Efficient Sensor Placement in Flow Networks and Sensor Networks

Patrick Traxler and Mariya Zhariy
Software Competence Center Hagenberg, Hagenberg, Austria
e-mail: {patrick.traxler, mariya.zhariy}@scch.at

Abstract
We study the problem of sensor placement for maximum structural fault detection and isolation in systems with a graphical structure. In particular, we consider flow networks and sensor networks. We are interested in placing as few sensors as possible. We get efficient approximation algorithms and exact algorithms for computing a smallest sensor set for maximum structural fault detectability and isolation.

1 Introduction
In the framework of structural fault diagnosis of Frisk and Krysander [1; 2], systems of mathematical equations describe technical systems. The systems of equations are treated in a purely structural or combinatorial way. In particular, we mainly study the bipartite variables-in-equations graph associated with a system of equations [1]. We use combinatorial concepts and techniques such as the Dulmage-Mendelsohn decomposition [3] and algorithms for computing maximum matchings to solve fault detection, fault isolation, and sensor placement problems [1].

The benefit of this approach is that we can apply it early in the design phase of a system. No sensor data is necessary. In this work we consider systems of equations which have themselves a graphical structure. In particular, we consider flow networks and sensor networks.

Flow networks and models thereof arise in areas such as electrical engineering, hydraulics, and transportation. They essentially contain two building components: Energy nodes (voltage, pressure) which are connected by flow edges (current, volume flow). See Fig. 1 for an example. The mathematical description of flow networks follows general principles. For example, the amount of flow into a node equals the amount of flow out of it, except for sources or sinks. This conservation constraint or flow preservation was formalized as Kirchhoff’s current law in the context of electrical circuits.

Sensor networks and models thereof become increasingly important with the advent of the Internet of Things in which systems, machines, and devices are connected via the Internet. One particular type of sensor network that we consider here has the property that all sensors measure the same quantity or similar quantities, e.g. inside or outside temperature, solar radiation, consumed or produced energy. A common way to model sensor networks is to consider the geographic location of sensors. Nearby sensors have often similar values. This can be used for fault detection. See e.g. [4].

We study the problem of placing as few sensors as possible. That is, the results of our algorithms are smallest or almost smallest sensor sets which allow maximum fault isolation. This is in contrast to enumerating all reasonable (i.e. minimal) sensor sets which achieve maximum fault detection or isolation [1]. We also aim at providing running time and approximation guarantees.

We study special cases which are motivated by different models of flow and sensor networks. Our main result reads as follows: The Minimum Sensor Placement (MSP) problem for instances with a symmetric variables-in-equations graph can be solved efficiently and approximatively. As far as flow and sensor networks are concerned, we can model the MSP problem for variable flow networks and sensor networks in terms of symmetric variables-in-equations graphs and can thus solve the problem efficiently. We formulate our special cases in terms of variables-in-equations graphs.

We provide the necessary definitions in Sec. 2, models of flow and sensor networks in Sec. 3, and algorithms and their analysis in Sec. 4.

1.1 Motivating Applications
Flow networks arise naturally in the study of technical systems. Here, we present a snippet of a real-world hydraulic system and show how to apply our results to it. In contrast to many flow networks that describe real-world systems, sensor networks may be considerably larger. In our applications, the graphical structure of our flow network has a few dozen of vertices and edges. The snippet of the hydraulic system we present, Fig. 1, has just 5 vertices. The sensor networks can have hundreds of vertices and edges in our application. Sensors in our application measure the produced energy, module temperature, plane-of-array irradiance of photovoltaic systems. But not every photovoltaic system delivers data all the time. In other words, the set of possible sensor locations changes over time. Once per day we check for missing data and place the sensors appropriately.

2 Definitions: Detection, Isolation, Sensor Placement
The purpose of this section is to introduce the Minimum Sensor Placement (MSP) problem.

An example of a set of equations is Eq. 1 and Eq. 2. It describes a part of a hydraulic system. In Fig. 2 we see
its variables-in-equations graph. In general, the variables-in-equations graph is a bipartite graph $B = (U, V, E)$ with $U \cap V = \emptyset$ and $U$ are the equations and $V$ the variables in $M$. Mathematically, the system of equations or model $M$ is a set and thus $U = M$. We draw an edge between $e \in U$ and $v \in V$ iff variable $v$ occurs in equation $e$. Note that $F = \{f_1, \ldots, f_6\}$ are the fault variables in the example. Fault variables do not belong to $V$. Moreover, every fault $f$ occurs in at most one equation which we denote by $e_f$, and at most one fault is associated with an equation. In the example, we can set $P = \{v_1, \ldots, v_6\}$. In general, $P \subseteq V$ is the set of possible sensor places or possible sensor locations and $F$ is the set of faults for which $e_f \in U$ for every $f \in F$.

