Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 176 (2007) 691–706

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Discrete Optimization

Estimation of flows in flow networks

Ron Zohar *, Dan Geiger

Department of Computer Science, Technion, 32000 Haifa, Israel

Received 23 August 2004; accepted 3 August 2005
Available online 22 November 2005

Abstract

Let G be a directed graph with an unknown flow on each edge such that the following flow conservation constraint is maintained: except for sources and sinks, the sum of flows into a node equals the sum of flows going out of a node. Given a noisy measurement of the flow on each edge, the problem we address, which we call the Most Probable Flow Estimation problem (MPFE), is to estimate the most probable assignment of flow for every edge such that the flow conservation constraint is maintained. We provide an algorithm called ΔY-mpfe for solving the MPFE problem when the measurement error is Gaussian (Gaussian-MPFE). The algorithm works in $O(|E| + |V|^2)$ when the underlying undirected graph of G is a 2-connected planar graph, and in $O(|E| + |V|)$ when it is a 2-connected serial-parallel graph or a tree. This result is applicable to any Minimum Cost Flow problem for which the cost function is $\tau_e(X_e - \mu_e)^2$ for edge e where μ_e and τ_e are constants, and X_e is the flow on edge e. We show that for all topologies, the Gaussian-MPFE's precision for each edge is analogous to the equivalent resistance measured in series to this edge in an electrical network built by replacing every edge with a resistor reflecting the measurement's precision on that edge.

Keywords: Quadratic programming; Estimation of network flows; Minimum cost flow problem; Gaussian; MPFE

1. Introduction

Let G be a directed graph with an unknown flow on each edge such that the following *flow conservation constraint* is maintained: the sum of flows into an internal node equals the sum of flows going out of an internal node. Given a noisy set of measurements of the flow on each edge and assuming an independent measurement error, the problem we address, which we call the Most Probable Flow Estimation problem (MPFE), is to estimate the most probable assignment of flow for every edge such that the flow conservation constraint is maintained.

* Corresponding author.
E-mail addresses: ronron@cs.technion.ac.il (R. Zohar), dang@cs.technion.ac.il (D. Geiger).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.08.009

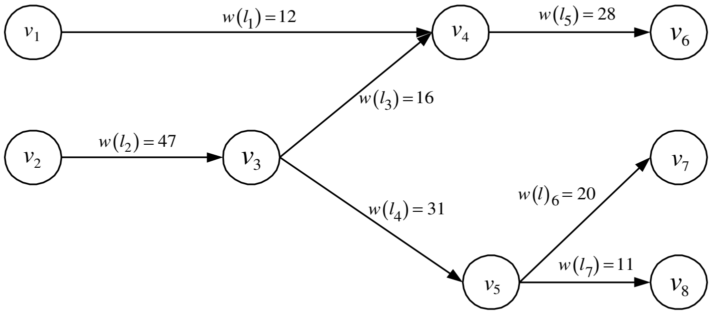


Fig. 1. Flow network.

Most probable flow estimation is appropriate whenever a coherent estimation of flow is desired. For example, estimation of currents in electrical networks, traffic estimation in roads, throughput estimation in computer networks and vehicle number estimation in group tracking scenarios, where edges represent tracks in an arena and internal nodes correspond to the merger or separation of groups of vehicles (Zohar and Geiger, 2003). Consider for example the network in Fig. 1. The numbers on the edges represent the exact flows in the network. Typically, in real-world scenarios, the exact flows are unknown and only a set of noisy measurements is at hand. In such scenarios, MPFE provides a mechanism for better utilizing individual independent sensors by employing the flow conservation constraints.

Notably, MPFE is a special case of the Minimum Cost Flow problem (Ahuja et al., 1993). The MPFE problem also relates to finding inference procedures for probabilistic networks over continuous random variables (Cowell et al., 1999). The MPFE problem was originally defined in Zohar and Geiger (2003), along with an $O(|E| + |V|)$ algorithm for solving the problem when the measurement error is Gaussian (Gaussian-MPFE), on networks whose underlying undirected structure is a tree.

We provide an algorithm called ΔY-mpfe for solving Gaussian-MPFE in $O(|E| + |V|^2)$ when the underlying undirected graph of G is a 2-connected planar graph, and in $O(|E| + |V|)$ when it is a 2-connected serial-parallel graph or a tree. This result is applicable to any Minimum Cost Flow problem for which the cost function is $\tau_e(X_e - \mu_e)^2$ for edge e where μ_e and τ_e are constants, and X_e is the flow on edge e. For general graphs, the problem can be solved in $O(|E|^3)$ steps by general techniques, as described in Section 3. Our algorithm consists of the application of a sequence of transformations on an input flow network G.

The rest of this paper is organized as follows. We first provide a formal description of the flow estimation problem (Section 2) and discuss related problems and past solutions (Section 3). Then, we review a set of graph transformations (Section 4), present the algorithm, and prove its correctness (Sections 5 and 6). We show that for all topologies, the Gaussian-MPFE's precision for each edge is analogous to the equivalent resistance measured in series to this edge in an equivqlent electrical network (Section 7) and conclude with a discussion (Section 8).

2. Problem formulation

A *flow network* is a weighted directed graph $G = (V, E, w)$ where V is a set of nodes, E is a set of m directed edges, and $w : E \to R$ is a *flow function* which assigns for every edge $(u, v) \in E$ a *flow* $w(u, v)$. This flow is said to be *directed* from u to v whereas the flow in the opposite direction is defined to have an opposite sign, that is, $w(v, u) \equiv -w(u, v)$. In the following, we also denote an edge (u, v) by uv. A flow on a set of edges Q is defined via $w(Q) = \sum_{l \in Q} w(l)$. An *internal node* is a node that has some edges coming into it and some that are leaving it. A *source* is a node with a single outgoing edge and no incoming edges and a *sink* is a node with a single incoming edge and no outgoing edges. Let S denote the set of sources and let T denote the set of sinks. Let $\alpha(v)$ and $\beta(v)$ denote the sets of edges into node v and out of node v, respectively. Then for every internal node,

$$\sum_{l\in\alpha(v)} w(l) = \sum_{l\in\beta(v)} w(l) \tag{2.1}$$

and

$$\sum_{s\in S} w(\beta(s)) = \sum_{t\in T} w(\alpha(t)). \tag{2.2}$$

These *flow conservation constraints* (aka. *Kirchhoff's law*) are linear and can be represented using an $n \times m$ *constraint matrix* A via the equation $AX = 0$, where n is the number of rows, one per internal node, $m = |E|$ is the number of columns, one per edge, $X_j = w(l_j)$ is the flow on edge l_j, and $X = (X_1, \ldots, X_m)$. In particular, for every internal node v_i and edge l_j, we set $A_{ij} = -1$ if edge l_j points to v_i, $A_{ij} = 1$ if edge l_j points away from v_i, and $A_{ij} = 0$ otherwise. For example, in Fig. 1, nodes v_3, v_4, v_5 are internal nodes, the flow conservation constraints are $X_2 = X_3 + X_4$, $X_1 + X_3 = X_5$, $X_4 = X_6 + X_7$ and the constraint matrix A is

$$\begin{pmatrix} 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 \end{pmatrix}.$$

We define the Most Probable Flow Estimation (MPFE) problem as follows:

Instance: Let $G = (V, E, \cdot)$ be a flow network constrained by $AX = 0$, where $X = (X_1, \ldots, X_m)$ is the (unknown) flow on each edge and A is the constraint matrix of the flows. Let e_i be the measured flow on the ith edge. Let the conditional probability $P(e_i|X_i)$ be an error term expressing the uncertainty in a measured flow e_i given the true flow X_i on an edge.

