Comparison for Sensor placement algorithms

Yi Dong and Gautam Biswas
Dept. of EECS/ISIS
Vanderbilt University
Nashville, TN 37235. USA

Abstract

Sensor placement algorithms find a minimum set of measurements that can provide maximum detectability and isolability for a set of faults that can occur in the system. Adding sensors to the system usually incurs additional cost and may reduce system reliability (because sensors themselves may degrade and fail). Therefore, sensor placement algorithms play an important role in system design and deployment. In this paper, we study two different algorithms and prove their equivalence. We also compare their computational complexities.

Keywords: *Dulmage-Mendelsohn Decomposition, A* algorithm, Detectability, Isolability, sensor selection, computational complexity*

1 Introduction

An important aspect of designing effective diagnosers involves determining the right set of sensor locations for detection and isolation of known faults in the system [1][13]. Sensor placement algorithms attempt to find a minimum number of sensors that can detect and isolate a known set of faults in a system given a system model [7][11]. Additional criteria, such as likelihood of fault occurrence and sensor failures [15], cost of sensor placement [2], and reliability of the overall diagnostic system [12] have also imposed in determining the minimum sensor set.

Sensor placement algorithms are also sensitive to the nature of the model employed. In most cases, the sensor placement algorithms are based on structural models of the system. This includes graph-based (e.g., [1][11]) and equation-based (e.g., [3][7][13]) models. The work described in this paper focuses on two algorithms to determine optimal sensor placement: (1) a systematic sensor placement algorithm [7] based on the Dulmage-Mendelsohn (DM) structural Decomposition of an equation-based model of a dynamic system [4]; and (2) an A*-based search algorithm from AI that uses a graph-based model of the system dynamics [9] and [11]. Furthermore, [9] much like the methods presented in [1] use signed-directed graphs for deriving fault signatures to represent the effects of fault parameters on observations or sensor measurements.

Our intent in this paper is not to compare different modeling approaches, and their impact on sensor placement algorithms. Instead, we adopt the equation-based framework [3][7][13] and compare the effectiveness of the algorithm based on DM decomposition in [7] with the heuristic A*-based search algorithm in [11]. The original A*-based algorithm used fault signatures derived from a Temporal Causal graph model of the system [9][11]. However, when comparing the two algorithms, we use a Fault Detectability Matrix (FDM) as the starting point for sensor placement using the A* algorithm to achieve a level playing field. The FDM matrix does not use the sign of the change associated with a measurement residual when computing the detectability of a fault given a measurement. We evaluate the ability of these algorithms to generate optimal sensor placement, and also compare their computational complexities.

Sensor placement is implemented offline during system design, taking into consideration factors such as observability, monitorability, controllability, and reliability of the system. Therefore, computational complexity may play an important factor in determining the smallest number of sensors required for full diagnosability, while also meeting some of the other requirements of the designed system.

2 Background

Structural diagnosis methods fall into two categories: graph-based and equation-based methods. However, the task of optimal sensor placement and the notions of detectability and diagnosability can be formally defined independent of the two representations. We begin with the definitions and then present details of the two algorithms under study in the rest of this section.

2.1 Definitions

Given a dynamic system described by its corresponding model, *M*, a set of faults, *F*, in this system that are of interest, and a possible set of sensor locations, *S*, we define the important notions of fault detectability and isolability, and use these definitions to formalize the sensor placement problem. We assume that the occurrence of a fault $f \in F$ causes one or more measurements, $m \in M$ to deviate from its nominal value, producing non-zero residuals that can be measured.

Definition (Fault Detectability) *Given a model M, associated faults F, and possible sensor locations S, iff for all faults $f_i \in F$, we can generate at least one measurement residual which is sensitive to f_i, we say the fault set F is detectable by sensor set S.*

Definition (Fault Isolability) *Given a model M, associated faults F, possible sensor locations S, iff for all pairs of faults $\{f_i, f_j \mid f_i, f_j \in F; f_i \neq f_j\}$, we can generate at least one measurement residual where the residual generated by f_j is different from the residual generated by f_i, we say the fault set F is isolable by sensor set S.*

For a system model *M*, given a set of faults, *F*, and a set of measurements, *S*, it is clear that detectability is a pre-requisite for isolability. Our focus is on isolability, therefore, the next step is to find a minimal set of sensors that make all the faults in *F* isolable. To do this we adapt the definition of minimal sensor set [7].