### Dulmage-Mendelsohn Decomposition

For every bipartite graph $B$ there exists an integer $n$ such that the vertices of $B$ and thus $M$ can be partitioned into an under-determined part $M_0$, a just-determined part $M^0$, and an over-determined part $M^+$. The concrete definitions of these parts can be found in [5]. We provide them in Sec. 4. The partition is called the Dulmage-Mendelsohn (DM) decomposition$^1$ and was introduced in [3].

An example of a DM-decomposition with one just-determined part, no under-determined, and no over-determined part is depicted in Fig. 3.

**Definition 1** ([11]). A fault $f$ is structurally detectable in $M$ iff $e_f \in M^+$.

This definition says that we need more equations than variables to detect faults. So, to detect faults for the example in Fig. 3 we need to add one more equation to $M$. The idea of sensor placement is to add equations of the form $v = c$ for some variable $v \in P \subseteq V$ and a value $c$. For a set $S \subseteq P$, define $M_S$ as the set of these equations. The goal is to find a small set $S$ such that $e_f \in (M \cup M_S)^+$.

**Definition 2** ([11]). A fault $f_i$ is structurally isolable from fault $f_j$ in $M$ iff $e_{f_i} \in (M \setminus \{e_{f_j}\})^+$.

This notion ensures that we can also identify a fault, i.e. we can locate the fault. Here, we make the assumption that only a single fault happens, i.e. only one fault $f \in F$ can have a value unequal to 0. See [6] for further discussion on this topic and in particular on residual generation. A second assumption is that sensors work correctly. See [6] for a solution to handle faulty sensors.

We are now able to define MSP. As above we want to find a small set $S$ such that $e_{f_i} \in (M \setminus \{e_{f_j}\})^+$. Additionally we want to maximize such pairs $(i, j)$. This motivates the following definitions.

**Definition 3.** Let $I = (M, F, P)$, $f \in F$, and $S \subseteq P$. Define

$$\tau_I(f, S) := \{ f' \in F : e_{f'} \in ((M \setminus \{e_f\}) \cup M_S)^+ \}$$

and

$$\tau_I(f) := \max_{S \subseteq P} \tau_I(f, S).$$

We call a set $S \subseteq P$ for which $\tau_I(f, S) = \tau_I(f)$ for all $f \in F$ a correct sensor placement.

$^1$In this work we will mainly cite [5] as it contains a presentation of the DM decomposition which is suitable for our needs. We refer the interested reader to [5] for references of the original work of Dulmage, Mendelsohn and others.

In words, a correct sensor placement is such that it achieves maximum isolation for every fault. It holds that a sensor set $S \subseteq P$ is a correct sensor placement iff $S$ maximizes $\sum_{f \in F} \tau_I(f, S)$. In words, $S$ is a correct sensor placement iff $S$ maximizes the total number of fault pairs $(f', f)$ where $f'$ is structurally isolable from $f$.

### Minimum Sensor Placement (MSP)

**Input:** Equations $M$ with faults $F$ and sensor locations $P$.

**Output:** A smallest set $S \subseteq P$ which is a correct sensor placement.

### 3 Models for Flow and Sensor Networks

#### 3.1 Flow Networks

Our models of flow networks, see for example Fig. 1, contain energy variables (e.g. pressure, voltage) and flow variables (e.g. volume flow, current.) Energy variables are vertex labels and flow variables are edge labels of a graph $G$. See Eq. 1 and 2 for an example. One type of equations express flow preservation: The flow into a vertex equals the flow out of the vertex. This is known as Kirchhoff’s current law in the context of electrical circuits. An equivalent for hydraulic systems are called pressure built-up equations.

The following example is the snippet of a hydraulic system. It is graphically depicted in Fig. 1. The energy variables $p_i$ represent pressure. The flow variables $c_i$ represent the volume flow through some hydraulic component, e.g. pipe, valve, etc.

