the classical version of the *security control problem* for sum queries. Indeed, a stronger version of this problem is of practical interest since the producers of statistical databases (e.g., Census Bureaux) demand that an "accurate" estimate of no sensitive sum should be possible in the sense that, if s is a sensitive sum, then the possible values of s consistent with the information released by the statistical-query system should span an interval that is not entirely contained in the interval $[(1-p)\,s, (1+p)\,s]$ where p is fixed percentage, called the *protection level* [13, 14]. Note that for p = 0, one has the classical version of the security problem: the exact value of no sensitive sum should be disclosed. The program of the statistical query system that should ensure the protection of sensitive sums is called the "auditor" [3, 10, 13] and it should work as follows. Let R be a relation scheme containing a confidential attribute `S` with domain σ, and let K be the set of category attributes in R. Suppose that sum queries $Q(1)$, …, $Q(n-1)$ on `S` have been already answered when a new sum query $Q(n)$ on `S` arrives. Let $P(v)$ and $q(v)$ be the category predicate and the response to $Q(v)$, $1 \le v \le n$. Without loss of generality [11], we assume that each $P(v)$ is of the form "K in $\kappa(v)$" where $\kappa(v)$ is a nonempty subset of the domain of K. The amount of information that would be released to the users if $Q(n)$ were answered will represented by a model which consists of a semantic part and an analytic part. The semantic part of the model contains the overlap relationships among the sum queries $Q(v)$ and consists of a hypergraph $G = (V, A)$ where $V = \{1, \ldots, n\}$ and A contains the nonempty subsets a of V such that the subset of the domain of K

$$\kappa(a) = \bigcap_{v\in a} \kappa(v) \; - \; \bigcup_{v\notin a} \kappa(v)$$

is not empty. The analytic part of the model consists of the system of linear constraints

$$\begin{cases} \sum_{a\in A(v)} x(a) = q(v) & (v \in V) \\ x(a) \in \sigma & (a \in A) \end{cases} \tag{1}$$

where $A(v) = \{a \in A\colon v \in a\}$. Here, variable $x(a)$ stands for the unknown (to the users) sum of S over the set of tuples selected by the predicate "K in $\kappa(a)$". Thus, a "snooper" can compute the two quantities $\min_X x(a)$ and $\max_X x(a)$ for each hyperarc a of G, where X is the solution set of constraint system (1). In order to decide if the response to $Q(n)$ leads to the disclosure of some sensitive sum, the auditor will compute the actual value $s(a)$ of each $x(a)$ and apply the sensitivity criterion in use to decide if $s(a)$ is sensitive; next, for each a having $s(a)$ sensitive, it will apply the safety test which checks that

$$\min_X x(a) < (1-p)\, s(a) \qquad \text{or} \qquad \max_X x(a) > (1+p)\, s(a)$$

where p is the protection level in use. If each sensitive sum $s(a)$ is protected at level p, then (and only then) the auditor will decide that $Q(n)$ can be answered safely.

Suppose that the weighted hypergraph (G, s), where s is the function on A that assigns to each hyperarc a of G the corresponding sum $s(a)$, has been constructed and suppose that a number of sensitive weights have been found (for details see [10, 12]). What remains to do is testing each sensitive sum $s(a)$ for safety, which requires computing $\min_X x(a)$ and $\max_X x(a)$. In what follows, the weighted hypergraph (G, s) will be referred to as the *map* of $\{Q(1), \ldots, Q(n)\}$ and the two quantities $\min_X x(a)$ and $\max_X x(a)$ are called the *tightest lower bound* and the *tightest upper bound* on the weight of the hyperarc a of the map (G, s), respectively.

Example 1. Consider a relation with scheme {NAME, DEPARTMENT, SALARY}. We make the following assumptions:

— the attribute SALARY is confidential and its domain is the set of nonnegative real numbers,
— the attribute DEPARTMENT is the only category attribute and its domain is {A, B, C, D, E, F},
— the sums of SALARY over the employees in the single departments are as follows: 15.0 for A, 9.0 for B, 7.5 for C, 6.5 for D, 6.0 for E and 5.5 for F,
— the sum of SALARY 15.0 for A is the only sensitive sum,
— the protection level in use is $p = 0$.

Consider four sum queries on SALARY specified by the following category sets: $\kappa(1)$ = {A, B}, $\kappa(2)$ = {A, C, D}, $\kappa(3)$ = {B, C, F} and $\kappa(4)$ = {D, E}. The map of the four sum queries can be pictured as a graph (see Figure 1) whose arcs correspond to κ(1, 2) = {A}, κ(1, 3) = {B}, κ(2, 3) = {C}, κ(2, 4) = {D}, κ(3, 3) = {F} and κ(4, 4) = {E}.

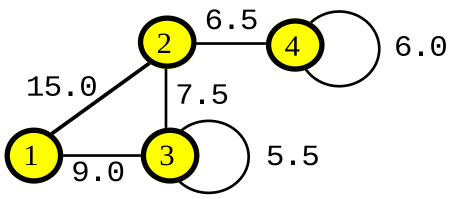


Fig. 1

The only sensitive weight is that of the arc (1, 2). Using standard algebraic methods, one finds that the tightest bounds on its weight are 9.25 and 24. So, the set of the four sum queries is safe. Suppose now that a fifth sum query arrives with category set κ(5) = {E, F}. The new query map is shown in Figure 2.