Definition (Minimal Sensor Set)*: Let S be the set of possible sensor locations, and let $\mathscr{S}$ be a multiset defined on S. Then, $\mathscr{S}$ is a minimal sensor set, if the sensors in $\mathscr{S}$ fulfill the detectability and isolability specification and all proper subsets of $\mathscr{S}$ do not.*

The notion of multiset is used because a sensor may be used more than once to isolate faults. In the next two sections, we briefly review the two sensor placement algorithms as they are originally defined.

2.2 Sensor placement using DM-Decomposition

DM decomposition [4] assumes that the system model is represented by a bipartite graph, with the set of equations represented as one type of nodes and the set of variables as a second type of nodes. The set of equations are rows in the structural model while variables representing system behavior appear in the columns. Fig. 1 illustrates a structural model from [7]. The gray areas imply edges between equations and the variables. By appropriate permutation of rows and columns, the structural model can be represented in an upper block triangular form, i.e., the Dulmage-Mendelsohn (DM) Decomposition. The method to derive the upper block triangular form is described in [4].

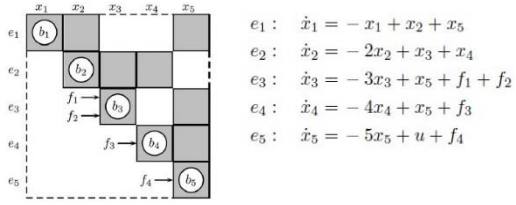

Figure 1. From [7]. Structural model representing the system described by the set of five first order differential equations on the right side. *e* stands for equation, *x* are unknown variables, *f* are faults, and *b* are strongly connected component blocks.

Starting with the set of equations in Fig. 1 the DM-Decomposition partitions the structural model into three parts: (1) *structurally underdetermined* part, which has more variables than equations; (2) *structurally just-determined* part, which has equal number of equations and variables; and (3) *structurally over determined* part, which has more equations than variables. The decomposition can be derived from the bipartite graph by finding the maximum matching, which is actually the non-zero diagonal in the structural model [4]. By appropriate permutation, we can then derive the block triangular form of the matrix [12], and this provides the three structural parts. The over determined part of the system equations are redundant, and form the basis for the fault diagnosis algorithm. The example shown in Fig. 1 is a just-determined system with the upper block triangular form that has a nonzero diagonal. This defines a perfect matching in the bipartite graph [5]. Adding sensors introduces additional equations, which become the over-determined part of the system model. The over-determined part makes fault parameters, $f \in F$, detectable.

Deriving sets of sensors that achieve maximum isolability is a special case of maximum detectability This is achieved by removing the equation which makes fault f_i detectable; if all other faults are still detectable, then fault f_i is isolable for the set of chosen sensors. Therefore, by selectively adding appropriate sensors as we cycle through all faults in *F*, the set *F* can be made isolable.

In order to be efficient, the algorithm applies the notion of Hasse diagram of the partial order over blocks and faults classes [2]. The Hasse diagram is derived according to orientations of unknown variables and equations [3], which is defined as follows: (a) if the edge (e_i, x_j) belongs to the matching (on the non-zero diagonal), it is oriented from e_i to x_j, and (b) otherwise, it is oriented from x_j to e_i.. Faults associated with topmost blocks (maximum faults classes) in the Hasse diagram are blocks that cannot be oriented to other faults-associated blocks. Sensor sets that can detect faults in maximum fault classes can detect all other faults because the fault classes associated with all other blocks have propagation paths to maximum fault class blocks according to the orientation. And since a block is strongly connected, any measurement in a block is sufficient to detect all faults within that block. This is shown in Figure 2.

The sensor placement algorithm will first find sensor sets S_D that achieve maximum detectability and then repeatedly remove the equation associated with a fault to find sensor sets S_I achieving maximum detectability for the rest of the system. By applying the Minimal Hitting Set algorithm [7] on resulting sensor sets S_D and S_I, we can get S, the sensor set that has a non-empty intersection with all of them, and it maximizes detectability and isolability.