Query: Compute

$$X^*_G = \mathop{\arg\max}_{\substack{X \\ \text{s.t. } AX=0}} \prod_{i=1}^{m} P(e_i|X_i). \tag{2.3}$$

Our formulation of the problem is based on the assumption that the measurement on each edge is independent of measurements on other edges given the true flow on that edge.

This paper focuses on Gaussian-MPFE problems where the error term $P(e_i|X_i)$ is a normal distribution function given by

$$N(e_i|X_i, \tau_i) \equiv \sqrt{\frac{\tau_i}{2\pi}} \exp\left\{-\frac{\tau_i(X_i - e_i)^2}{2}\right\}, \tag{2.4}$$

where τ_i is the *precision* of the measurement on the ith edge. The precision is the reciprocal of the variance, namely $\tau_i \equiv \sigma_i^{-2}$. We denote an instance of the Gaussian-MPFE problem by a tuple (G, e, τ), where $G = (V, E, \cdot)$ is a flow network with unknown flows, e is a vector of flow measurements, and τ is a vector of corresponding measurement precisions. The selection of the error term as a normal distribution function is adequate for various applications and yields efficient algorithms as discussed in the following.

3. Related work

The Gaussian-MPFE problem can be solved for arbitrary topologies of flow networks in $O(|E|^3)$ steps by various well-known techniques which we now explicate.

First, the MPFE problem is a special case of the Minimal Cost Flow (MCF) problem (Ahuja et al., 1993). The input of an MCF problem is a directed graph with a cost function $G_e(f_e)$ of the flow f_e on each edge e. The output is an assignment of flows that minimizes the total cost $\sum_e G_e(f_e)$ and satisfies the flow conservation constraints. The MPFE problem is a special case of MCF because the cost function can be selected for edge l_i to be $-\log P(e_i \mid X_i)$. There are several efficient algorithms for solving MCF with convex cost functions which are applicable to the MPFE problem. For example, the capacity scaling algorithm works in $O((|E|\log U)(|E| + |V|\log|V|))$ where U is the maximal capacity in the network. For the Gaussian-MPFE problem, taking the logarithm of Eq. (2.3), using Eq. (2.4), yields a special form of a Minimum Cost Flow Problem, namely, an MCF problem where $G_e(X_e) = \tau_e\,(X_e - \mu_e)^2$. Such quadratic optimization problems can be solved exactly using Lagrange multipliers, in time complexity $O(|E|^3)$ for general graphs.

Another approach is Linear Minimum Variance Unbiased estimation (LMVU). It is based on the following theorem from Kay (1993) which provides a unified approach for the formulation and solution of diverse problems including Gaussian-MPFE as a special case:

Theorem 3.1. *If data can be modeled as*

$$x = H\theta + w,$$

where x is a vector of N observations, H is a known $N \times t$ matrix ($N > t$) of rank t, θ is a vector of t parameters to be estimated, and w is a noise vector with pdf $N(x|0, C^{-1})$ then the LMVU estimator of θ is

$$\hat{\theta} = (H^{\mathrm{T}}C^{-1}H)^{-1}H^{\mathrm{T}}C^{-1}x$$

and the (estimator's) covariance matrix is

$$C_{\hat{\theta}} = (H^{\mathrm{T}}C^{-1}H)^{-1}.$$

For the Gaussian-MPFE problem, the θ vector represents a set of t independent flows. Since the flow conservation constraints are linear, the flow on each edge is represented using $H\theta$. Since the computation of $\hat{\theta}$ and $C_{\hat{\theta}}$ requires matrix inversions, it requires $O(|E|^3)$ time complexity and $O(|E|^2)$ space complexity. The main disadvantage of this solution for the Gaussian-MPFE problem and the solution via Lagrange multipliers is the fact that these solutions do not use the topology of the graph and are therefore less efficient in some cases, as we demonstrate in this work. An advantage of this method is the computation of $C_{\hat{\theta}}$ which is not available via minimal cost flow algorithms. We return to this point in Section 7.

A third approach is to translate a given flow network into a Gaussian network (Cowell et al., 1999; Lauritzen, 1992; Shachter and Kenley, 1989; Lauritzen and Jensen, 2001). In Gaussian networks each variable v is normally distributed with a mean that is a linear function of the nodes that point into it and with a conditional variance that is fixed given the variables that point into v. This framework can be used to represent flow conservation constraints because, in the limit, when a variance of a variable tends to zero, the variable becomes a linear function of the variables that point to it. However, by transforming the flow network into a Gaussian network, some independence assertions are lost and therefore the clique-tree algorithm (Jensen, 1996) yields an $O(|E|^3)$ algorithm even in cases where an $O(|E| + |V|)$ algorithm can be found (Zohar and Geiger, 2003).

4. Series-parallel and ΔY transformations

We now concentrate on the class of $\Delta Y\Delta$-*reducible* graphs which include the classes of planar and serial-parallel graphs. The reduction algorithms for these graphs are used by our algorithm for efficiently solving the MPFE problem (Section 5).

The *underlying graph* of a directed graph G is an undirected graph having the same set of nodes and having one undirected edge xy for every directed edge xy or yx in G. Two nodes are *connected* in a directed graph G if there exists a path between them in the underlying graph of G. A *2-connected directed graph* is a directed graph whose underlying graph remains connected when a single node and its incident edges are removed. A *polytree* is a directed graph whose underlying graph is a tree. In a polytree there are possibly several sources, in contrast to directed trees that have one source called *root*. A *wye node* (Y) is a node of degree three. A *delta triangle* (Δ) is a cycle of three different nodes.

A 2-connected directed graph is *series-parallel reducible* if it can be reduced to a single edge by a sequence of the following transformations:

T0 (*edge-reversal*): Reverse the direction of an edge xy to yx.
T1 (*self-loop reduction*): Delete an edge l whose both endpoints are the same node.
T2 (*series reduction*): Delete a node y of degree two, remove its two incident edges xy and yz, and add a new edge xz.
T3 (*parallel reduction*): Delete one of a pair of parallel edges.

