Identifying defective valves in a blowout preventer valve network

Diptesh Ghosh

W.P. No. 2013-04-03
April 2013

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD – 380015
INDIA

Identifying defective valves in a blowout preventer valve network

Diptesh Ghosh

Abstract

Blowouts are financially damaging for drilling companies and are ecological hazards. Hence blowout prevention equipment is critical infrastructure for drilling companies. Blowout preventer valves are important components of blowout prevention equipments and need to be checked regularly. However, since these valves are often physically inaccessible, they are checked in batches called test sets. In this paper we present an exact method to check the functional status of all blowout preventer valves using a minimum number of test sets. We also present a heuristic method to identify malfunctioning valves if they exist. We illustrate both methods using a real world example.

1 Introduction

Crude oil and natural gases, called formation fluids, typically occur in porous rock formations underground. These formation fluids are under high pressure in the rock formations, so that when a well is drilled into the formation, they rise to the surface. The pressure of formation fluids is carefully balanced by pumping in mud into the well-bore from mud pumps at pre-calculated rates. However, if the pressure of the formation fluids is under-estimated, these fluids can rise to the surface at uncontrolled rates causing damage to the drilling equipment; and since these fluids are inflammable, damaging the oil rig. Such uncontrolled influx of formation fluids into the well-bore is called a blowout. Blowouts are tremendously expensive to a drilling company since they not only cause enormous damage to the drilling rig, but also to the environment, and ultimately to the reputation of the drilling company that is in charge of the well. Azwell et al. (2011), Westergaad (1987) provide details about blowouts in drilling rigs and implications of such blowouts.

Drilling companies attempt to prevent blowouts through blowout preventer (BOP) valve networks. BOP valves are high pressure valves that can be closed rapidly. In a BOP valve network, BOP valves are arranged in a primarily series-parallel configuration which can contain any unwanted influx from a well-bore and divert it though manifolds away from the rig to be released in a safe and controlled manner. Once the influx is contained, the valves are returned to their original configuration so that normal drilling operations can continue. The advantage of a series-parallel structure is that it allows, if any of the valves leak, fail or wear out, to continue to deal with the influx in a controlled fashion while the leaking or worn out valves are repaired or replaced. Typical BOP valve networks have between 10 and 50 BOP valves. Figure 1 shows a schematic diagram of a BOP valve network. The numbered circles correspond to BOP valves, and the lines correspond to manifolds.

The BOP valve network is critical to the safety of the drilling rig, and so the components of this network are tested at regular intervals. The tests are of three types; tests checking for the integrity of the manifolds, tests checking whether or not the BOP valves open on command, and tests checking whether or not the BOP valves close on command. The third test is the most important for the functioning of the network and BOP valves are tested on this parameter regularly and frequently. In this paper, we restrict ourselves to this test.

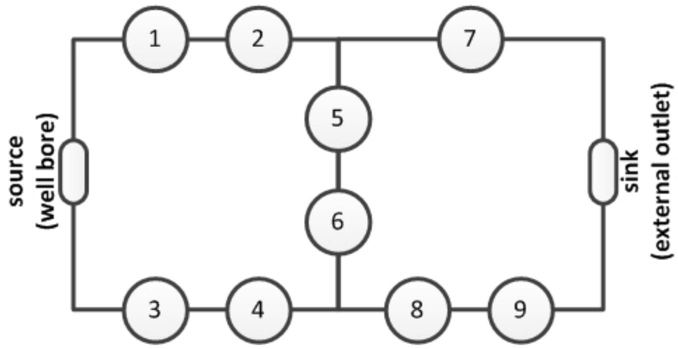


Figure 1: A schematic diagram of a BOP valve network

Since the BOP valve network is often deployed in locations that are difficult to reach, BOP valves are not checked individually, but in groups called test sets. A test set is a minimal collection of BOP valves in the network such that closing all BOP valves in the test set disconnects the source and sink nodes. In practice, once the valves comprising a test set are closed, formation fluid pressure is applied at the source node, and the pressure at the sink node is measured. If there is no rise in pressure at the sink node, then the test set is said to have passed the check and all the valves in the test set are considered to be functioning properly. If however there is a rise in pressure at the sink node, then the test set is said to have failed the check and at least one of the valves in the test set is malfunctioning. In that eventuality the testing effort is directed at identifying the malfunctioning valve(s). In order to check all BOP valves for malfunction, one has to create and check an adequate number of test sets, such that each BOP valve in the network is a member of at least one test set. Normal drilling operations are suspended while test sets are checked, and checking each test set requires approximately 30 minutes. Thus the checking operation is an expensive operation for a drilling company, and the following blowout preventer valve testing problem is a natural problem that arises. The problem is stated as follows.