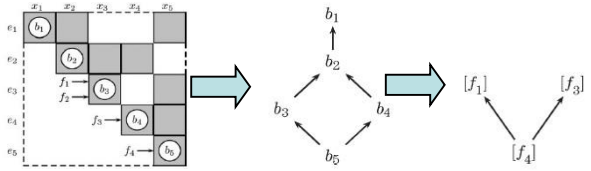

Figure 2. Procedure of generating Hasse diagram of partial order over blocks (right) and faults classes (bottom), where $[f_1]$ and $[f_3]$ are maximum faults classes. (This figure is from [7])

2.2 A* based Sensor Placement algorithm

Before describing this algorithm [11], we introduce the notion of temporal causal graph (TCG) and how to derive fault signatures from the TCG. A TCG is a directed graph derived from a bond graph model using causal assignments to the bonds derived from the SCAP algorithm [5], and creating a directed graph that captures all of these relations [9]. A fault signature defines the changes to a sensor value induced by a fault (typically, we define a fault by a $\pm$ change to the parameter value associated with the fault). Propagation starts with 0-order change through all causal paths in the TCG from the fault towards the measurements. Every integrating edge increases the temporal order of the signature by 1 (indicating a delay in the propagation). Since the TCG is a global model, i.e., it contains all relations between variables and faults, it is sufficient to determine fault signatures for all faults on all possible observations in the system model, based on a specific causal assignment. Figure 3 shows this procedure for a simple one tank system. The fault signature

of C on e_2 is +,–, where $C^- \rightarrow e_2^+ \rightarrow e_3^+ \rightarrow f_3^+ \rightarrow f_2^-$, and $f_2 \rightarrow e_2^{'-}$ is a first-order effect because of the integrating edge. This implies an instantaneous jump in e_2 (0^{th} order change) if C decreases, and then a gradual decrease in its residual (1^{st} order change).

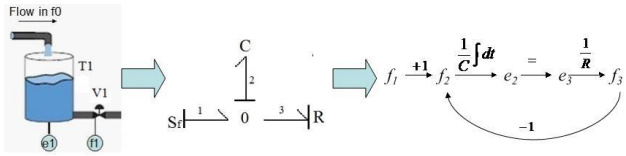

Figure 3. (left) One tank system; (middle) Bond Graph; and (right) Temporal Causal Graph(TCG)

The A* search algorithm finds the optimal path from a start state (no sensors selected) to a goal state (a minimal sensor set). A* introduces two measures: *g*, the distance from the initial state to the current state (i.e., the number of sensors added), and *h*, the estimate of the distance from the current state to the goal state (i.e., estimate of minimum number of sensors needed to achieve full diagnosability) [11]. The algorithm finds the optimal, i.e., lowest-cost path if *h* underestimates the true distance to the goal.

The algorithm is best described by a partitioning approach. The system configuration in each step is {*P*, *OSET*}, where *P* is the partition of faults not yet diagnosed, and *OSET* is the set of sensors that have been selected. The initial configuration is $\{P_{init}, OSET_{init}\}$, where P_{init} is made up of a set that contains all faults (no fault can be isolated), and $OSET_{init}$ is empty. Similarly, the goal state $\{P_{goal}, OSET_{goal}\}$, has each element of P_{goal} with exactly one fault, and $OSET_{goal}$ includes the smallest number of sensors that can isolate all faults. This set may not be unique. To achieve the best sensor placement, at each iteration we add the sensor that has the highest discriminatory power. Fault signatures derived for each fault and possible measurement using the Qualitative Fault Signature scheme [9]. As proven in [9][11] a fault can have one of 4 different signatures on an observation: '+,+', '+,–', '–,–' and '–,+' if magnitude and first order computations are formed for each measurement. Therefore, each observation can discriminate at most 4 sets of faults. This provides the information for deriving *h*. There exist two different way of specifying *h*, and because configurations differ from each other, the two derived values for *h* are not necessarily to be the same. The value h_1 is derived by achieving max partitioning for all partitions. Suppose the current configuration is $\{P_{current}, OSET_{current}\}$, where $P_{current}$ has *t* partitions. A new observation can split each partition into at most 4 new partitions. So $4^h \times t \geq n$, and $h_1 = \log_4(\frac{n}{t})$. The value h_2 is derived by achieving max partitioning of the largest partition. Say, $P_{current}$ has a largest partition P_l that includes t_l faults. It can be split to at most 4 partitions. In order to split it into partitions contain exactly one fault with minimum observations, it should be split into 4 partitions at each iteration. So h_2 can be calculated by $t_l / 4^h = 1$, where $h_2 = \log_4(t_l)$. Then h = Max{h_1, h_2}.