A 2-connected directed graph is called $\Delta Y\Delta$-*reducible* if it can be reduced to a single edge by transformations T0–T3 and the following:

T4 (*delta-wye transformation*): Delete the edges of a delta triangle (xy, xz, zy), and add a new node w and new edges xw, wy, and wz.
T5 (*wye-delta transformation*): Delete a wye node w and its three incident edges xw, wy, wz, and add a delta triangle (xy, xz, zy).

Transformations T0–T5 are depicted in Fig. 2.

The existence of an efficient recognition algorithm of $\Delta Y\Delta$-reducible graphs is an open question, but partial results are known. Every 2-connected planar graph is $\Delta Y\Delta$-reducible (Epifanov, 1966). The best known reduction algorithm for 2-connected planar graphs, named DWR, requires $O(|E| + |V|^2)$ operations (Feo and Provan, 1993). In this remarkable paper, it is conjectured that an $O(|E| + |V|^{1.5})$ algorithm is attainable. In Valdes et al. (1982) an $O(|E| + |V|)$ algorithm is given for recognizing and reducing 2-connected series-parallel reducible graphs. Clearly, not every 2-connected graph is $\Delta Y\Delta$-reducible because, for example, none of the transformations T1–T5 are applicable to any graph with no parallel edges such that each node is connected to at least four other nodes and where the minimum length of a cycle is four. Another example is K_6, namely, the full graph with 6 nodes, which can not be reduced by these transformations to a single edge (Warkentyne, 1988).

Some authors allow in their definition of $\Delta Y\Delta$-*reducible* graphs an additional transformation, named the *pendant edge reduction*, for reducing a node of degree 1 and its incident edge. This addition changes the class of $\Delta Y\Delta$-*reducible* graphs. However, for 2-connected graphs, the two classes of graphs are equivalent. Furthermore, although polytrees are not 2-connected directed graphs, the MPFE problem on a polytree G is reduced to an MPFE problem on a 2-connected directed graph by connecting all sources and sinks of G to a single node because the resulting graph is planar. On the other hand, if a planar graph G is not 2-connected, then connecting all sources and sinks of G to a single node may result in a non-planar graph. This is the reason for requiring the input graphs to be 2-connected. Finally, in some papers it is implicitly assumed that the original graph contains no parallel edges. This affects complexity calculation since an $O(|E|)$ process is required for transforming a given graph to a graph with no parallel edges, using T3. Our algorithm allows an input graph to contain parallel edges and therefore we have adjusted the complexity statements accordingly in our presentation of previous reduction algorithms, replacing $O(|V|)$ with $O(|E| + |V|)$ and $O(|V|^2)$ with $O(|E| + |V|^2)$.

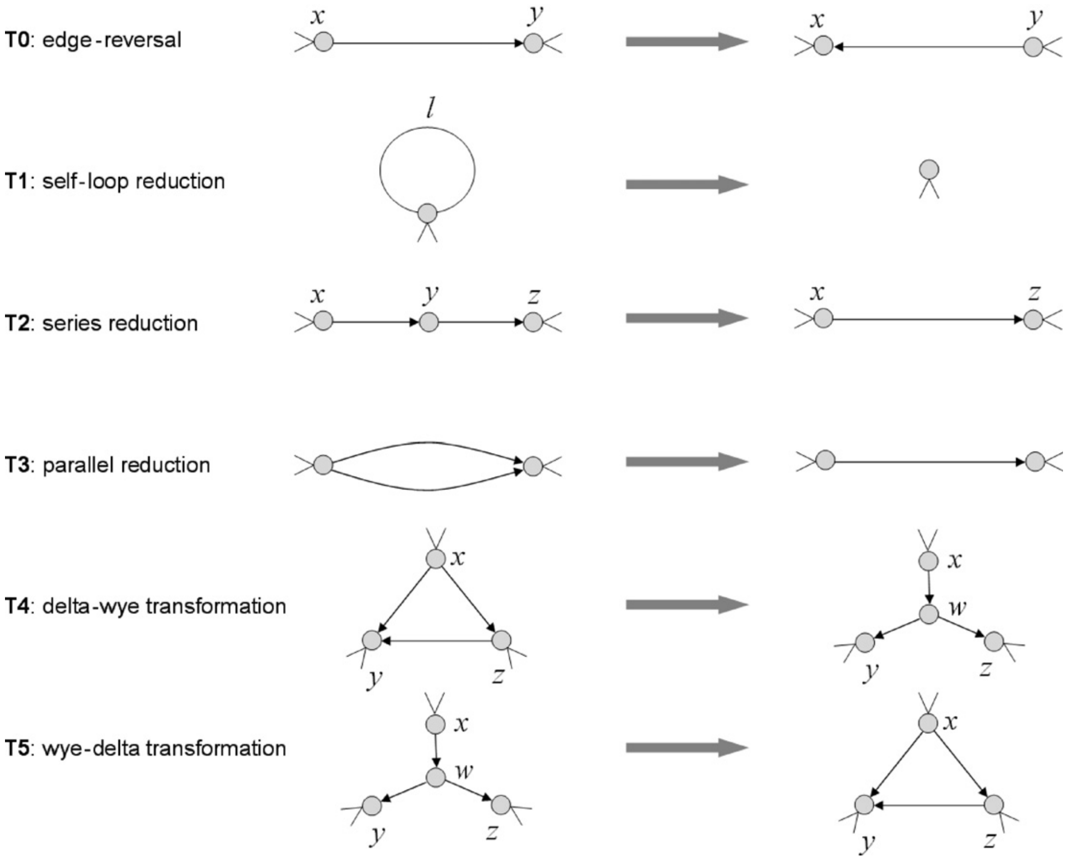


Fig. 2. Transformations T0–T5.

5. The algorithm

In this section we develop an algorithm for the flow estimation problem, where the flow network G is a $\Delta Y\Delta$-reducible graph. The algorithm, ΔY-MPFE, performs at each step a single transformation which combines a graph transformation as presented in the previous section (T0–T5) with the required changes of the measurements and their precisions. The combined transformations are denoted by M0–M5, corresponding to transformations T0–T5, respectively. When a single edge remains, the optimal flow for the remaining edge becomes available. Then, inverse transformations are used for backtracking the sequence of transformations and for computing in reverse order the most probable flow for every edge in the graph. Assuming k transformations reduce a graph G to a single edge, the algorithm finds the most probable flow for every edge in the graph in $O(k)$ steps. The size k is determined by the reduction algorithm used, which yields an $O(|E| + |V|^2)$ algorithm when G is planar (using DWR) and an $O(|E| + |V|)$ algorithm when G is a serial-parallel reducible graph.