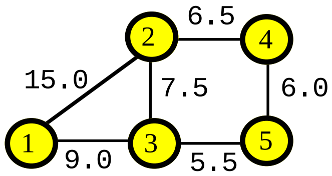


Fig. 2

Now, the tightest bounds on the weight of the arc (1, 2) coincide both with its weight. Consequently, the auditor will leave the fifth query unanswered. ■

What makes the work of the auditor hard is that the number of hyperarcs of the map of $\{Q(1), \ldots, Q(n)\}$ may be exponential in n. To overcome this difficulty, a query-overlap restriction [1] can be introduced which, for a fixed positive integer r, requires that the response to the (current) sum query $Q(n)$ is soon denied if in $\{Q(1), \ldots, Q(n)\}$ there is a *nonempty overlap of order* r, that is, if there exists a subset $\{Q(v)\colon v \in V'\}$ of size r such that $\bigcap_{v\in V'} \kappa(v) \neq \emptyset$. The simplest nontrivial case is $r = 2$ [10, 12], which implies that the map of $\{Q(1), \ldots, Q(n)\}$ is a graph (where loops are allowed) as in Example 1.
In this paper we address the problem of computing the tightest bounds on the weight of a graph under the assumption that the arc weights are nonnegative real numbers. This problem can be solved using linear programming methods [5, 6]. A more efficient approach consists in transforming that into a network flow problem and Gusfield [7] proved that, in the bipartite case, the tightest bounds on the weight of an arc can be computed with two maximum-flow computations. Here we shall show that for a nonbipartite graph the tightest bounds on the weight of an arc can be found with two or four maximum-flow computations depending on whether the arc is or is not a loop.
The paper is organised as follows. In Section 2 we review the results on bipartite maps. In Section 3 we present a bipartite transformation of a nonbipartite map. In Section 4 we present the maximum-flow computations for the tightest bounds on the weight of an arc of a nonbipartite map. Section 5 deals with a special case where there are closed formulas for the tightest bounds on the weight of each arc. Section 6 contains some open problems.

2 The bipartite case

In this section, we review the maximum-flow technique proposed by Gusfield [7] to compute the tightest bounds on the weight of a given arc of a map (G, s) where $G = (V, A)$ is a bipartite, connected graph. Let $\{V_1, V_2\}$ be the vertex bipartition of G. First, a flow network [2] is built up as follows. Let $q(v) = \sum_{a\in A(v)} s(a)$ for each v in V, and let M be a finite number larger than $max\ \{q(v)\colon v \in V_1\}$. First, each arc (u, v) of G is directed both from V_1 to V_2 and from V_2 to V_1. Then, the capacity of each arc $u \to v$ is set to M if $u \in V_1$, and to $s(u, v)$ otherwise. We denote the resulting flow network by $N(G, s; V_1, M)$.

Example 2. Consider the map (G, s) shown in Figure 3.

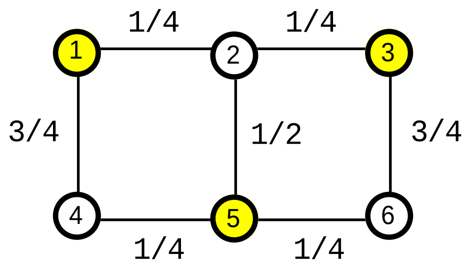


Fig. 3. A bipartite map

Let $V_1 = \{2, 4, 6\}$ and $V_2 = \{1, 3, 5\}$, and let $M = 2$. Figure 4 shows the network $(G, s; V_1, M)$.

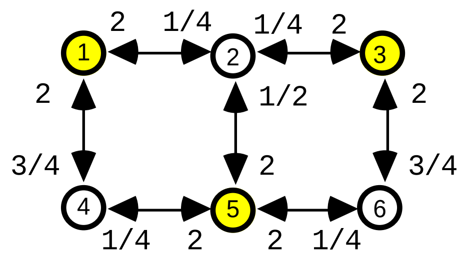


Fig. 4. The network associated with the map of Figure 3 ■

Let $f_{u,v}$ be a maximum flow in $N(G, s; V_1, M)$ from vertex u to vertex v, and let $F_{u,v}$ be the value of $f_{u,v}$, that is,

$$F_{u,v} = \sum_w f_{u,v}(u \to w) - \sum_w f_{u,v}(w \to u) = \sum_w f_{u,v}(w \to v) - \sum_w f_{u,v}(v \to w)$$

Note that if $u \in V_1$ then

$$F_{u,v} = M + \sum_{w\in V_2-\{v\}} f_{u,v}(u \to w) - \sum_{w\in V_2-\{v\}} f_{u,v}(w \to u)\ .$$

The following three propositions were proved by Gusfield [7]. Recall that X is the set of solutions of constraint system (1) where σ is the set of nonnegative reals.

Proposition 1. Given a map (G, s) where G is a bipartite connected graph with bipartition $\{V_1, V_2\}$, let a be an arc of G with end-points $u \in V_1$ and $v \in V_2$. Then

$$\min_X x(a) = \max\ \{0, s(a) + M - F_{u,v}\} \qquad \text{and} \qquad \max_X x(a) = F_{v,u}$$

where $F_{u,v}$ and $F_{v,u}$ are the values of maximum flows in $N(G, s;\ V_1, M)$ from u to v and from v to u, respectively.