At each iteration *i*, we choose an observation that has lowest value for $h_i + g_i$ to guarantee the underestimate distance from current point to goal point to get an optimal solution on measurement selection. The correctness of the solution has been demonstrated in [11].

The A* algorithm is general, and it can be used with any discriminatory function. In the Case Study we include only binary information, i.e., a sensor measurement can or cannot detect a fault. In this case, a fault set can be decomposed into two sets by a measurement, therefore, the base of the logarithm used for the computation is 2.

3 Case Study

We will use a traditional three tank system configuration shown in Fig. 4 to illustrate the two algorithms. The bond graph model (Fig. 5) implies six sensor variables in the system, that is pressures in the tanks, *p1*, *p2*, *p3*, and flow rates out of the tanks, *q1*, *q2* and *q3* (*q0*, the input flow rate is known). The sensor locations (circled) are shown in Fig. 4. In the bond graph model, the tanks are modeled as capacitors, and the pipes and valves as resistances. 0-junctions represent common effort locations, and 1-junctions imply current flow locations [5].

Following the constituent relations of the basic elements of the bond graph, we derive a set of equations (1):

$p1 = e1 = e2 = e3$
$p2 = e5 = e6 = e7$
$p3 = e9 = e10 = e11$
$q1 = f3 = f4 = f5$
$q2 = f7 = f8 = f9$
$q3 = f11$

In order to clearly represent causality, we introduce three more variables $\dot{p}_1$, $\dot{p}_2$ and $\dot{p}_3$, which represent the derivatives of variables *p1*, *p2* and *p3*, respectively for the DM-Decomposition method.

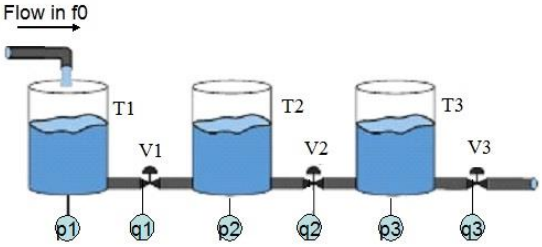

Figure 4. Three-tank system

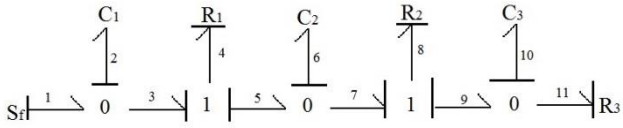

Figure 5. Bond graph model of the three tank system

4 Comparison of the two Methods

4.1 DM-Decomposition method

The set of equations deriving three tanks system behavior is given by (These equations do not imply any causality but only relations between variables and parameters):

E_1: $q_1 = \frac{1}{R_1}(p_1 - p_2)$ E_2: $q_2 = \frac{1}{R_2}(p_2 - p_3)$

E_3: $q_3 = \frac{1}{R_3} \cdot p_3$ E_4: $\dot{p}_1 = \frac{1}{C_1}(q_0 - q_1)$

E_5: $\dot{p}_2 = \frac{1}{C_2}(q_1 - q_2)$ E_6: $\dot{p}_3 = \frac{1}{C_3}(q_2 - q_3)$

E_7: $\dot{p}_1 = \frac{dp_1}{dt}$ E_8: $\dot{p}_2 = \frac{dp_2}{dt}$

E_9: $\dot{p}_3 = \frac{dp_3}{dt}$

The 9 equations and 9 unknowns imply a just-determined system. Six possible single faults associated with the system are R_1, R_2, R_3, C_1, C_2 and $C3$ (faults in the valves and tank capacities, respectively). After proper permutation of equations and unknown variables, we obtain the DM-decomposition shown in Figure 6.