Let e_{xy} and τ_{xy} denote the measurement and precision of measurement on an edge xy. The *MPFE transformations* are defined as follows:

M0 (*MPFE edge-reversal*): Reverse the direction of an edge xy to yx. Set

$$e_{yx} = -e_{xy}, \qquad \tau_{yx} = \tau_{xy}.$$

M1 (*MPFE self-loop reduction*): Delete an edge l whose both endpoints are the same node.
M2 (*MPFE series reduction*): Delete a node y of degree two, remove its two incident edges xy and yz, and add a new edge xz. Set

$$e_{xz} = \frac{\tau_{xy}e_{xy} + \tau_{yz}e_{yz}}{\tau_{xy} + \tau_{yz}}, \qquad \tau_{xz} = \tau_{xy} + \tau_{yz}.$$

M3 (*MPFE parallel reduction*): Delete a pair of parallel edges $xy1$, $xy2$, and add a new edge $xy3$. Set

$$e_{xy3} = e_{xy1} + e_{xy2}, \qquad \tau_{xy3} = \frac{\tau_{xy1}\tau_{xy2}}{\tau_{xy1} + \tau_{xy2}}.$$

M4 (*MPFE delta-wye transformation*): Delete the edges of a delta triangle (xy, xz, zy), and add a new node w and new edges xw, wy, and wz. Set

$$e_{xw} = e_{xy} + e_{xz}, \qquad \tau_{xw} = \frac{\tau_{xy}\tau_{xz}}{\tau_{xy} + \tau_{xz} + \tau_{zy}}$$

$$e_{wy} = e_{xy} + e_{zy}, \qquad \tau_{wy} = \frac{\tau_{xy}\tau_{zy}}{\tau_{xy} + \tau_{xz} + \tau_{zy}}$$

$$e_{wz} = e_{xz} - e_{zy}, \qquad \tau_{wz} = \frac{\tau_{xz}\tau_{zy}}{\tau_{xy} + \tau_{xz} + \tau_{zy}}.$$

M5 (*MPFE wye-delta transformation*): Delete a wye node w and its three incident edges xw, wy, wz, and add a delta triangle (xy, xz, zy). Set

$$e_{xy} = e_{wy} - \frac{\tau_{xw}\tau_{wz}(e_{wz} + e_{wy} - e_{xw})}{\tau_{xw}\tau_{wy} + \tau_{xw}\tau_{wz} + \tau_{wy}\tau_{wz}}, \quad \tau_{xy} = \frac{\tau_{xw}\tau_{wy} + \tau_{xw}\tau_{wz} + \tau_{wy}\tau_{wz}}{\tau_{wz}},$$

$$e_{xz} = e_{wz} - \frac{\tau_{xw}\tau_{wy}(e_{wz} + e_{wy} - e_{xw})}{\tau_{xw}\tau_{wy} + \tau_{xw}\tau_{wz} + \tau_{wy}\tau_{wz}}, \quad \tau_{xy} = \frac{\tau_{xw}\tau_{wy} + \tau_{xw}\tau_{wz} + \tau_{wy}\tau_{wz}}{\tau_{wy}},$$

$$e_{zy} = 0, \quad \tau_{zy} = \frac{\tau_{xw}\tau_{wy} + \tau_{xw}\tau_{wz} + \tau_{wy}\tau_{wz}}{\tau_{xw}}.$$

Note that M0–M5 transform one instance (G, e, τ) of MPFE to another. We denote by $M(G, e, \tau)$ the MPFE instance resulting from applying a transformation M to (G, e, τ).

The *inverse MPFE transformations* are used to undo the MPFE transformations by using the most probable flow found for some edges to find the most probable flow for others. We use X^*_{xy} to denote the most probable flow on edge xy and X^*_G to denote the most probable flow for a graph G. Denote by $\mathrm{IM}(X^*_G)$ the most probable flow for a graph obtained from G by applying an inverse MPFE transformation IM. The inverse MPFE transformations are defined as follows:

IM_0 (*inverse MPFE edge-reversal*): Reverse the direction of an edge yx to xy. Set

$$X^*_{xy} = -X^*_{yx}.$$

IM_1 (*inverse MPFE self-loop reduction*): Restore a deleted loop l. Set

$$X^*_l = e_l.$$

IM_2 (*inverse MPFE series reduction*): Delete an edge xz and restore the serial edges xy and yz. Set

$$X^*_{xy} = X^*_{xz},$$

$$X^*_{yz} = X^*_{xz}.$$

IM$_3$ (*inverse MPFE parallel reduction*): Delete an edge $xy3$ and restore the parallel edges $xy1$, $xy2$. Set

$$X^*_{xy1} = \frac{\tau_{xy1} e_{xy1} + \tau_{xy2}(X^*_{xy3} - e_{xy2})}{\tau_{xy1} + \tau_{xy2}},$$

$$X^*_{xy2} = X^*_{xy3} - X^*_{xy1}.$$

IM$_4$ (*inverse MPFE delta-wye transformation*): Delete the wye node w and edges xw, wy, and wz, and restore the delta triangle (xy, xz, zy). Set

Algorithm ΔY-MPFE

Input: An Instance (G, e, τ) of the Gaussian-MPFE problem where G is a $\Delta\text{Y}\Delta$-reducible graph.

Output: The most probable flow for every edge in G, represented as a graph G with optimal flow values on each edge, denoted X^*_G.

Comments:

1. ΔYR denotes a reduction algorithm capable of reducing G to a single edge. The input to ΔYR is a graph and the output is a single transformation from T0-T5 as the next step in the reduction process.
2. $T(G)$ denotes the graph resulting from applying transformation T to G
3. (G, e, τ) denotes an instance of the MPFE problem and $M(G, e, \tau)$ denotes the MPFE instance resulting from applying an MPFE-transformation M to (G, e, τ)

Initialization:

1. For each edge l_i, let e_i be the measurement of the flow on that edge and let τ_i be the precision of this measurement.

Main:

$X^*_G \leftarrow \text{RecurseMPFE}(G, e, \tau)$

Procedure RecurseMPFE(G, e, τ)

If G has a single edge, xy, **Then**
  $X^*_{xy} \leftarrow e_{xy}$
  Return X^*_{xy}
If G has two edges,
  Then $T \leftarrow T2$ (series reduction)
  Else $T \leftarrow \Delta\text{YR}(G)$
$M \leftarrow$ MPFE transformation corresponding to T
IM $\leftarrow$ inverse MPFE transformation of M
$(G', e', \tau') \leftarrow M(G, e, \tau)$
$X^*_{G'} \leftarrow \text{RecurseMPFE}(G', e', \tau')$
$X^*_G \leftarrow \text{IM}(X^*_{G'})$
Return X^*_G

Fig. 3. Algorithm ΔY-MPFE.