![Flow network $G_{FN}$](image)

The direction of each edge in the flow network graph $G_{FN}$ determines the sign of the corresponding flow summand. The mathematical description of the flow network in Fig. 1 is given by

$$
\begin{align*}
e_1 : & \quad \dot{p}_1 = c_1 + c_4 - c_6 \\
e_2 : & \quad \dot{p}_2 = c_3 \\
e_3 : & \quad \dot{p}_3 = -c_1 + c_2 \quad (1) \\
e_4 : & \quad \dot{p}_4 = -c_4 + c_5 \\
e_5 : & \quad \dot{p}_5 = -c_2 - c_3 - c_5 + c_6
\end{align*}
$$

We assume that the possible sensors are located at the energy variables $p_i$ and that only the flow variables $c_j$ are affected by faults. The variables $f_1, \ldots, f_6$ are faults.

$$
\begin{align*}
e_6 : & \quad c_1 = c_1'(p_1, p_3) + f_1 \\
e_7 : & \quad c_2 = c_2'(p_3, p_5) + f_2 \\
e_8 : & \quad c_3 = c_3'(p_2, p_5) + f_3 \\
e_9 : & \quad c_4 = c_4'(p_1, p_4) + f_4 \quad (2) \\
e_{10} : & \quad c_5 = c_5'(p_4, p_5) + f_5 \\
e_{11} : & \quad c_6 = c_6'(p_1, p_3) + f_6
\end{align*}
$$

The flow variables $c_i$ depend on the energy variables $p_i$ which are adjacent to $c_i$ in the network flow graph $G$. The
functions $c'(\cdot)$ may be known. But we do not need to know them for sensor placement. It makes however a difference if the functions $c'$ are independent of the energy variables.

The variables-in-equations graph $B = (U, V, E)$ for our example is depicted in Fig. 2: $U = \{e_1, \ldots, e_{11}\}$, $V = \{p_1, \ldots, p_5, c_1, \ldots, c_6\}$ and we draw an edge between $e \in U$ and $v \in V$ if $v$ occurs in $e$.

We make an interesting case distinction in Fig. 2. On the left, we see the variables-in-equations graph in case of constant flow, i.e. the functions $c'$ are independent of the energy variables. On the right, we see the variables-in-equations graph for variable flow. We call such a bipartite graph symmetric which has the same structure if we exchange the left and right side. We provide the exact definition in Sec. 4.3.

We can easily generalize the above sample derivation of the set of equations from our graph in Fig. 1 to arbitrary graphs $G$. We call these systems constant flow networks if the functions $c'(\cdot)$ are constants and variable flow networks in the other case.

**Figure 2:** Variables-in-equations graphs for the flow network examples with constant flow (left) and variable flow (right), resp.

### 3.2 Sensor Networks

In sensor networks, we model variables as the vertex labels of an undirected graph $G$. Equations are of the form $x_i = f(X_i)$ where $X_i$ is the set of adjacent variables to $x_i$ in $G$. See Fig. 3 for an example.

**Figure 3:** Sensor network $Gsn$ (left) and its variables-in-equations graph $B = B(M_{sn})$ (right).

We can derive the system of equations directly from the graph $G$ in Fig. 3. We have 5 equations in our example. A sample equation is $e_1$ with $X_1 = \{x_2, x_3\}$; $x_1 = g_1(x_2, x_3)$ if $f_1 \not\in F$ and $x_1 = g_1(x_2, x_3) + f_1$ if $f_1 \in F$. We may read the latter equation in two ways: The derivation of the value of $x_1$ from its neighboring variables in $G$ is possibly faulty. Or, the value of $x_1$ is possibly faulty.

In applications, the measured quantities are for example temperature, solar radiation, consumed or produced energy. The functions $g_i$ are often linear in the variables from $X_i$.

In the above way we can derive for a given sensor network $G_{sn}$ with variables $x_1, \ldots, x_n$ and faults $F$ a model of sensor networks suitable for fault detection and isolation. We call such a system a sensor network.

### 4 Algorithms: Description and Analysis

#### 4.1 Sensor Placement as Graph Reachability

We show how sensor placement reduces to graph reachability. This is similar to Lemma 1 in [1]. We also provide the formal definition of the DM decomposition here. The presentation follows [5].