Using the DM-Decomposition method for maximum detectability we get the sensors sets: $\{q_1\}$, $\{q_2\}$, $\{q_3\}$, $\{p_1\}$, $\{p_2\}$, $\{p_3\}$, $\{\dot{p}_1\}$, $\{\dot{p}_2\}$, $\{\dot{p}_3\}$, and for each variable corresponding to a sensor, we add an equation that measures the unknown variable. This makes all faults detectable. Since the system represents a strongly connected component of the structural model, any one sensor makes all faults detectable. And by repeatedly remove one equation associated with a fault we find sensors that achieve full detectability for the rest of the system. The union of the sensor sets after each iteration defines the sensor set that achieves maximum isolability. Among the list of completely diagnosable sensor sets of size 3 are$\{\dot{p}_1, q_1, q_3\}$, $\{p_1, q_1, q_3\}$, $\{p_2, q_1, q_2\}$, $\{\dot{p}_2, q_1, q_2\}$[1].

	q_1	q_2	q_3	p_1	p_2	p_3	$\dot{p}_1$	$\dot{p}_2$	$\dot{p}_3$
E_1	1	0	0	1	1	0	0	0	0
E_2	0	1	0	0	1	1	0	0	0
E_3	0	0	1	0	0	1	0	0	0
E_7	0	0	0	1	0	0	1	0	0
E_8	0	0	0	0	1	0	0	1	0
E_9	0	0	0	0	0	1	0	0	1
E_4	1	0	0	0	0	0	1	0	1
E_5	1	1	0	0	0	0	0	1	0
E_6	0	1	1	0	1	0	0	0	1

Figure 6. DM-decomposition of the example system, where the whole system is in one strongly connected component. Notice there is a nonzero diagonal in the matrix which is a perfect matching of the bipartite graph from the system.

4.2 A* approach

For comparison purposes, we use the same diagnosability information for the A* algorithm as the DM-Decomposition. This information is introduced as the Fault Detection Matrix (FDM). So this is a modified version A* algorithm. The FDM for the three tank system is shown in Figure 7. In this matrix, a "1" for $\{s_i, F_i\}$ implies that if F_i is detectable when removing all other faults from the system under s_i, which actually means, F_i has a propagation path to s_i. For example, R_1 is detectable under measurement q_1 when removing equations E_2, E_3, E_4, E_5, E_6 associated with R_1, R_2, C_1, C_2, C_3, respectively(propagation path: $R_1 \rightarrow E_1 \rightarrow q_1$).

$m \backslash F$	C_1	C_2	C_3	R_1	R_2	R_3
q_1	1	1	0	1	0	0
q_2	0	1	1	0	1	0
q_3	0	0	1	0	0	1
p_1	1	0	0	1	0	0
p_2	0	1	0	1	1	0
p_3	0	0	1	0	1	1
$\dot{p}_1$	1	0	0	1	0	0
$\dot{p}_2$	0	1	0	1	1	0
$\dot{p}_3$	0	0	1	0	1	1

Figure 7. Fault Detection Matrix

Fault Detection Matrix is a two dimensional matrix that contains detectability information of each pair of $\{O, F\}$, where O is an observation, F is a fault

Initially, $P_{init} = \{C_1, C_2, C_3, R_1, R_2, R_3\}$, and all observations can split the partition. Calculating, $h = \text{Max}\{h_1, h_2\}$, we get

$$h_{q_1} = \max\{\log_2 3, \log_2 3\} = \log_2 3$$
$$h_{q_2} = \max\{\log_2 3, \log_2 3\} = \log_2 3$$
$$h_{q_3} = \max\{\log_2 3, \log_2 4\} = \log_2 4$$
$$h_{p_1} = \max\{\log_2 3, \log_2 4\} = \log_2 4$$
$$h_{p_2} = \max\{\log_2 2, \log_2 3\} = \log_2 3$$
$$h_{p_3} = \max\{\log_2 3, \log_2 3\} = \log_2 3$$
$$h_{\dot{p}_1} = \max\{\log_2 3, \log_2 4\} = \log_2 4$$
$$h_{\dot{p}_2} = \max\{\log_2 2, \log_2 3\} = \log_2 3$$
$$h_{\dot{p}_3} = \max\{\log_2 3, \log_2 3\} = \log_2 3$$

Since $g = 0$ initially, and there are some tied values for $\min\{h\}$, we randomly choose one, say q_2 and add it to $OSET$, $OSET = \{q_2\}$, and $P = \{C_1, R_1, R_3\}, \{C_2, C_3, R_2\}$. In the next iteration, we repeat this procedure to re-calculate h values. The minimum value occurs for sensor q_1. Then $OSET = \{q_2, q_1\}$, and $p = \{C_1, R_1\}, \{R_3\}, \{C_2\}, \{C_3, R_2\}$.