If $x \in X$ has $x(a) = \min_X x(a)$ ($x(a) = \max_X x(a)$, respectively), we call the map (G, x) an *a-minimal* (*a-maximal*, respectively) *variant* of the map (G, s).

Proposition 2. Given a map (G, s) where G is a bipartite connected graph with bipartition $\{V_1, V_2\}$, let a be an arc of G with end-points $u \in V_1$ and $v \in V_2$.

(*i*) Given a maximum flow $f_{u,v}$ in $N(G, s;\ V_1, M)$ from u to v, an a-minimal variant (G, x) of (G, s) can be constructed as follows. For each arc a' of G with end-points $u' \in V_1$ and $v' \in V_2$ take

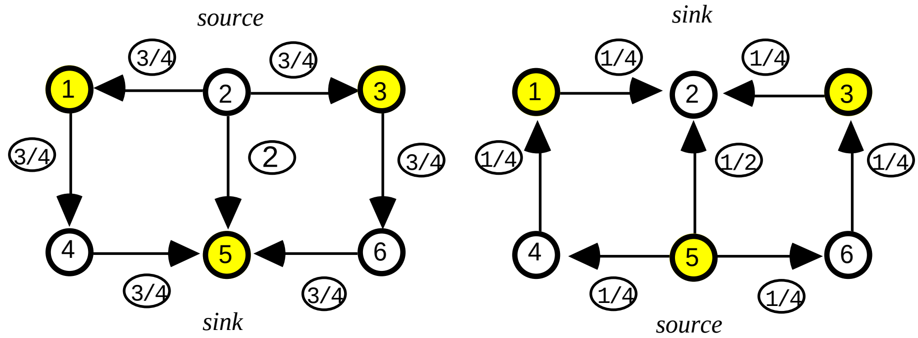
$$x(a') = \begin{cases} \max\{0, s(a) + M - F_{u,v}\} & \text{if } a' = a \\ s(a') + \tau[f_{u,v}(u' \to v') - f_{u,v}(v' \to u')] & \text{else} \end{cases}$$

where $\tau = \min\left\{1,\ \dfrac{s(a)}{F_{u,v} - M}\right\}$.

(*ii*) Given a maximum flow $f_{v,u}$ in $N(G, V_1, V_2;\ s,\ M)$ from v to u, an a-maximal variant (G, x) of (G, s) can be constructed as follows. For each arc a' of G with end-points $u' \in V_1$ and $v' \in V_2$ take

$$x(a') = \begin{cases} F_{v,u} & \text{if } a' = a \\ s(a') + f_{v,u}(u' \to v') - f_{v,u}(v' \to u') & \text{else} \end{cases}$$

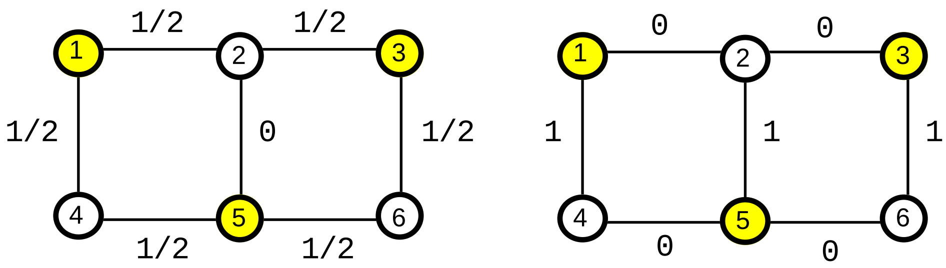
Example 2 (continued). Figure 5(*a*) shows a maximum flow $f_{2,5}$ from the vertex 2 to the vertex 5 and Figure 5(*b*) shows a maximum flow $f_{5,2}$ from the vertex 5 to the vertex 2. So, $F_{2,5} = 7/2$ and $F_{5,2} = 1$.



(*a*) (*b*)

Fig. 5.

By Proposition 1, the tightest lower and upper bounds on the weight of the arc (2, 5) are 0 and 1, respectively. Using Proposition 2, we obtain a (2, 5)-minimal variant and a (2, 5)-maximal variant of the map of Figure 3, which are shown in Figures 6(*a*) and 6(*b*) respectively.