Blowout preventer valve testing problem

Given: A BOP valve network.

Required:

1. A minimum cardinality set of test sets in which each BOP valve in the network is a part of at least one test set.
2. If one or more test sets fail the check, the set of malfunctioning BOP valves in the network.

It is worth noting that BOP valve network configurations can exist in which all malfunctioning BOP valves cannot be identified using test sets. Consider for example the BOP valve network shown in Figure 2. In this network, BOP valve 1 is a part of all test sets. Hence if BOP valve 1 and any other BOP valve malfunctions, it is impossible to identify the malfunctioning valves using test sets. However, in practice such anomalies are taken care of while designing the networks.

To the best of our knowledge there is no systematic method used to address this problem. In practice, test sets are created mainly through visual inspection of the BOP valve network. The problem has also not been addressed in the optimization literature. Two related problems have however been studied in the literature. The first is the combinatorial group testing problem (see Du and Hwang 2000)), in which one is required to isolate a subset of elements with certain characteristics from a group of elements using a minimum number of tests. This problem is applied, for example, in isolating blood samples corresponding to AIDS patients within a batch of blood samples. Combinatorial group testing problems differ from the problem at hand in that any arbitrary subset of

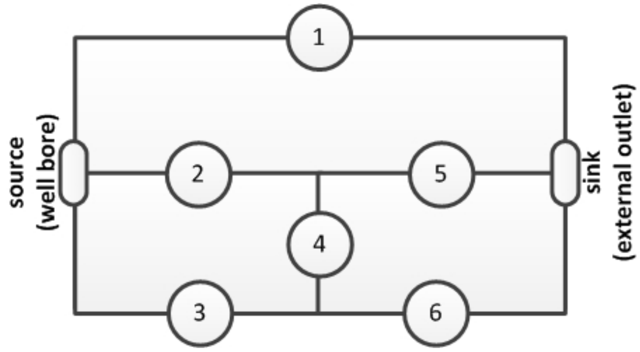


Figure 2: A BOP valve network where all valves cannot be checked through test sets

elements can be tested in the combinatorial group testing problem, while in the problem addressed in this paper, one is restricted to minimal test sets. The other related problem, addressed in Loulou (1992), constructs a minimum cardinality cover of elements in an electronic board to minimize the effort of testing the functioning of all the components. However, this problem too is not identical to the current problem since it has no restrictions on the elements that form the sets of the cover.

In this paper we address the blowout preventer valve testing problem. The work presented here is under two assumptions. First we assume that the number of malfunctioning valves in a BOP valve network is small at any point in time. This is a reasonable assumption since BOP valve networks are critical infrastructure and are regularly checked for defects. The second assumption is that each BOP valve in a BOP valve network has the same probability of malfunctioning.

In the next section we present a method for detecting the presence of malfunctioning BOP valves in a BOP valve network using the minimum number of test sets. In Section 3 we present a heuristic method which identifies the malfunctioning BOP valves in a network should they exist. In Section 4 we illustrate the methods presented in Sections 2 and 3 using a BOP network from an oil rig in Australia. Finally, in Section 5 we summarize this work and provide directions for future work in this area.