In the next and last iteration, we choose p_2 as the next observation, then $OSET = \{q_2, q_1, p_2\}$ which splits p into $\{C_1\}, \{R_1\}, \{R_3\}, \{C_2\}, \{C_3\}, \{R_2\}$, and this implies a fully diagnosable system. This $OSET = \{q_2, q_1, p_2\}$ is one of the optimal set of sensors we derived by using DM-Decomposition method. We could derive the other minimum sensor sets by making different choices when ties occur in the choice of sensors.

4.3 Comparison of the two approaches

According to example results shown above, the A* algorithm derives one of the optimal sensor sets derived by DM-Decomposition approach.

[1] We do not list all possible sensor combinations for minimal diagnosis.

4.3.1 Comparison on functionality

Equivalence for Detectability

The A* algorithm does not have an explicit notion for deriving maximum detectability, so we use the same method as the DMD approach.

Equivalence for Isolability

The A* approach can be seen as tracking a binary search tree shown in Figure 8. There is a set of faults for each vertex, and an observation O_i for each non-leaf vertex, which can split the set into two parts, where the set of faults on its left child is detectable under O_i while the set on its right child is not. So it starts from the root with all faults in one set$\{f_1 \dots f_n\}$, adds an observation O_0 splitting the partition into $\{f_1 \dots f_i\}$ and $\{f_{i+1} \dots f_n\}$. Repeat the procedure until we have full isolation, i.e., $\{f_1\}, \{f_2\} \dots \{f_{n-1}\}, \{f_n\}$. In addition, we may need an observation O_n to make sure the system is fully detectable. $S_{A^*} = \{O_0, O_1 \dots O_k \dots O_p\} \cup \{O_n\}$ is an optimal sensor set for maximum detectability and isolability.

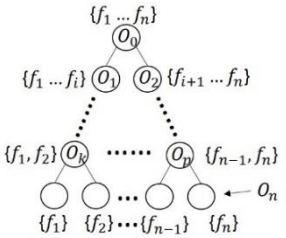

Figure 8. Binary Search Tree for A* algorithm

For DM-Decomposition, we start from the leaves of the binary search tree, removing equations associated with a fault one by one and getting sensor sets that detect the rest of the faults. The minimal hitting set provides the optimal sensor placement.

So, we want to prove that S_{A^*} is also an optimal answer derived by DMD method. In order to do this, we need to prove that there is always a set $S_i \subseteq S_{A^*}$, which is sufficient to detect all faults in the system $M_s \backslash e_{f_i}$.

Proof: Notice that when a vertex in the binary tree is a left-child of its parent, all faults in that partition are detectable under its parent's observation. So all faults of its descendants are also detectable. Obviously, every leaf that is a left child can be detectable under its parent's observation, i.e., f_1 is detectable under O_k. And for the leaf which is a right child, we just need to climb the tree to a level where it reaches a vertex that is a left child of its parent, then the observation of this parent can detect this fault. The only exception is right-most leaf which can only be detected by O_n. So the observation set $S_i \subseteq S_{A^*}$ is always sufficient to detect rest of faults when we remove one from the system. ■

Second, we need to prove that every observation $O_i \in S_{A^*}$ is needed at least once in the iterations of the DMD method if we want only observations from S_{A^*}.

Proof: Notice that the observation of an vertex V_i in binary tree is the only one in S_{A^*} which can detect the right-most leaf-fault of its left sub-tree (every non-leaf vertex can have two sub-tree, left and right) because the right-most leaf of that sub-tree cannot get to an vertex which is a left child by climbing up the tree. And since the root of the sub-tree is the left child of V_i, observation of V_i is needed to detect that right-most leaf-fault. ■

According to the proof above, we can always find an observation set $S_i \subseteq S_{A^*}$ when we run iterations of DMD method, and every $O_i \in S_{A^*}$ will happen at least once in S_i. So after getting the minimal hitting sets, the set S_{A^*} must be one of the resulting sets by DMD method since A* algorithm is guaranteed to give an optimal answer.