(*a*) (*b*)

Fig. 6. Minimal and maximal variants ■

Proposition 3. Given a map (G, s) where G is a complete bipartite graph with bipartition $\{V_1, V_2\}$, let $q(v) = \sum_{a\in A(v)} s(a)$, $v \in V$, and let $N = \sum_{v\in V_1} q(v)$. For each arc a of G with end-points u and v, one has

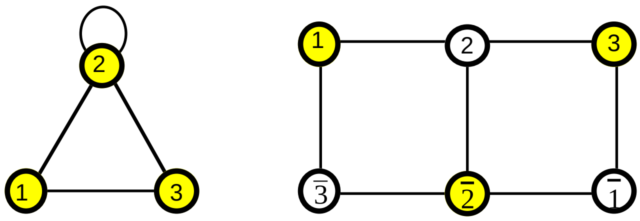
$$\min_X x(a) = \max\{0, q(u) + q(v) \pm N\} \qquad \max_X x(a) = \min\{q(u), q(v)\}.$$

3 A bipartite map associate with a nonbipartite map

Consider now a nonbipartite, connected graph $G = (V, A)$. The arcs of G that are not loops will be referred to as *links*. With G we associate a bipartite, connected graph $H = (W, B)$ with $|W| = 2|V|$ and $|B| = 2|A|-l$, where l is the number of loops of G. The graph H, which will be referred to as a *bipartite transform* of G [9], is constructed as follows. Let $\overline{V}$ = $\{\overline{v} : v \in V\}$ be a "copy" of V (that is, $\overline{V} \cap V = \emptyset$). The vertex set of H is taken to be $W = V \cup \overline{V}$. The arc set B of H is defined as follows. Arbitrarily chosen a spanning tree T of G, let G' be the bipartite graph obtained from T by adding all nontree arcs that do not create odd cycles. Take $B = \cup_{a\in A} \beta(a)$ where β is the function defined on A as follows:

$$\beta(u, v) = \begin{cases} \{(u,v),(\bar{u},\bar{v})\} & \text{if } (u,v) \text{ is an edge of } G' \\ \{(u,\bar{v}),(\bar{u},v)\} & \text{if } (u,v) \text{ is a link of } G \text{ but not of } G' \\ \{(u,\bar{u})\} & \text{if } u = v \text{ and } (u,u) \text{ is a loop of } G \end{cases}$$

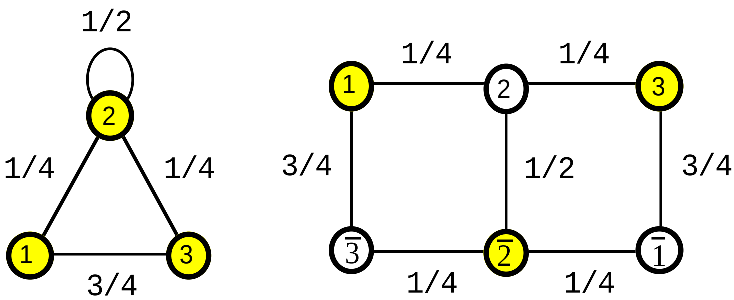
For vertex w of H, if $w = v$ or $w = \bar{v}$, we say that v is the vertex of G corresponding to w; similarly, for arc b of H, if b belongs to $\beta(a)$, we say that a is the arc of G corresponding to b. Let $\{V_1, V_2\}$ be the bipartition of the vertex set of G' and let $\overline{V_i}$ = $\{\bar{v} : v \in V_i\}$, i = 1, 2. Note that H is a bipartite, connected graph with bipartition $\{W_1, W_2\}$ where $W_1 = V_1 \cup \overline{V_2}$ and $W_2 = \overline{V_1} \cup V_2$.



(a) (b)

Fig. 7. A nonbipartite graph and a bipartite transform ■

Given a nonbipartite, connected map (G, s), let $H = (W, B)$ be a bipartite transform of G and let t be the function defined on B with $t(b) = s(a)$ if a is the arc of G corresponding to b. We call (H, t) a bipartite map *associated* with (G, s).



(a) (b)

Fig. 8. (a) A nonbipartite map; (b) an associated bipartite map ■

For vertex w of H, let $r(w) = \sum_{b \in B(w)} t(b)$ where $B(w)$ is the set of arcs incident to w. Let Y be the set of nonnegative real-valued solutions of the equation system

$$\sum_{b \in B(w)} y(b) = r(w) \qquad (w \in W) \qquad (2)$$

The following two obvious facts show that Y is closely connected with X.

Fact 1. For every x in X, the function y on B with $y(b) = x(a)$, where a is the arc of G corresponding to b, belongs to Y.

Fact 2. For every y in Y, the function x on A with

$$x(a) = \begin{cases} y(b) & \text{if } a \text{ is a loop of } G \text{ with } \beta(a) = \{b\} \\ \frac{1}{2}[y(b') + y(b'')] & \text{if } a \text{ is a link of } G \text{ with } \beta(a) = \{b', b''\} \end{cases}$$

belongs to X.

4 Bounds on the weight of an arc

In this section we show that the tightest bounds on the weight of an arc of a nonbipartite map can be computed with two or four maximum-flow computations depending whether the arc is a loop or a link. The following is an immediate consequence of Facts 1 and 2.

Lemma 1. Let (G, s) be a nonbipartite map, and let (H, t) be a bipartite map associated with (G, s). For each arc a of G,

(i) if a is a loop with $\beta(a) = \{b\}$, then the tightest lower and upper bounds on the weight of a coincide with the tightest lower and upper bounds on the weight of b, respectively;

(ii) if a is a link with $\beta(a) = \{b', b''\}$, then the tightest lower and upper bounds on the weight of a are respectively equal to

$$\frac{1}{2} \min_Y [y(b') + y(b'')] \quad \text{and} \quad \frac{1}{2} \max_Y [y(b') + y(b'')].$$

By part (i) of Lemma 1, the tightest bounds on the weight of a loop of (G, s) coincide with the tightest bounds on the weight of the corresponding arc of (H, t) and, hence, by Proposition 1 they can be determined with two maximum-flow computations.