We recall that $X \subseteq E$ is a (perfect) matching in a bipartite graph $B = (U, V, E)$ if every vertex in $B$ occurs in (exactly one) at most one edge in $M$. Let $X$ be a maximum matching in $B$. An alternating path w.r.t. $X$ is a path in $B$ such that no two neighboring edges in it are both from $X$ or both from $E \setminus X$. A vertex is called matched by $X$ if it occurs in $X$ and unmatched otherwise. We define VR (HR) as the set of all vertices from $U$ which are reachable from some unmatched vertex in $U$ (V) via some alternating path w.r.t. $X$. We define VC (HC) as the set of all vertices from $V$ which are reachable from some unmatched vertex in $U$ (V) via some alternating path w.r.t. $X$. Set $SR := U \setminus (HR \cup VR)$ and $SC := V \setminus (HC \cup VC)$.

Let $M$ be a set of equations and $B = B(M) = (U, V, E)$ the corresponding variables-in-equations graph. Since $|HR| < |HC|$, $|SR| = |SC|$, and $|VR| > |VC|$ (see [5]), we call the respective subgraphs of $B$ which are induced by $HR \cup HC$, $SR \cup SC$, and $VR \cup VC$ as the under-determined part $M_0$, the just-determined part $M^0$, and the over-determined part $M^+$. This partition of $B$ and thus $M$ is independent of the choice of $X$, see Theorem 2.1 in [5].

**Proposition 1.** Any two maximum matchings $X_1$ and $X_2$ in a bipartite graph $B$ yield the same sets $HR, HC, VR, VC, SR, SC$.

This completes the formal definition of structural detectability, Def. 1, and structural isolability, Def. 2.

Next, we define for $B$ with a perfect matching $X$ the graph $G(B, X)$. It is $B$ where edges not in $X$ are directed from equation vertices $U$ to variable vertices $V$ and edges in $X$ are shrunken into a single vertex. Thus, $G(B, X)$ is a directed graph and we can naturally identify the vertices in $G(B, X)$ from $U$ and $V$.

Let us consider some example. We derive the variables-in-equations graph $B_{sn} = B(M_{sn})$ in Fig. 3 from the sensor network $G_{sn}$. A maximum and actually a perfect matching $X$ in $B$ is given by the edges $\{e_i, x_i\}$, $i \in \{1, \ldots, 5\}$. We make two observations.

First, $G(B_{sn}, X)$ is equivalent to $G_{sn}$ in our example. The difference is that $G(B_{sn}, X)$ does have edge orientations. Second, there is no under-determined part $M_0$ and no over-determined part $M^+$ in $B_{sn}$. In particular, $SR = SC = \emptyset$ and thus no fault is detectable.

Adding sensor measurement equations $M_S$, $S \subseteq P \subseteq V$, to $M_{sn}$ is equivalent to adding new vertices to $U$ in $B_{sn}$. In our example, the measurement of a single sensor variable suffices to yield maximum fault detectability. We just note that the perfect matching in our example is a maximum matching in $M \cup M_S$.

This result holds in general if we have a set of equations and $B = B(M)$ has a perfect matching. The situation is however more complicated if we first remove an equation...
from $M$ and then add equations for a sensor measurement – the situation that arises in case of fault isolation.

We will need the concept of an augmenting path. We recall that an augmenting path $p$ of a matching $X$ is an alternating path w.r.t. $X$ that begins and ends with vertices which are unmatched by $X$ (see e.g. [51]). We observe that the symmetric difference $p \oplus X$ is a new matching of size $|X| + 1$.

**Lemma 1.** Let $M$ be a set of equations and $B = B(M) = (U, V, E)$ its variables-in-equations graph. Let $F$ be the set of faults and $P$ be a set of possible sensor locations. Assume that $B$ is connected and has a perfect matching $X$. For every fault $f \in F$, let $v_f$ be the corresponding vertex in $G(B, X)$.

1. Let $f, f' \in F$ and $S \subseteq P \subseteq V$. The fault $f$ is structurally detectable in $M \cup M_f$ iff there exists $s \in S$ such that $v_f$ is reachable from $s$ in $G(B, X)$.

2a. Let $f, f' \in F$ and $S \subseteq P \subseteq V$. Assume that the size of a maximum matching in $B(M \setminus \{e_f\} \cup M_S)$ is $|X| - 1$. The fault $f'$ is structurally isolable from $f$ in $M \cup M_S$ iff there exists $s \in S$ such that $v_{f'}$ is reachable from $s$ in $G(B(M), V \setminus \{v_f\})$.