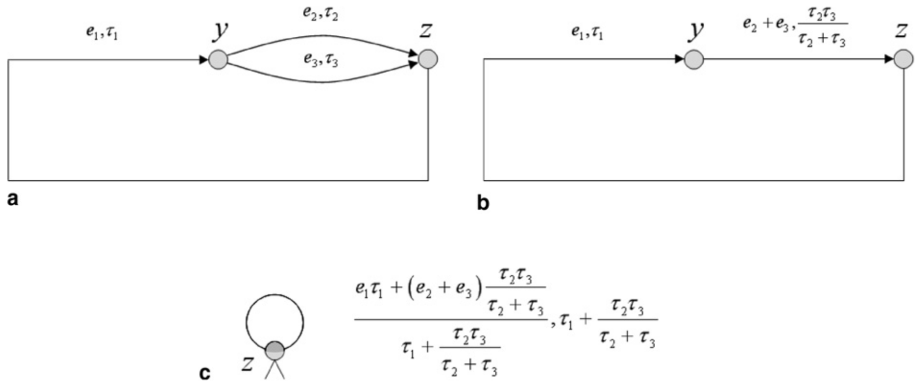


Fig. 4. (a) A flow network with three edges; (b) the network obtained by applying MPFE transformation M3 on (a); (c) the network obtained by applying MPFE transformation M2 on (b). On each edge the flow and precision are indicated.

$$X^*_{xy} = \frac{\tau_{xz}X^*_{xw} + \tau_{zy}X^*_{wy} + \tau_{xy}e_{xy} - \tau_{xz}e_{xz} - \tau_{zy}e_{zy}}{\tau_{xy} + \tau_{xz} + \tau_{zy}},$$
$$X^*_{xz} = \frac{(\tau_{xy} + \tau_{zy})X^*_{xw} - \tau_{zy}X^*_{wy} - (\tau_{xy}e_{xy} - \tau_{xz}e_{xz} - \tau_{zy}e_{zy})}{\tau_{xy} + \tau_{xz} + \tau_{zy}},$$
$$X^*_{zy} = \frac{-\tau_{xz}X^*_{xw} + (\tau_{xy} + \tau_{xz})X^*_{wy} - (\tau_{xy}e_{xy} - \tau_{xz}e_{xz} - \tau_{zy}e_{zy})}{\tau_{xy} + \tau_{xz} + \tau_{zy}}.$$

IM_5 (*inverse MPFE wye-delta transformation*): Delete a delta triangle (xy, xz, zy) and restore the wye node w and its three incident edges xw, wy, wz. Set

$$X^*_{xw} = X^*_{xy} + X^*_{xz},$$
$$X^*_{wy} = X^*_{xy} + X^*_{zy},$$
$$X^*_{wz} = X^*_{xz} - X^*_{zy}.$$

Our algorithm is presented in Fig. 3. It uses a generic ΔY reduction algorithm, denoted ΔYR, which reduces a given $\Delta Y\Delta$-*reducible* graph using transformations T0–T5. Such ΔY reduction algorithms are usually presented for undirected graphs and therefore do not use transformation T0. We can adopt any ΔY reduction algorithm for undirected graphs by using T0 for reversing edges as required. In Fig. 4, an example is presented of applying the forward-phase of ΔY-MPFE on a simple network.

6. Correctness and properties

We start with several algebraic propositions needed for proving correctness of algorithm ΔY-MPFE. It is convenient to denote the normal distribution $N(e_i|X_i, \tau_i)$ with $f_{e_i,\tau_i}(X_i)$.

Proposition 6.1

$$\arg\max_X \prod_{j=1}^{k} f_{e_j,\tau_j}(X) = \arg\max_X f_{E,T}(X),$$

where $T = \sum_{j=1}^{k}\tau_j$ *and* $E = \frac{1}{T}\sum_{j=1}^{k}\tau_j e_j$.

Proof

$$\prod_{j=1}^{k} f_{e_j,\tau_j}(X) = C_1 \prod_{j=1}^{k} \exp\left\{-\frac{1}{2}\tau_j(X-e_j)^2\right\} = C_2 \exp\left\{-\frac{1}{2}\left(\sum_{j=1}^{k}\tau_j\right)\left(X^2 - 2X\frac{\sum_{j=1}^{k}\tau_j e_j}{\sum_{j=1}^{k}\tau_j}\right)\right\}$$
$$= C_3 \exp\left\{-\frac{1}{2}\mathrm{T}(X-E)^2\right\} = C_4 f_{E,T}(X),$$

where C_1 through C_4 are constants. □

An immediate consequence of Proposition 6.1 is that serial edges in an instance of the Gaussian-MPFE problem may be replaced by a single edge with equivalent measurement and precision of measurement, such that performing most probable flow estimation on the resulting network is the same as on the original. This idea, of iteratively simplifying the graph structure along with locally modifying measurements and precisions values, also motivates the following propositions.

Proposition 6.2. *Let* $L(X_p) = \max_{X_l,X_r} f_{\mu_l,\tau_l}(X_l) f_{\mu_r,\tau_r}(X_r)$, *where* $X_l + X_r = X_p$ *and* X_p *is given. Then*

$$L(X_p) = C f_{\mu_l+\mu_r,\frac{\tau_r\tau_l}{\tau_r+\tau_l}}(X_p),$$

where C is a constant.

Proof. $L(X_p)$ can be rewritten as follows:

$$L(X_p) = \max_{X_l} f_{\mu_l,\tau_l}(X_l) f_{\mu_r,\tau_r}(X_p - X_l). \tag{6.5}$$

The value X_l^* which maximizes $L(X_p)$ is given by

$$X_l^* = \arg\min_{X_l} \tau_l(X_l-\mu_l)^2 + \tau_r(X_p - X_l - \mu_r)^2 = \frac{\tau_l\mu_l + \tau_r X_p - \tau_r\mu_r}{\tau_l+\tau_r}, \tag{6.6}$$

which is a global minimum because (6.6) is quadratic. Consequently, the two terms of Eq. (6.5) can be written as follows:

$$f_{\mu_l,\tau_l}(X_l^*) = C_1 \exp\left\{\frac{-\tau_l\tau_r^2}{2(\tau_l+\tau_r)^2}(X_p - (\mu_r+\mu_l))^2\right\} = C_1' f_{\mu_r+\mu_l,\frac{\tau_l\tau_r^2}{(\tau_l+\tau_r)^2}}(X_p)$$

and

$$f_{\mu_r,\tau_r}((X_p - X_l^*)) = C_2' f_{\mu_l+\mu_r,\frac{\tau_r\tau_l^2}{(\tau_l+\tau_r)^2}}(X_p).$$