2 Testing for the presence of malfunctioning BOP valves

Recall from the previous section that BOP valves in a BOP valve network are checked for malfunction using test sets, and that a test set is a set of BOP valves such that closing of a test set disconnects the source of the network and the sink. In BOP testing, tests are carried out using minimal test sets which have the property that a test set stops being a test set if any of the component valves is opened. This is important because checking a minimal test set yields information that cannot be deduced if the test set is not minimal. Consider for example, the BOP valve network in Figure 1. Closing BOP valves 1, 2, and 3 disconnects the source and the sink of this network, but if this test set passes the check, we cannot conclude that all three valves are functioning. For instance, the test set will disconnect the source and the sink if valves 1 and 3 are functioning properly regardless of the functional status of valve 2. However such a conclusion would have been possible if the test set was indeed minimal. For example, a minimal test set in the network in Figure 1 is a set comprising of valves 1 and 3; the source and sink get disconnected by closing these valves if and only if both the valves are functioning properly.

Our objective in this section therefore is to decide on a collection of minimal test sets consisting of BOP valves with minimum cardinality such that all BOP valves in the network are members of at least one of the test sets, i.e., to find a set of minimal test sets that cover the set of BOP valves. If all the test sets in the collection pass the check, then we can conclude that all the BOP valves in the network are functioning properly, otherwise, at least one of the valves is malfunctioning and we have to use the heuristic described in the next section to identify the malfunctioning valve(s). Note that a preliminary version of the work presented in this section has been reported in Ghosh (2012), we include a formal version here for the sake of completeness.

Our method of determining a minimum cardinality collection of minimal test sets is divided into three stages. In the first stage we characterise sets of valves that can form a minimal test set. We do this by defining "equivalent valves". In the second stage we determine a minimum cardinality collection of test sets in terms of equivalent valves. In the third stage, we construct test sets in terms of BOP valves from the test sets in terms of equivalent valves.

Stage 1: Characterising minimal test sets

In this stage, we reduce the BOP valve network to a smaller network consisting of equivalent valves. The reduction follows from the observation that in any minimal test set, no two BOP valves can be in series in the BOP valve network. If two BOP valves in a test set are in series, then the test set will pass the check as long as one of the valves in each set of BOP valves that are in series is functioning, and a test set passing the check is not a certificate of all valves in the test set functioning properly. Thus for example, in the network in Figure 1, the pairs of BOP valves 1 and 2, 3 and 4, 5 and 6, and 8 and 9 cannot be parts of the same minimal test set.

We define *equivalent valves* as valves which serve as representative for a set of BOP valves that are in series in the BOP valve network. Each equivalent valve j is said to have a *weight* w_j which denotes the number of BOP valves that are being represented by the equivalent valve j.

Thus in the network in Figure 1, BOP valves 1 and 2 can be represented by an equivalent valve A with weight 2, BOP valve 3 and 4 by an equivalent valve B with weight 2, BOP valves 5 and 6 by an equivalent valve C with weight 2, BOP valve 7 by an equivalent valve D with weight 1, and BOP valves 8 and 9 by an equivalent valve E with weight 2. The network in Figure 1 can thus be represented as a network of equivalent valves shown in Figure 3, in which the rectangles represent equivalent valves.