2b. Let $f, f' \in F$ and $S \subseteq P \subseteq V$. Assume that the size of a maximum matching in $B(M \setminus \{e_f\} \cup M_S)$ is $|X|$. The fault $f'$ is structurally isolable from $f$ in $M \cup M_S$ iff there exists $s, t \in S$ such that $v_{f'}$ is reachable from $s$ in $G(B(f, s), X)$ with $B(f,s) := B(M \setminus \{e_f\} \cup M(t))$. Here, $X$ is a perfect matching in $B(f,s)$ and it emerges from $X' := X \setminus \{e_f\}$ via an augmented path $p$ of $X'$ in $B(f,s)$, i.e. $X = X' \oplus p$.

**Proof.** We start with (1). Let $e_f$ be the equation associated with $f$. By the definition of structural fault detection, fault $f$ is detectable if $e_f \in M^+ \cup M$ such that $e_f \not\in M \cup M_S$. We observe that $X$ is a maximum matching in $B(M \cup M_S)$, i.e. the addition of a measurement equation will not increase the size of a matching. Moreover, there is a one-to-one correspondence between an alternating path in $B(M)$ w.r.t. $X$ and a path in $G(B(M), X)$.

We continue with (2). By the definition of structural fault isolation, fault $f'$ is isolable from fault $f$ if $e_f \in M^+ \cup M''$ with $M'' := (M \setminus \{e_f\}) \cup M_S$. Let $X'$ be the edge removed which contains $e_f$. We observe that $X'$ is a maximum matching in $B(M \setminus \{e_f\})$ and that a maximum matching in $B(M')$ either has size $|X'|$ or $|X'| + 1$.

Assume the former, case (2.a). This is the simple case since $X'$ is a maximum matching in $B(M')$. Thus, reachability in $B(M')$ via an alternating path w.r.t. $X'$ starting from unmatched equations in $B(M')$ corresponds to reachability in $G(B(M), X') \setminus \{v_f\}$ starting from a vertex in $S$. Just note that the unmatched equations $M_S$ and every measurement equation vertex is connected to exactly one variable vertex.

Assume that the size of a maximum matching in $B(M')$ is $|X'| + 1$, case (2.b). This case is more complicated since a maximum matching in $B(M')$ can be rather different from $X'$. However, the following holds: There exists an augmenting path $p$ of $X'$ such that the induced matching $X'' = p \oplus X'$ is maximum in $B(M')$. This result is known as Berge’s Lemma. We observe that $p$ starts with an unmatched equation vertex $e \in M_S$. It is connected to a variable vertex. Let $t$ be this vertex. Thus, $X_t = X''$ is maximum in $B(M')$ and a perfect matching in $B(f,t)$. The claim follows from the above correspondence between alternating paths in $B(f,t)$ w.r.t. $X_t$ and paths in $G(B(f,t), X_t)$.

### 4.2 Sensor Placements as Hitting Sets

We show how sensor placement reduces to the computation of hitting sets. It is similar to Theorem 2 in [11]. In particular, we are going to show that a sensor set $S \subseteq P$ is a correct sensor placement, Def. 3, iff $S$ is the hitting set of a set system $D$, i.e. $S \cap D \neq \emptyset$ for all $D \in D$.

We start with some simplifications. Let $I = (M, F, P)$. In the definition of $\tau_I(f)$ we take the maximum over all $S \subseteq P$. It follows from the definition of fault isolation that, for fixed $f \in F$, $\tau_I(f, \cdot)$ is monotone, i.e. $\tau_I(f, S) \leq \tau_I(f, S')$ if $S \subseteq S'$.

**Proposition 2.** It holds that $\tau_I(f) = \tau_I(f, P)$.

The following lemma characterizes correct sensor placements as hitting sets. We describe the algorithm Reduce first. Its input is a model $M$ and a perfect matching $X$ of $B(M)$. It computes $G(B, X)$ from $M$. For every fault $f \in F$, it checks if an augmenting path $p$ of $X \setminus \{e_f\}$ exists in $B(M \setminus \{e_f\} \cup M_S)$. If no augmenting path exists we set $G' := G(B, X) \setminus \{v_f\}$. Otherwise, we set $G' := G(B(f, s), X_t)$ with $B(f, s)$ as in Lemma 1, and $t$ is some vertex from $S$ such that we can reach $v_f$ from $t$. Define $D(f,t)$ as the set of all vertices $s \in P$ in $G'$ such that $v_f$ is reachable from $s$. The output of Reduce is $D := \{D(f,t) : f \in F, D(f,t) \neq \emptyset\}$.