Using Proposition 6.1 yields the desired claim. □

Proposition 6.3. *Let*

$$L(X_a,X_b) = \max_{\substack{X_1,X_2,X_3 \\ \text{s.t.: } X_1+X_2=X_a \\ X_1+X_3=X_b}} f_{e_1,\tau_1}(X_1) f_{e_2,\tau_2}(X_2) f_{e_3,\tau_3}(X_3),$$

where X_a *and* X_b *are given. Then*

$$L(X_a,X_b) = f_{e_1+e_2,\frac{\tau_1\tau_2}{\tau_1+\tau_2+\tau_3}}(X_a) f_{e_1+e_3,\frac{\tau_1\tau_3}{\tau_1+\tau_2+\tau_3}}(X_b) f_{e_2-e_3,\frac{\tau_2\tau_3}{\tau_1+\tau_2+\tau_3}}(X_a - X_b).$$

Proof. The values X_1^*, X_2^*, X_3^* which maximize $L(X_a, X_b)$ are given by

$$(X_1^*, X_2^*, X_3^*) = \mathop{\arg\min}_{\substack{X_1,X_2,X_3 \\ \text{s.t.: } X_1+X_2=X_a \\ X_1+X_3=X_b}} \tau_1(X_1 - e_1)^2 + \tau_2(X_2 - e_2)^2 + \tau_3(X_3 - e_3)^2. \tag{6.7}$$

The unique solution of this quadratic problem, using Lagrange multipliers, is

$$X_1^* = \frac{\tau_2 X_a + \tau_3 X_b + \tau_1 e_1 - \tau_2 e_2 - \tau_3 e_3}{\tau_1 + \tau_2 + \tau_3},$$
$$X_2^* = \frac{(\tau_1 + \tau_3)X_a - \tau_3 X_b - (\tau_1 e_1 - \tau_2 e_2 - \tau_3 e_3)}{\tau_1 + \tau_2 + \tau_3},$$
$$X_3^* = \frac{-\tau_2 X_a + (\tau_1 + \tau_2)X_b - (\tau_1 e_1 - \tau_2 e_2 - \tau_3 e_3)}{\tau_1 + \tau_2 + \tau_3}.$$

Substituting these values in $L(X_a, X_b)$ and rearranging terms yields the required result. □

Proposition 6.4. *Let*

$$L(X_a, X_b) = \max_{\substack{X_1,X_2,X_3 \\ \text{s.t.: } X_1+X_2=X_a \\ X_1+X_3=X_b}} f_{-c_1+e_b-\frac{\tau_a\tau_c(e_c+e_b-e_a)}{T},\frac{T}{\tau_c}}(X_1) f_{c_1+e_c-\frac{\tau_a\tau_b(e_c+e_b-e_a)}{T},\frac{T}{\tau_b}}(X_2) f_{c_1,\frac{T}{\tau_a}}(X_3),$$

where $T = \tau_a\tau_b + \tau_a\tau_c + \tau_b\tau_c$, X_a *and* X_b *are given, and where* c_1 *is a constant. Then,*

$$L(X_a, X_b) = c_2 f_{e_a,\tau_a}(X_a) f_{e_b,\tau_b}(X_b) f_{e_c,\tau_c}(X_a - X_b),$$

where c_2 *is a constant.*

Proof. By applying Proposition 6.3 and rearranging terms the proof follows. □

Propositions 6.3 and 6.4 are key contributions that facilitate our algorithm. Together with Propositions 6.1 and 6.2 they imply that the most probable flow for a given MPFE instance (G, e, τ) is derivable from the most probable flow for a simpler instance obtained from (G, e, τ) by one of the transformations M0–M5. This claim is proven in the following lemma.

Lemma 6.5. *Let* (G, e, τ) *be an instance of the* Gaussian*-MPFE problem where* G *has at least* 2 *edges, and let* M_i *and* IM_i *be the MPFE transformations and their inverses, where* $i = 0, \ldots, 5$. *Then, the most probable flow, denoted by* X_G^*, *can be found as follows*:

1. $(G', e', \tau) \leftarrow M_i(G, e, \tau)$.
2. $X_{G'}^* \leftarrow$ MPFE *for the instance* (G', e', τ').
3. $X_G^* \leftarrow \mathrm{IM}_i(X_{G'}^*)$.

Proof. The claim is proven for each of the six transformations.

*Transformation M*0. Eq. (2.3) has the form

$$X_G^* = \mathop{\arg\max}_{\text{s.t. } AX=0} f_{e_{xy},\tau_{xy}}(X_{xy}) \prod_{i \in E\setminus\{xy\}} f_{e_i,\tau_i}(X_i) = \mathop{\arg\max}_{\text{s.t. } AX=0} f_{-e_{xy},\tau_{xy}}(-X_{xy}) \prod_{i \in E\setminus\{xy\}} f_{e_i,\tau_i}(X_i),$$

where $E\setminus\{xy\}$ are all the edges of G excluding edge xy.

*Transformation M*1. The flow in a self-loop l is not constrained by the flow in other edges of the flow network. Hence removing l from (G, e, τ) (step 1), solving (G', e', τ') (step 2), and then assigning $X_l^* \leftarrow e_l$ which maximizes f_{e_l,τ_l} yields X_G^*.

*Transformation M*2. Rewrite Eq. (2.3) as

$$X_G^* = \operatorname*{arg\,max}_{\text{s.t. } AX=0} f_{e_{xy},\tau_{xy}}(X_{xy}) f_{e_{yz},\tau_{yz}}(X_{yz}) \prod_{i \in E \setminus \{xy,yz\}} f_{e_i,\tau_i}(X_i). \tag{6.8}$$

For serial edges xy and yz, we have $X_{xy} = X_{yz}$. Using Proposition 6.1 on Eq. (6.8) yields

$$X_G^* = \operatorname*{arg\,max}_{\text{s.t. } AX=0} f_{\frac{\tau_{xy}e_{xy}+\tau_{yz}e_{yz}}{\tau_{xy}+\tau_{yz}},\tau_{xy}+\tau_{yz}}(X_{xy}) \prod_{i \in E \setminus \{xy,yz\}} f_{e_i,\tau_i}(X_i). \tag{6.9}$$

Replacing X_{xy} with X_{xz}, which are equal, in Eq. (6.9) yields the solution for the MPFE instance (G', e', τ) where the serial edge xy and yz are replaced with the edge xz.