Example 3. Consider the nonbipartite map of Figure 8(a). We apply Part (i) of Lemma 1 to compute the tightest bounds on the weight of the loop (2, 2). The associated bipartite map of Figure 8(b) looks like the map of Figure 3. Therefore, the tightest bounds on the weight of the loop (2, 2) of the map of Figure 8(a) coincide with the tightest bounds on the weight of the arc (2, 5) of the map of Figure 3, which were 0 and 1. ■

Consider now the case of a link a of (G, s) with $\beta(a) = \{b', b''\}$. By Part (ii) of Lemma 1, we have to compute

$$\min_Y [y(b') + y(b'')] \quad \text{and} \quad \max_Y [y(b') + y(b'')].$$

We shall show that they can be obtained as follows: The minimum (maximum, respectively) of the function $y(b') + y(b'')$ over Y equals the tightest lower (upper, respectively) bound on the weight of the arc b' of H plus the tightest lower (upper, respectively) bound on the weight of the arc b'' of the map that is obtained from a b'-minimal (b'-maximal, respectively) variant of (H, t) by deleting b'.

We now prove the correctness for $\max_Y [y(b') + y(b'')]$. The proof of the correctness for $\min_Y [y(b') + y(b'')]$ is similar.

Consider the bipartite graph $H = (W, B)$. A *circulation* in H is any real-valued solution z of the homogeneous equation system associated with equation system (2); that is,

$$\sum_{b\in B(w)} z(b) = 0 \qquad\qquad (w \in W)$$

The *signed support* of circulation z is the ordered couple (S^+, S^-) where $S^+ = \{b \in B\colon z(b) > 0\}$ and $S^- = \{b \in B\colon z(b) < 0\}$, and the *support* of z is the set $S^+ \cup S^-$. A nonzero circulation z in H is *minimal* if there is no circulation whose support is a proper subset of the support of z. It is well-known from linear algebra that, for every (nonzero) circulation z and for each b with $z(b) \neq 0$, there exists a minimal circulation ζ such that b belongs to the support of ζ and, if (S^+, S^-) and (C^+, C^-) are respectively the signed supports of z and ζ, then $C^+ \subseteq S^+$ and $C^- \subseteq S^-$. Finally, since H is a bipartite graph, the supports of minimal circulations are all and the only cycles (i.e., simple circuits) of H (e.g., see [4]). To sum up, one has the following

Fact 3. Let z be a circulation in H with signed support (S^+, S^-), and let b be an element of the support of z. Then b lies in a simple cycle C which is the support of a minimal circulation in H having $(C\cap S^+, C\cap S^-)$ as its signed support.

Lemma 2. Let (H, t) be a bipartite map and let b' and b'' be two arcs of H. There exists a nonnegative real-valued solution y of equation system (2) that maximises both $y(b')$ and $y(b') + y(b'')$.

Proof. Let (H, y_o) be a b'-maximal variant of (H, t). Moreover, let Y_1 be the set of nonnegative real-valued solutions of equation system (2) that maximise $y(b') + y(b'')$, and let y_1 be in Y_1 and such that $y_1(b') \geq y(b')$ for every $y \in Y_1$. Of course, one has $y_1(b') \leq y_o(b')$. We shall prove that $y_1(b') = y_o(b')$. Consider the circulation $z = y_o - y_1$, and let (S^+, S^-) be the signed support of z. Suppose, by contradiction, that $y_1(b') < y_o(b')$. Then b' belongs to S^+. By Fact 3, b' lies in a cycle C which is the support of a minimal circulation having signed support (C^+, C^-), where $C^+ = C\cap S^+$ and $C^- = C\cap S^-$. Such a minimal circulation can be explicitly constructed as follows. Let

$$\varepsilon = \min\ \{|c(b)|\colon b \in C^-\}$$

and let ζ be the circulation with

$$\zeta(b) = \begin{cases} +\varepsilon & b \in C^+ \\ -\varepsilon & b \in C^- \\ 0 & \text{else} \end{cases}$$

Let $y_2 = y_1 + \zeta$. Of course, y_2 is a solution of equation system (2). Indeed, y_2 is nonnnegative everywhere because, for each arc b of H, if b is not in C^-, then

$$y_2(b) = y_1(b) + \zeta(b) \geq y_1(b) \geq 0;$$

otherwise,

$$y_2(b) = y_1(b) - \varepsilon \geq y_1(b) + z(b) = y_o(b) \geq 0\ .$$

We now show that from the foregoing a contradiction always follows. Consider the following three cases that can occur for b'':

Case 1. b'' is in C^+. Then $y_2(b'') = y_1(b'') + \varepsilon$ which leads to the following

$$y_2(b') + y_2(b'') = y_1(b') + y_1(b'') + 2\varepsilon > y_1(b') + y_1(b'')$$

which contradicts the membership of y_1 in Y_1.