**Lemma 2.** Let $M$ be a set of equations and $B$ its variables-in-equations graph. Let $F$ be the set of faults and $P$ be a set of possible sensor locations. Assume that $B$ is connected and has a perfect matching $X$. A set $S \subseteq P$ is a correct sensor placement iff $S$ is a hitting set of $D := \{D(f,t) : f \in F, D(f,t) \neq \emptyset\}$.

Moreover, algorithm Reduce computes $D$ in time $O(n^2m)$ where $n$ is number of variables and $m$ is the number of variable occurrences in $M$.

**Proof.** By Proposition 2 and Definition 3, a sensor set $S \subseteq P$ is a correct sensor placement iff $\tau_I(f, S) = \tau_I(f) = \tau_I(f, P)$ for all $f \in F$. For every $f \in F$, $\tau_I(f, P) = \{|f' \in F : D(f', t) \neq \emptyset\|$ |. Here, we apply Lemma 1. In words, a hitting set of $D$ satisfies the properties of a correct sensor placement and vice versa.

The running time follows since there are $O(n^2m)$ pairs $(f, f') \in F \times F$ and since a depth-first search can be done in time $O(mn)$. Note that $m$ is the number of edges in $B$ and $G$. Moreover, we can search for the augmenting paths in advance. Since an augmenting path is an alternating path, the search reduces to a reachability problem, i.e. a depth-first search. We can do this in time $O(nm)$.

The reduction is almost optimal since $D$ can contain $O(n^2)$ sets with at least $\Omega(n)$ elements each. A possible improvement would be $O(n^2 + m)$. We also note that our reduction improves upon the slightly more general reduction in [11]. It has a running time of $O(n^2m)$. See also [7]. We achieve this by searching for an augmenting path instead of computing a maximum matching.

Let $S^* \subseteq P$ be a smallest correct sensor placement of some MSP instance. We call a set $S \subseteq P$ a $c$-approximate solution if $|S| \leq c \cdot |S^*|$ and $S$ is a correct sensor placement.
Theorem 1. There is a polynomial time algorithm for MSP that, given an instance \((M, F, P)\) of MSP such that \(B(M)\) is connected and has a perfect matching, outputs a \(O(\log(n))\)-approximate solution.

Proof. The one-to-one correspondence in Lemma 2 between correct sensor placements and hitting sets allows us to apply any algorithm which computes exact or approximate solutions of minimum hitting sets in the following way. We first compute a perfect matching (see e.g. [8] pg. 664) and then apply Reduce. The result is a set system. We then apply the algorithm in [9] for computing an approximate solution to MHS. It has an \(O(\log(n))\) approximation guarantee. \(\square\)

4.3 Symmetric Variables-In-Equations Graphs

In this section we consider the case of symmetric variables-in-equations graphs. Examples are variable flow networks as defined in Sec. 3.1 and sensor networks as defined in Sec. 3.2. In particular, sensor networks motivate the results in this section. We aim at deriving an efficient algorithm which is capable to solve MSP for instances up to some thousands of sensor locations.

We call a bipartite graph \(B = (U, V, E)\) symmetric w.r.t. to some perfect matching \(X\) of \(B\) if for all \((i, j) \in E\) there exists vertices \(k\) and \(l\) such that \([k, l] \in E\), \([i, k] \in X\), and \([j, l] \in X\). See Fig. 3 for an example of a symmetric bipartite graph.

A consequence of symmetry is the following: For every edge \(e\) in the directed graph \(G(B, X)\) as defined in Sec. 4.1, \(G(B, X)\) also contains an edge with the opposite orientation. Thus, reachability in the directed graph \(G(B, X)\) is equivalent to reachability in the undirected graph \(G^u(B, X)\) which emerges from \(G(B, X)\) by removing edge orientations.

For our example of a sensor network, Sec. 3.2, we can summarize the graph transformations as follows: We start with an undirected graph \(G_{SN}\) which describes the structure of the sensor network. We derive the model \(M_{SN}\) and its bipartite variables-in-equations graph \(B = B(M_{SN})\). A perfect matching \(X\) is naturally given. We derive \(G(B, X)\) and thus \(G^u(B, X)\). The graph \(G^u(B, X)\) is identical to \(G_{SN}\).