*Transformation M*3. Rewrite Eq. (2.3) as

$$X_G^* = \operatorname*{arg\,max}_{\text{s.t. } AX=0} f_{e_{xy1},\tau_{xy1}}(X_{xy1}) f_{e_{xy2},\tau_{xy2}}(X_{xy2}) \prod_{i \in E \setminus \{xy1,xy2\}} f_{e_i,\tau_i}(X_i). \tag{6.10}$$

For parallel edges $xy1$ and $xy2$, every constraint in $AX = 0$ which includes either of the variables X_{xy1} or X_{xy2} includes their sum. We replace their sum each time it appears by a variable X_{xy3}, and add the constraint $X_{xy1} + X_{xy2} = X_{xy3}$. Since this is the only constraint where either X_{xy1} or X_{xy2} appears, Eq. (6.10) is rewritten as follows:

$$X_{G'}^* = \operatorname*{arg\,max}_{\text{s.t. } A'X'=0} \prod_{i \in E \setminus \{xy1,xy2\}} f_{e_i,\tau_i}(X_i) \max_{\text{s.t. } X_{xy1}+X_{xy2}=X_{xy3}} f_{e_{xy1},\tau_{xy1}}(X_{xy1}) f_{e_{xy2},\tau_{xy2}}(X_{xy2}),$$

where $A'X' = 0$ denotes the modified set of constraints. Using Proposition 6.2 we obtain

$$X_{G'}^* = \operatorname*{arg\,max}_{\text{s.t. } A'X'=0} f_{e_{xy1}+e_{xy2},\frac{\tau_{xy1}\tau_{xy2}}{\tau_{xy1}+\tau_{xy2}}}(X_{xy3}) \prod_{i \in E \setminus \{xy1,xy2\}} f_{e_i,\tau_i}(X_i). \tag{6.11}$$

Eq. (6.11) is the MPFE equation for the MPFE instance $(G', e', \tau') = M3(G, e, \tau)$ (step 1). Let $X_{G'}^* = \text{MPFE}(G', e', \tau')$ be the optimal flow for the edges of G' (step 2). One of these edges is $xy3$ whose flow is the sum of flows on $xy1$ and on $xy2$. Eq. (6.6) yields the optimal flow on $xy1$ for every flow on $xy3$, namely,

$$X_{xy1}^* = \frac{\tau_{xy1}e_{xy1} + \tau_{xy2}(X_{xy3}^* - e_{xy2})}{\tau_{xy1} + \tau_{xy2}}.$$

The flow X_{xy2}^* is determined by $X_{xy1} + X_{xy2} = X_{xy3}$. These flows are those computed by $\text{IM}_3(G')$ (step 3).

*Transformation M*4. The proof is similar to the previous case. Proposition 6.3 is used similarly to the way Proposition 6.2 is used for transformation M3.

*Transformation M*5. The proof is similar to that of transformation M3, using Eq. (6.11) first, then Eq. (6.10) and finally invoking Proposition 6.4. □

The next theorem proves the correctness of algorithm ΔY-MPFE and analyzes its complexity.

Theorem 6.6. *Let (G, e, τ) be an instance of the* GAUSSIAN-*MPFE problem where G is a* 2-*connected $\Delta Y\Delta$-reducible graph. Let ΔYR denote a ΔY reduction algorithm which reduces G to a single edge using k* T0–T5 *transformations. Then, ΔY-*MPFE *computes the most probable flow for every edge in G in time complexity* O(*k*).

Proof. If G has a single edge, then the optimal value for that edge is the measurement on that edge. When G has two edges or more, Lemma 6.5 applies and the proof follows inductively. The ΔY-MPFE algorithm requires k MPFE-transformations and k inverse MPFE-transformations each of O(1) time complexity yielding overall O(k) time complexity. □

Theorem 6.7. *Let* (G, e, τ) *be a* GAUSSIAN*-MPFE problem. Then,* ΔY-MPFE *computes the most probable flow for every edge in G in time complexity* $O(|E| + |V|^2)$ *if G is a* 2*-connected planar graph and in* $O(|E| + |V|)$ *if G is a* 2*-connected serial-parallel graph or a tree.*

Proof. The proof follows from Theorem 6.6 and from the complexity of known reduction algorithms for different topologies. For a 2-connected planar graph, algorithm DWR (Feo and Provan, 1993; Feo and Provan, 1996) has time complexity $O(|E| + |V|^2)$, where serial-parallel graphs may be reduced to a single edge in $O(|E| + |V|)$ (Valdes et al., 1982). The problem of solving MPFE on trees is reduced to solving it on serial-parallel graphs by merging all sources and sinks to a single node. □

Corollary 6.8. *A* MINIMUM COST FLOW *problem for which the cost function is* $\tau_e (X_e - \mu_e)^2$ *for edge e, can be solved in* $O(|E| + |V|^2)$ *for* 2*-connected planar graphs and in* $O(|E| + |V|)$ *for* 2*-connected serial-parallel graphs.*

Proof. The claim follows from Theorems 6.6 and 6.7, and from the fact that such instances of the MINIMUM COST FLOW problem are in fact instances of the MPFE problem. □

Corollary 6.8 is an extension of Theorem 3 in Feo and Provan (1993) in which $\mu_e = 0$ for all edges.

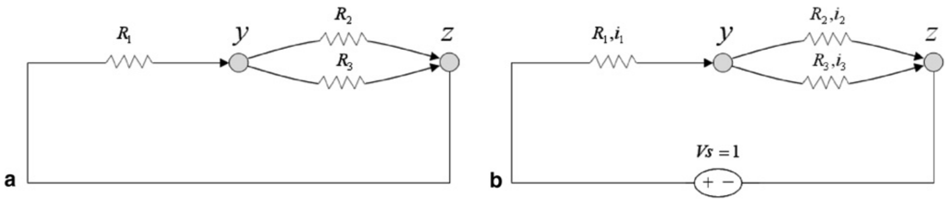
7. MPFE precision

The *precision* of the flow estimation on the *i*th edge is the reciprocal of the variance for the flow estimation for that edge. In Section 3 we described the LMVU method and its application to flow estimation. This method can be used to compute the covariance matrix $C_{\hat{\theta}}$, where the diagonal values represent the flow estimation variances. The precisions of the flow estimations are the diagonal elements of the inverse matrix $C_{\hat{\theta}}^{-1}$.

In this section we examine an analogy between Gaussian-MPFE and electrical networks, relating precisions of flow estimators and equivalent resistances. In an electrical network with no sources, the *equivalent resistance* R^{eq} *in series to resistor r* is defined as the reciprocal of the current through r resulting from a unit-voltage-source added in series to it. A well-known result is that the equivalent resistance of resistors connected in series is the sum of their resistances. Proposition 6.1 shows that the precision of a flow estimator for edges connected in series is the sum of their measurement's precisions. A similar analogy holds for parallel edges (Proposition 6.2). Figs. 4 and 5 develop this example further.

Let (G, e, τ) be a GAUSSIAN-MPFE problem. A (G, e, τ)-*equivalent electrical network* is a network built from a flow network G by replacing every edge l_i with a resistor r_i with resistance $R_i = \tau_i$. The rest of this section shows that for all graph topologies, the precision of the (G, e, τ)-MPFE estimator for edge l_i is identical to the equivalent resistance in series to l_i in the (G, e, τ)-equivalent electrical network. This result connects the Most Probable Flow Estimation problem with several research areas related to electrical networks and can be used for leveraging results from one domain to another.