Case 2. b'' is in C^-. Then $y_2(b'') = y_1(b'') - \varepsilon$ and, hence, one has

$$y_2(b') + y_2(b'') = y_1(b') + \varepsilon + y_1(b'') - \varepsilon = y_1(b') + y_1(b'')$$

so that y_2 belongs to Y_1. But, since b' is in C^+, one has

$$y_2(b') = y_1(b') + \zeta(b') = y_1(b') + \varepsilon > y_1(b')$$

which contradicts the choice of y_1.

Case 3. $b'' \notin C^+ \cup C^-$. Then $y_2(b'') = y_1(b'')$ which leads to the following

$$y_2(b') + y_2(b'') = y_1(b') + \varepsilon + y_1(b'') > y_1(b') + y_1(b'')$$

which contradicts the membership of y_1 in Y_1. □

The following is an immediate consequence of Lemma 2.

Lemma 3. The maximum of the function $y(b') + y(b'')$ over the set Y of nonnegative real-valued solutions of equation system (2) equals the tightest upper bound on the weight of the arc b' of (H, t) plus the tightest upper bound on the weight of the arc b'' of the map obtained from a b'-maximal variant of (H, t) by deleting b'.

Combining Lemmas 1 and 3 with Proposition 1, we obtain the following algorithm which, given a bipartite map (H, t) of (G, s) and a link a of G with $\beta(a) = \{b', b''\}$, computes the tightest upper bound on the weight of a link a of a nonbipartite map (G, s). The input data are:

(H, t)	a bipartite map associated with (G, s)
$\{W_1, W_2\}$	the bipartition of H
$N(H, t; W_1, M)$	a network associated with (H, t)

$b' = (w_1', w_2')$ with $w_1' \in W_1$ and $w_2' \in W_2$, $b'' = (w_1'', w_2'')$ with $w_1'' \in W_1$ and $w_2'' \in W_2$.

Algorithm MAX

(1) Find a maximum flow f in $N(H, t; W_1, M)$ from w_2' to w_1', and let F be the value of f.

(2) Given f and using Proposition 2(*ii*), construct a b'-maximal variant (H, y) of (H, t).

(3) Let (H', t') be the map obtained from (H, y) by deleting b'. Find a maximum flow f' in $N(H', t'; W_1, M)$ from w_2'' to w_1'', and let F' be the value of f'.

(4) Set the tightest upper bound on the weight of a to $\dfrac{F+F'}{2}$.

Analogously, the following algorithm correctly computes the tightest lower bound on the weight of a.

Algorithm MIN

(1) Find a maximum flow f in $N(H, t; W_1, M)$ from w_1' to w_2'.

(2) Given f and using Proposition 2(i), construct a b'-minimal variant (H, y) of (H, t).

(3) Let (H', t') be the map obtained from (H, y) by deleting b'. Find a maximum flow f' in $N(H', t'; W_1, M)$ from w_1'' to w_2'', and let F' be the value of f'.

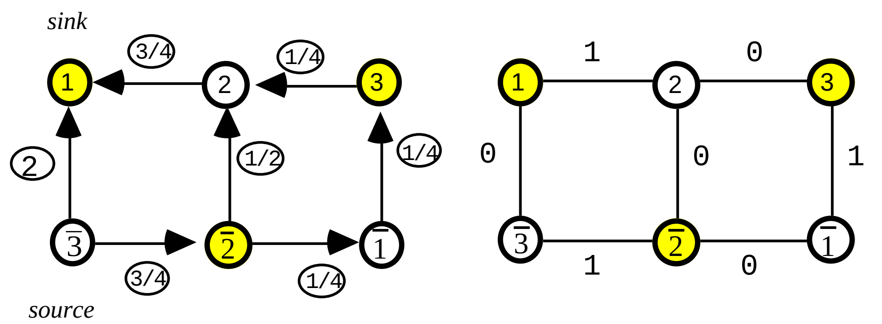
(4) Set the tightest lower bound on the weight of a to

$$\frac{max\ \{0, t(b') + M - F\}\ +\ max\ \{0, t'(b'') + M - F'\}}{2}.$$

To sum up we have the following

Theorem 1. The tightest bounds of an edge of a nonbipartite map can be found with two or four maximum-flow computations depending on whether the arc is a loop or a link.

Example 3 (continued). We now apply the procedure above to compute the tightest lower and upper bounds on the weight of the link (1, 3) of the nonbipartite map of Figure 8(a). Recall that this arc corresponds to the two arcs ($\bar{3}$, 1) and ($\bar{1}$, 3) of the associated bipartite map shown in Figure 8(b), and that the vertices $\bar{1}$ and $\bar{3}$ are on the side W_1 and the vertices 1 and 3 are on the side W_2.
We first apply Algorithm MIN. Figure 9(a) shows a maximum flow in the network associated with the bipartite map shown in Figure 8(b) from the vertex $\bar{3}$ to the vertex 1, and Figure 9(b) shows the corresponding ($\bar{3}$, 1)-minimal variant of the map of Fig. 8(b).