We will need the concept of biconnected components. We recall that a biconnected component \(C\) of an undirected graph \(G\) is a maximal biconnected subgraph of \(B\), i.e. removing any vertex in \(B\) will yield a connected subgraph of \(B\). If two biconnected components have a vertex \(v\) in common we call it a cut vertex. Removing a cut vertex in \(G\) yields a graph with at least two connected components. Removing a non-cut vertex in a connected graph, yields a connected graph. We call an undirected graph biconnected if it contains only one biconnected component.

Lemma 3. Let \(M\) be a set of equations and \(B = B(M)\) its variables-in-equations graph. Let \(F\) be the set of faults and \(P\) be a set of possible sensor locations. Assume that \(B\) is symmetric.

1. If \(G^u(B)\) is connected, every sensor \(s \in P\) achieves maximum fault detectability.\(^2\)

2. If \(G^u(B)\) is biconnected, any two sensors \(\{s, t\} \subset P\) are a correct sensor placement.

Proof. As for case (1), we can directly apply Lemma 1. For case (2), we make the case distinction as in Lemma 1. Let \(f, f' \in F\) and \(S \subset P \subset V\) and assume that the size of a maximum matching in \(B(M \setminus \{e_f\} \cup M_S)\) is \(|X| - 1\). Case (2a). Then, fault \(f'\) is structurally isolable from \(f\) in \(M \cup M_S\) if there exists \(s \in S\) such that \(v_{sf}\) is reachable from \(s\) in \(G(B(M), X) \setminus \{e_f\}\). The latter is true due to 2-connectedness of \(G^u(B)\).

For the second case, we have to check if there exist \(s, t \in S\) such that \(v_{sf}\) is reachable from \(s\) in \(G(B_f, X_1)\) with \(B_f := B(M \setminus \{e_f\} \cup M_{\{t\}})\). Here, \(X_1\) is a perfect matching in \(B_f\). It emerges from \(X' := X \setminus \{e_f\}\) via an augmented path \(P\) of \(X'\) in \(B_f\), i.e. \(X_1 = X' \cup P\). The difficulty is that the graph \(G(B_f, X_1)\) is no longer undirected since \(B_f\) is not symmetric. However, it holds that \(G(B_f, X_1)\) consists of single strongly connected component plus an additional vertex. To see this, we show that the vertices of \(P\) lie on a directed cycle in \(G(B_f, X_1)\). First, we recall that \(P\) corresponds to some path in \(G^u(B)\) that starts with \(t\) and ends with \(v_{sf}\). Second, due to the biconnectedness of \(G^u(B)\) there exists two vertex-disjoint (undirected) paths \(P_1, P_2\) from \(v_{sf}\) to \(t\). This result is known as Menger’s Theorem. We set \(\tilde{P}\) such that it corresponds to \(P_1\). The difference between \(G^u(B)\) and \(G(B_f, X_1)\) are the vertices of \(P_1\). Because of the alternating path \(P\), the bipartite graph \(B_f\) and thus \(G(B_f, X_1)\) changes. There is a correspondence between the vertices in \(G(B_f, X_1)\) which are affected by \(P\) and the vertices in \(G^u(B)\) which lie on \(P_1\). Moreover, the application of the augmenting path \(P\) will change the orientation. In the resulting graph \(G(B_f, X_1)\) there are vertices \(u\) and \(v\) and a directed path from \(u\) to \(v\). The vertices \(u\) and \(v\) correspond to the end and the vertex after the start vertex of \(P\). We can make this path a cycle by using \(P_2\). This implies that the connectedness is preserved in \(G(B_f, X_1)\). Moreover, all the sensor and fault locations are preserved in \(G(B_f, X_1)\). We conclude for any fault \(f' \neq f, f' \in F\) is detectable since \(v_{sf}\) is reachable from \(s\) in \(G(B_f, X_1)\). (See Fig. 4 for an example.) \(\square\)