To sum up, the A* algorithm and the DMD methods generate equivalent results. The DMD method produces multiple optimal sets, while the A* algorithm only generates one answer. However, if we maintain multiple search trees whenever ties occur in *h*, by making different choices, we get exactly the same results as the DMD method.

4.3.2 Comparison on computational complexity

We compare the computational complexity of the two algorithms for isolability. Let us assume first, the system consists of *n* possible observation locations, which is typically the number of unknown variables, and *m* faults. It is clear that there are at most *k* numbers of iterations in A* algorithm, where $k = |OSET_{final}|$. This is because exactly one observation will be added to *OSET* in each iteration. Then the algorithm will calculate the value of $g+h$ for each observation that does not belong to *OSET* repeatedly, and at most *n* observations will be checked in each iteration. It takes constant time to get the value of g by $g = |OSET_{current}| + 1$. And for *h*, we need to go through all the partitions in $P_{current}$ at least once either to find the largest partition or count the number of partitions. Since there are at most $O(m)$ partitions in each iteration, it take $O(m)$ time to get the value of t_l (size of largest partition)or t (number of partitions), and then obviously, constant time to compute $h_1 = log_4(t)$ and $h_2 = log_4(t_l)$. Overall, the time complexity for the A* algorithm is $k*n*O(m) = O(k*n*m)$, where *k* is the number of observations been selected, *n* is the number of possible sensor locations and *m* is the number of faults.

To perform isolation, the DM-Decomposition algorithm, repeatedly removes one equation associated with a fault class and finds the maximum detectability in the rest of the system. So the algorithm is going to repeat for at most $O(m)$ times. For each iteration, a matching for the bipartite graph is needed to form the non-zero diagonal in decomposition. The computational complexity of finding the maximum matching is $O(E\sqrt{V})$, where *V* is the number of vertices and *E* is the number of edges in the bipartite graph [10]. In our case, there are $2*n$ vertices (*n* unknown variables and typically the same number of equations in the bipartite graph), while the number of edges is at most $O(n^2)$. Overall, this step takes $O(n^{2.5})$ time. Then the step to find strongly connected component can be done in $O(E+V)$ time by using Tarjan's algorithm [14], which is $O(n^2+n)$ in our case. And deriving maximum classes takes the same time as finding strongly connected component by using depth first search. This is an $O(n^{2.5} * m)$ running time.

It is much higher than $O(k*n*m)$ for the A* algorithm. And we have not even take the final step of DMD method: for all sets of observations (derived for all pairs of isolable faults), find the minimum set of observations that has none empty intersection with each of those sets. This problem, called minimum hitting set problem is NP-hard, and it can be deduced from vertex cover problem that is known to be NP-Complete. So the running time would be exponential to the number of pairs of faults. However, if a system is well designed, this part of computation would not take that much time [7]. And since sensor placement analysis is always implementing off-line, the exponential complexity may not be a problem. So the overall computational complexity for the DMD based method is $O(n^{2.5} * m) + U$, where U indicates an unpredictable time value from the NP-hard problem. We are analyzing the worst cases for both algorithms, so the lower bound computations are omitted.

5 Conclusion and discussion

We have shown that the detectability and isolability algorithms for the A* and DMD approaches are equivalent, and we can infer that they have the same discriminatory capability. For observations which can accommodate higher order and multiple types of deviations, we can introduce integral causality into the DM-Decomposition as well as making different observation type. Different observation types can have multiple measurement equations on the same measurement location. By following the DMD procedure, we can find the maximum detectability and isolability using fault signatures. The two algorithms are equivalent in functionality.