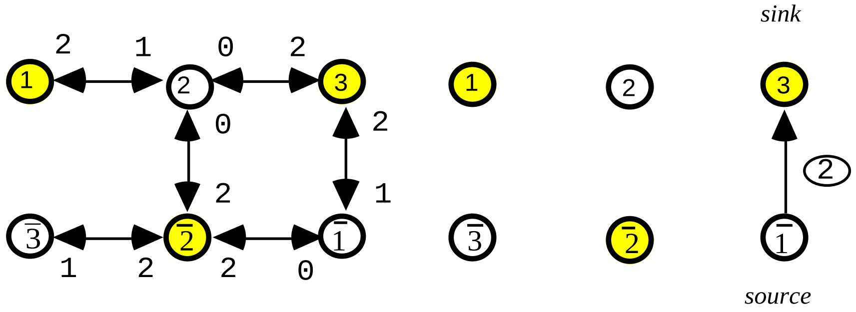


(a) (b)

Fig. 9

So, the tightest lower bound on the weight of the arc ($\bar{3}$, 1) is max $\{0, \frac{3}{4}+2-\frac{11}{4}\} = 0$.
Figure 10(a) shows the network associated with the bipartite map of Figure 9(b) with the

arc ($\bar{3}$, 1) deleted, and Figure 10(b) shows a maximum flow in this network from the vertex 3 to the vertex $\bar{1}$.

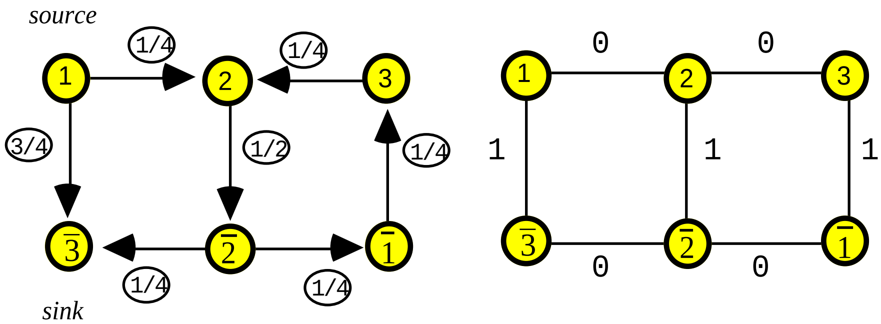


(a) (b)

Fig. 10

So, the tightest lower bound on the weight of the arc ($\bar{1}$, 3) is max {0, 1+2−2} = 1, and the tightest lower bound on the weight of its corresponding arc (1, 3) is (0+1)/2 = $\frac{1}{2}$.

We now apply Algorithm MAX. Figure 11(a) shows a maximum flow in the network associated with the bipartite map shown in Figure 8(b) from the vertex 1 to the vertex $\bar{3}$, and Figure 11(b) shows the corresponding ($\bar{3}$, 1)-maximal variant of the map of Fig. 8(b).



(a) (b)

Fig. 11

So, the tightest upper bound on the weight of the arc ($\bar{3}$, 1) is equal to 1. Figure 12(a) shows the network associated with the bipartite map of Figure 11(b) with the arc ($\bar{3}$, 1) deleted, and Figure 12(b) shows a maximum flow in this network from the vertex 3 to the vertex $\bar{1}$.

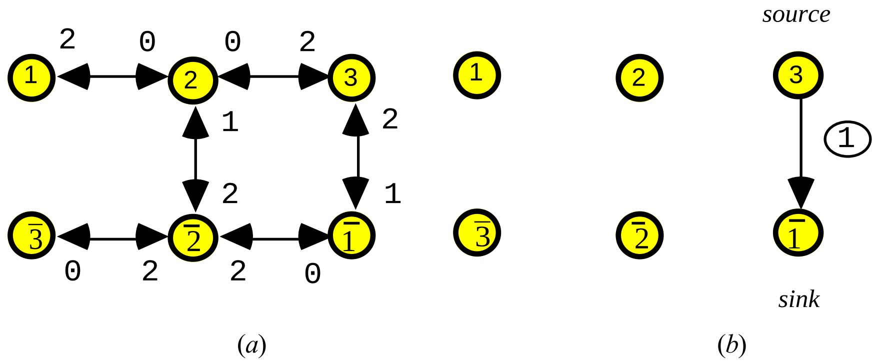


Fig. 12

So, the tightest upper bound on the weight of the arc ($\overline{1}$, 3) is 1, and the tightest upper bound on the weight of the arc (1, 3) is equal to (1+1)/2 = 1. ■

5 A special Case

In this section, we consider the special case of a complete graph with the addition of one loop for each vertex. We now prove that the tightest bounds on the weight of an edge can be computed with a number of arithmetic operations and comparisons proportional to the number of vertices.

Let (G, s) be a map where $G = (V, A)$ is a complete graph with the addition of one loop for each vertex. Let $q(v) = \sum_{a\in A(v)} s(a)$ for each v in V. The constraint system (1) always admits the solution x (see Figure 13) with

$$x(u, v) = \begin{cases} q(v) & \text{if } u = v \\ 0 & \text{else} \end{cases}$$

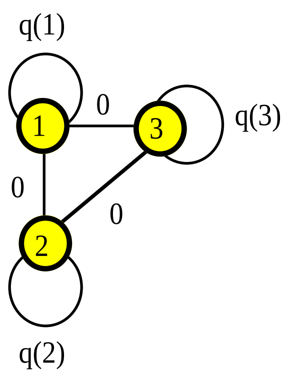


Fig. 13

Moreover, given a link (u^*, v^*) of G with $q(u^*) \le q(v^*)$, then the constraint system (1) always admits the solution x (see Figure 14) with

$$x(u, v) = \begin{cases} q(u^*) & \text{if } (u,v) = (u^*,v^*) \\ 0 & \text{else} \end{cases}$$

for each link (u, v), and

$$x(u, u) = \begin{cases} 0 & \text{if } u = u^* \\ q(v^*) - q(u^*) & \text{if } u = v^* \\ q(u) & \text{else} \end{cases}$$

for each loop (u, u).