In Fig. 4 we describe on an example what happens if we remove one equation \(e_2 = e_f\) from a model \(M_{SN}\) of a sensor network. The sensor network \(G_{SN}\) is depicted on the left in the figure. The natural perfect matching \(X\) in our sensor network is \(\{e_i, x_i\}\) for \(i \in \{1, \ldots, 4\}\). After removing the edge from \(X\) which contains \(e_2\) we need to find a new maximum matching. In our example in Fig. 4 we start at the sensor measurement equation \(m_2\) and compute an alternating path to \(x_2\). Note that both \(m_2\) and \(x_2\) are unmatched by the natural perfect matching \(X\) without \(e_2\). We thus have an augmenting path. We compute the new maximum matching from the augmenting path. It is depicted by bold lines in the figure (middle). On the right we see the resulting graph. The variables-in-equations graph is no longer symmetric in general. In our example, there is only an edge from \(e_4x_2\) to \(e_4x_3\) but not in the other direction. We observe that the digraph consists of a single strongly connected component with the exception of vertex \(m_2x_2\). Also note that all the sensor locations are preserved. Thus, since we have a second sensor measurement equation in our example, we can detect every remaining fault, in particular from \(e_4x_3\).

\(^2\)A sensor set \(S \subset P\) in a model \(M\) achieves maximum detectability if \((M \cup M_S)^+ = (M \cup M_P)^+\), i.e. we can structurally detect the same set of faults if place them at sensor locations \(S\) or if we add all possible sensors.
The following theorem is a direct consequence of the lemma: If \( S \subseteq P \) shares at least two sensors places with every biconnected component in \( G(B, X) \), then \( S \) is a correct sensor placement. We also use the algorithm of Hopcroft and Tarjan [10] for the computation of biconnected components. Its running time yields the claimed running time.

Theorem 2. There is an algorithm for MSP that, given an instance \((M, F, P)\) of MSP such that \( B(M) \) is connected, symmetric and such that every biconnected component \( B_1, \ldots, B_k \) of \( G(B, X) \) has at least two sensor locations, outputs a \( k \)-approximate solution. The running time is \( O(n^2m) \) where \( n \) is the number of variables and \( m \) is the number of variable occurrences in \( M \).

4.4 Maximum Detectability of Constant Flow Networks

So far we dealt with placing sensors to achieve maximum fault isolation. In this section we demonstrate how to apply Lemma 2 to study the special case of maximum fault detection for constant flow networks. The algorithm is Reduce. In the proof of the theorem we show that Reduce outputs a set system which is a graph. We use here that sensors can be placed at any energy variable and that faults can happen at any flow variable. This leads to a particular structure of a constant flow network as depicted in Fig. 2 on the left.

Theorem 3. There is an algorithm that, given a constant flow network \((M, F, P)\), outputs a sensor set \( S \subseteq P \) which achieves maximum fault detection and is at most 2 times larger than the smallest such sensor set. Its running time is \( O(n^3m) \) where \( n \) is the number of variables and \( m \) is the number of variable occurrences in \( M \).

Proof. Let \( M_{CFN} \) be some constant flow network with the perfect matching \( X \) which is naturally given by \( M_{CFN} \). The claim of the proposition follows from Lemma 2. We just have to observe that \( |D| \leq 2 \) for every \( D \in D \). This is due to the partition of the vertices of \( G = G(B(M_{CFN}), X) \) into \( V_1 \) and \( V_2 \): \( V_1 \) are the vertices where sensors can be placed and \( V_2 \) are the vertices where a fault can happen. Moreover, every vertex \( v \) in \( V_2 \) has in-degree exactly two. These two neighbors, \( v_1 \) and \( v_2 \), are from \( V_1 \) and \( G \) contains only edges from \( v_1 \) to \( v_2 \) to \( v \). The vertices \( v_1 \) and \( v_2 \) have in-degree 2 and are thus not reachable from any other vertices in \( V_1 \). (See also Fig. 2, left.)

Thus, we have an instance of Minimum Vertex Cover. A 2-approximation algorithm can be found e.g. in [8], pg. 1024.

5 Conclusion

We provided efficient algorithms for sensor placement in flow and sensor networks in the framework of Frisk and Krysander [11]. We showed how to reduce the study of fault detection and isolation to graph reachability. Our reduction runs in time \( O(n^2m) \). We also used the concepts of graph reachability to design and analyze an efficient approximation algorithm for the case of symmetric variables-in-equations graphs. Our algorithm runs in time \( O(n + m) \) and is thus able to handle moderately large instances.

Although our results are tailored towards studying particular models of flow networks and sensor networks, we think that our approach makes it easy to study sensor placement for other special cases of sensor placement too. For example, we presented another application of our approach to study maximum fault detection of constant flow networks.

References