DM-Decomposition based methods have a complexity given by $O(n^{2.5} * m)+U$, whereas it is $O(k*n*m)$ for the A* algorithm. One of the reasons for this is that the DM-Decomposition based method repeats some iterations unnecessarily, i.e., doing exhaustive search for all possible fault pairs. For example, when both f_i and f_j are detectable, if f_i is isolable from f_j, it is not necessarily true that f_j is also isolable from f_i by the notion of faults being detectable in the overdetermined part of a system. However, this is actually true, because f_i will have deviation on at least two observations while f_j has deviation on only one of them as we have discussed above, both of them are isolable from each other. On the other hand, the DM-Decomposition based algorithm can give us all possible sets of sensors locations that can achieve maximum isolability, while A* algorithm can only derive them sequentially, when there are ties in the $g+h$ value among some observations. Besides, the accuracy or timeliness of diagnosis can decrease as the number of sensors is decreased, because it implies longer propagation paths from faults to the sensors. The accuracy or timeliness of detection and isolation may decrease becomes significant for higher order systems. Therefore, if we consider accuracy and timeliness of diagnosis, the DM Decomposition based algorithm may provide better results because one can choose additional criteria to pick the sensor set that best satisfies these criteria.

References

[1] M. Bhushan and R. Rengaswamy, "Design of sensor location based on various fault diagnostic observability and reliability criteria", *Computers and Chemical Engineering*, 24(2): 735-741, 2000.

[2] R. Duan, O. Dongxiu, D. Decun, Z. Huilin, "Optimal Sensor Placement for Fault Diagnosis Based on Diagnosis Cost Specifications" *Journal of Computational Information Systems*, 7(9): 3253-3260, 2011.

[3] V. de Flaugergues , V. Cocquempot, M. Bayart and M. Pengov, "Structural Analysis for FDI: a modified, invertibility-based canonical decomposition", *Proc. 20th Intl Workshop on Principles of Diagnosis (DX-09)*, Stockholm, Sweden, 59-66, 2009.

[4] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bipartite graphs," Can. J. Math., 10(4): 516–534, 1958.

[5] Karnopp, D. C., Rosenberg, R. C. and Margolis, D. L., *System dynamics: a unified approach*", Wiley, ISBN 0-471-62171-4, 1990.

[6] E. Frisk, A. Bregon, J. Aslund, M. Krysander, B. Pulido and G. Biswas, "Diagnosability Analysis Considering Causal Interpretations for Differential Constraints", IEEE Transactions on Systems, Man and Cybernetics. Part A, Systems and humans, 42(5), 1215-1229, 2012.

[7] M. Krysander and E. Frisk, "Sensor Placement for Fault Diagnosis, *IEEE Trans on Systems, Man, and Cybernetics—Part A: ystems and Humans*, 38(6): 1398-1410, 2008.

[8] D. Maquin, M. Luong, and J. Ragot. "Fault detection and isolation and sensor network design," *European Journal of Automation*, 31(2): 393–406, 1997.

[9] Mosterman, P., and G. Biswas, "Diagnosis of Continuous Valued Systems in Transient Operating Regions", *IEEE Transactions on Systems, Man, and Cybernetics*, 29(6): 554-565, 1999.

[10] Micali, S. and Vazirani, V. V., "An scriptstyle $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs," *Proc. 21st IEEE Symp. Foundations of Computer Science*, 17–27, 1980.

[11] S. Narasimhan, P.J. Mosterman and G. Biswas, "A systematic analysis of measurement selection algorithms for fault isolation in dynamic systems," *Proc. 9th Intl Workshop on Principles of Diagnosis (DX-98)*, Cape Cod, MA, 94-101, 1998.

[12] A. Pothen and C.J. Fan, "Computing the Block Triangular Form of a Sparse Matrix," ACM Transactions on Mathematical Software, 16(4): 303-324, 1990.

[13] A. Rosich, A.A. Yassine, and S. Ploix, "Efficient Optimal Sensor Placement for Structural Model Based Diagnosis," *Proc. 21th Intl Workshop on Principles of Diagnosis (DX-10)*, Portland, OR, 59-66, 2009.

[14] Tarjan, R. E. (1972), "Depth-first search and linear graph algorithms", SIAM Journal on Computing 1 (2): 146–160, doi:10.1137/0201010

[15] B. R. Upadhyaya, S. R. P. Perillo, X. Xu, and F. Li, "Advanced Control Design, Optimal Sensor Placement, and Technology Demonstration for Small and Medium Nuclear Power Reactors*," Proc. 17th Intl Conf on Nuclear Engineering (ICONE17)*, 763-773, 2009.