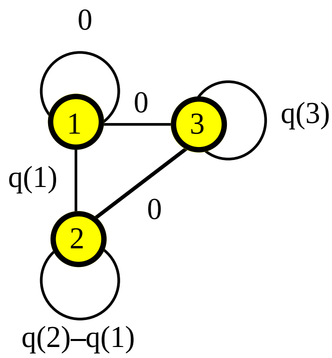


Fig. 14

It follows that the tightest upper bound on the weight of loop (u, u) is equal to $q(u)$, and the tightest lower and upper bounds on the weight of link (u, v) are 0 and *min* $\{q(u), q(v)\}$, respectively. What remains is a formula for the tightest lower bound on the weight of loop (u, u). We now prove that it is given by *max* $\{0, 2q(u) - N\}$, where $N = \sum_{v\in V} q(v)$. Let (H, t) be a bipartite map associated with (G, s) so that the loop (u, u) corresponds to the edge $(u, \bar{u})$ of H. For each vertex w of H, let $r(w) = q(v)$ if v is the vertex of G corresponding to w. By Proposition 3, the tightest lower bound on the weight of $(u, \bar{u})$ is

$$\max \{0, r(u) + r(\bar{u}) - \textstyle\sum_{w\in W_1} r(w)\} .$$

But $r(u) = r(\bar{u}) = q(u)$ and $\sum_{w\in W_1} r(w) = \sum_{v\in V} q(v) = N$ so that the statement follows from part (*i*) of Lemma 2. To sum up, we have the following result.

Theorem 2. Let (G, s) be a map where G is a complete graph with the addition of one loop for each vertex. Let $q(v) = \sum_{a\in A(v)} s(a)$ for each v in V, and let $N = \sum_{v\in V} q(v)$. Then,

(*i*) the tightest lower and upper bounds on the weight of loop (u, u) are respectively *max* $\{0, 2q(u) - N\}$ and $q(u)$;

(*ii*) the tightest lower and upper bounds on the weight of link (u, v) are respectively 0 and *min* $\{q(u), q(v)\}$.

6 Future research

We considered graphs with arcs weighted by nonnegative reals. The case of integral weights is open; however, since the incidence matrix of a bipartite graph is totally unimodular, the integrality constraint can be relaxed and the results recalled in Section 2 still hold. Another direction for future research should cover the general case of weighted hypergraphs. It is worth mentioning that, for hyperarc weights from $\{0, 1\}$, deciding if there exists a hyperarc whose weight is uniquely determined is a co*NP*-complete problem [8].

References

1. Adam, N.R., Wortmann, J.C.: Security control methods for statistical databases: a comparative study. ACM Computing Surveys **21** (1989) 515-556.
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows. Prentice Hall, Englewood Cliffs, 1993.
3. Chin, F.Y., Ozsoyoglu, G.: Auditing and inference control in statistical databases. IEEE Trans. on Software Engineering **8** (1982) 574-582.
4. Conforti, M., Rao, M.R.: Some new matroids on graphs: cut sets and the max cut problem. Mathematics of Operations Research **12** (1987) 193-204.
5. Cox, L.H.: Suppression methodology and statistical disclosure control. J. American Statistical Association **75** (1980) 377-385.
6. Cox, L.H., Zayatz, L.V.: An agenda for research on statistical disclosure limitation. J. Official Statistics **11** (1995) 205-220.
7. Gusfield, D.: A graph-theoretic approach to statistical data security. SIAM J. Computing **17** (1988) 552-571.
8. Kleinberg, J.M., Papadimitriou, C.H., Raghavan, P.: Auditing Boolean attributes. Proc.XIX ACM Symp. on "Principles of Database Systems" (2000) 86-91.
9. Malvestuto, F.M., Mezzini, M.: A linear algorithm for finding the invariant edges of an edge-weighted graph. SIAM J. on Computing **31** (2002) 1438-1455.
10. Malvestuto, F.M., Mezzini, M.: On the hardness of protecting sensitive information in a statistical database. Proc. World Multiconference on "Systemics, Cybernetics and Informatics", vol. XIV (2001) 504-509.
11. Malvestuto, F.M., Moscarini, M.: Query evaluability in statistical databases. IEEE Transactions on Knowledge and Data Engineering **2** (1990) 425-430.
12. Malvestuto, F.M., Moscarini, M.: An audit expert for large statistical databases. In Statistical Data Protection. EUROSTAT (1999) 29-43.
13. Willenborg, L., de Waal, T.: Statistical Disclosure Control in Practice. Lecture Notes in Statistics, Vol. 111. Springer-Verlag, New York (1996).
14. Willenborg, L., de Waal, T.: Elements of Statistical Disclosure. Lecture Notes in Statistics, Vol. 155. Springer-Verlag, New York (2000).