Proposition 7.1. *Let* (G, e, τ) *be a* GAUSSIAN*-MPFE problem constrained by* $AX = 0$, *where A is a constraint matrix which represents the flow conservation constraints on internal nodes. Let C be an* $|e|$ *by* $|e|$ *diagonal matrix with* $c_{ii} = \tau_i$*. Then, the most probable flow is given by*



c
$$\begin{cases} i_1 = i_2 + i_3 \\ i_1R_1 + i_2R_2 = 1 \\ i_2R_2 = i_3R_3 \end{cases}$$

d
$$i_1 = \frac{R_2 + R_3}{R_1R_2 + R_1R_3 + R_2R_3}$$
$$R_1^{eq} = \frac{1}{i_1} = R_1 + \frac{R_2R_3}{R_2 + R_3}$$

Fig. 5. (a) An electrical network with a topology identical to that of the flow network in (a); in (b) a unit-voltage-source is added in series to resistor r_1; (c) presents the electrical network equations; (d) presents the solution of these equations for resistor r_1.

$$X^* = e - CA^{\mathrm{T}}(ACA^{\mathrm{T}})^{-1}Ae \tag{7.12}$$

and the covariance matrix is

$$C_{X^*} = (C - CA^{\mathrm{T}}(ACA^{\mathrm{T}})^{-1}AC). \tag{7.13}$$

Proof. Taking the logarithm of Eq. (2.3) yields a sum of quadratic terms, constrained by a set of linear equations. The Lagrangian is

$$L = (X - e)^{\mathrm{T}}C^{-1}(X - e) + \lambda^{\mathrm{T}}AX,$$

where λ is a vector of Lagrange multipliers. Solving the Lagrangian yields Eq. (7.12). Noticing that in the expression for the estimator X^* only e is non-deterministic, the variance of X^* may be written as

$$C_{X^*} = (I - CA^{\mathrm{T}}(ACA^{\mathrm{T}})^{-1}A)C(I - CA^{\mathrm{T}}(ACA^{\mathrm{T}})^{-1}A)^{\mathrm{T}}.$$

By rearranging terms, Eq. (7.13) follows. □

Theorem 7.2. *Let (G, e, τ) be a* Gaussian*-MPFE problem. Let τ_i^* be the precision of the flow estimation on edge l_i. Let R_i^{eq} be the equivalent resistance in series to resistor r_i in the (G, e, τ)-equivalent electrical network. Then, for every edge i, $\tau_i^* = R_i^{\mathrm{eq}}$.*

Proof. Standard methods (Desoer and Kuh, 1969, pp. 424–425) provide a formula for the currents I on each edge of an electrical network as a function of the resistances and voltage-sources in the network. The standard formula is

$$I = (GA^{\mathrm{T}}(AGA^{\mathrm{T}})^{-1}AG - G)V, \tag{7.14}$$

where V is the voltage-sources vector, such that $-v_i$ is the voltage source in series to resistor r_i, and where G is a diagonal matrix, such that g_{ii} equals $1/R_i$, and where A is a constraint matrix which represents Kirchhoff's law on internal nodes, namely $AI = 0$. Since G is an electrical network with no sources, in order to determine R_i^{eq} for a resistor r_i, we add a unit-voltage-source in series to resistor r_i and compute the resulting current on r_i. Hence, V is a vector such that $v_j = 0$ for $j \neq i$ and $v_i = -1$. It follows from Eq. (7.14) that $R_i^{\mathrm{eq}} = 1/d_{ii}$, where D is given by

$$D = G - GA^{\mathrm{T}}(AGA^{\mathrm{T}})^{-1}AG. \tag{7.15}$$

Eq. (7.15) is identical to Eq. (7.13), which completes the proof. □

8. Discussion

In this paper we have developed an algorithm for solving Gaussian-MPFE problems. The algorithm's time complexity is $O(|E| + |V|^2)$ when the underlying undirected graph of G is a 2-connected planar graph, and $O(|E| + |V|)$ when it is a 2-connected serial-parallel graph or a tree. This result may be used to solve, in the same time complexity, any Minimum Cost Flow problem for which the cost function is $\tau_e(X_e - \mu_e)^2$ for edge e. In addition, we have shown an analogy between the precision of the estimated flow on each edge and the equivalent resistance measured in series to this edge in an equivalent electrical network.

Our algorithm uses a generic ΔY-reduction algorithm. Consequently, it will benefit from any improvements in reduction schemes for $\Delta Y \Delta$ reducible graphs, and from new classes of $\Delta Y \Delta$ reducible graphs; in particular—the understanding of which non-planar topologies may exploit a ΔY-reduction procedure. Further reductions beyond M0–M5 can extend the algorithm to additional graphs. Theorem 7.2 can be used for leveraging results from the theory of electrical networks and related areas (e.g., Lyons and Peres, to be published). For example, given an infinitely growing family of MPFE instances, to determine if and when the variance of the most probable flow estimation tends to zero.

Acknowledgement

The authors thank Gady Zohar for useful discussions.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms and Applications. Prentice-Hall, New Jersey.

Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J., 1999. Probabilistic Networks and Expert Systems. Springer-Verlag, New York.

Desoer, C.A., Kuh, E.S., 1969. Basic Circuit Theory. McGraw-Hill, NY.

Epifanov, G.V., 1966. Reduction of a plane graph to an edge by a star-triangle transformations. Soviet Mathematics Doklady 166, 13–17.

Feo, T.A., Provan, J.S., 1993. Delta-wye transformations and the efficient reduction of two-terminal planar graphs. Operational Research 41 (3), 572–582.

Feo, T.A., Provan, J.S., 1996. The delta-wye approximation procedure for two terminal reliability. Operational Research 44 (5), 745–757.

Jensen, F.V., 1996. An Introduction to Bayesian Networks. Springer-Verlag, New York.

Kay, S.M., 1993. Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice-Hall, New Jersey.

Lauritzen, S.L., 1992. Propagation of probabilities, means and variances in mixed graphical association models. Journal of the American Statistical Association 87, 1098–1108.

Lauritzen, S.L., Jensen, F., 2001. Stable local computation with conditional Gaussian distributions. Statistics and Computing 11, 191–203.

Lyons, R., Peres, Y., to be published. Probabilities on Graphs and Networks. Cambridge University Press, Cambridge.

Shachter, R.D., Kenley, C.R., 1989. Gaussian influence diagrams. Management Science 35 (5), 527–550.

Valdes, J., Tarjan, R.E., Lawler, E.L., 1982. The recognition of serial-parallel digraphs. SIAM Journal of Computation 11 (2), 298–313.
Warkentyne, H.M.K., 1988. ΔY reduction. M.Sc, Math Thesis, Univ. Waterloo.
Zohar, R., Geiger, D., 2003. A novel framework for tracking groups of objects. Bayesian Statistics 7, 419–440.