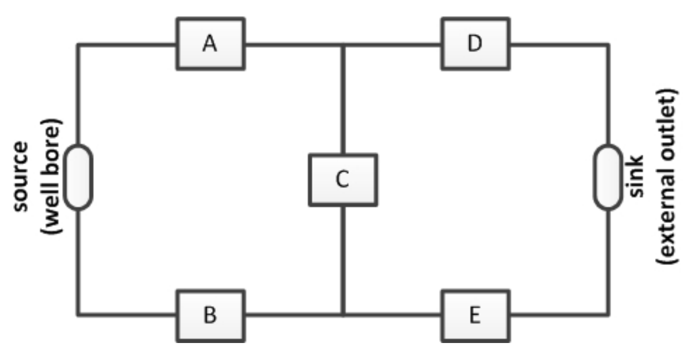


Figure 3: The BOP valve network in Figure 1 in terms of equivalent valves

Every minimal test set in the BOP valve network can be represented as a minimal test set in the network of equivalent valves by replacing each BOP valve in the test set with the equivalent valve that it corresponds to. For example, the minimal test set {1,3} in the BOP valve network in Figure 1 can be represented by the minimal test set {A,B} in the network in Figure 3. However, a minimal test set in the equivalent valve network may correspond to multiple minimal test sets in the BOP valve network. For example the minimal test set {A,B} in the network in Figure 3 can represent any of the minimal test sets {1,3}, {2,3}, {1,4} and {2,4} in the BOP valve network in Figure 1. In the interest of disambiguation, in the remainder of the paper we refer to a minimal test set in a BOP valve network as a *valve test set (VTS)* and a minimal test set in a network of equivalent valves as an *equivalent valve test set (ETS)*. Since every VTS can be represented by an ETS, and since multiple VTSs can be represented by a single ETS, we characterise possible minimal VTSs using ETSs. Such a characterisation reduces computational effort in finding minimal test sets. For example, there are nine BOP valves in the network in Figure 1 which leads to $2^9 - 1 = 511$ sets of BOP valves that need to be examined to see if they are minimal test sets. On the other hand, the equivalent valve network in Figure 3 has five equivalent valves, which leads to $2^5 - 1 = 31$ sets of equivalent valves that need to be examined. Theorem 1 presents conditions that a minimal ETS must satisfy.

Theorem 1 *In a network of equivalent valves, a set of valves S constitute a minimal test set if it satisfies the following three conditions.*

Condition 1: Removal of the valves in S removes all paths from the source to the sink in the valve network.

Condition 2: There is a path from the source to each valve in S that does not pass through any other valve in S.

Condition 3: There is a path from each valve in S to the sink that does not pass through any other valve S.

Proof: Suppose by contradiction that a set of valves S in the equivalent valve network is a minimal test set and does not satisfy at least one of the three conditions.

If it does not satisfy Condition 1 then there exists a path from the source to the sink which does not involve any valve in S. Therefore closing all the valves in S does not disconnect the source and the sink, which implies that S is not a test set.

Next let us suppose that S satisfies Condition 1 but does not satisfy Condition 2 or Condition 3. Then there exists a node $e \in S$ such that there are either no paths from the source to e or no paths from e to the sink. This implies that there are no paths from the source to the sink which includes e. In that case, removal of e from S does not affect the satisfaction of Condition 1, and $S \setminus \{e\}$ is also a test set, which implies that S is not minimal. ∎

Theorem 1 is made algorithmically operational as follows. Let us denote the equivalent valve network as a graph $G = (N \cup V, E)$ where N is the set of equivalent valves, V is the set of nodes in the graph which are not equivalent valves, and E is the set of manifolds. Consider a candidate ETS $S \in N$. First, all members of S and edges connected to them are removed from G. If the maximum flow from the source to the sink in the residual network is non-zero, S fails to satisfy Condition 1 and is not an ETS. Next all edge capacities of G are set to infinity. A dummy sink node is defined, and arcs of unit capacity are defined from all equivalent valves in S to the dummy sink node. Outflows from all equivalent valves in S to all other nodes in G are disallowed. If the maximum flow from the source node to the dummy sink node under these conditions is less than $|S|$, then S fails to satisfy Condition 2 and is not an ETS. Finally, all edge capacities of S are again set to infinity. A dummy source node is defined and arcs of unit capacity are added from the dummy source node to the equivalent valves in S. Inflows to all equivalent valves in S from all other nodes in G are disallowed. If the maximum flow from the dummy source node to the sink node under these conditions is less than $|S|$, then S fails to satisfy Condition 3 and is not an ETS. Otherwise S is a minimal ETS.

The number of candidate ETSs that need to be tested can be reduced by obtaining a lower bound on the number of equivalent valves that are members of a minimal ETS. This lower bound is obtained by noting the fact that every minimal ETS is a node cut in the equivalent valve network. To obtain the lower bound, each edge in the equivalent valve network is assigned an infinite capacity and equivalent valve in the equivalent valve network is replaced by a pair of nodes with an edge of unit capacity connecting them. By the max-flow min-cut theorem (see, e.g., Ahuja et al. 1993) the maximum flow from the source node to the sink node in the modified network is the cardinality of a minimum cut in the modified network, and hence the smallest cardinality of a test set in the equivalent valve network.

We now describe Algorithm CHARACTERISE which outputs the set of all minimal ETSs in an equivalent valve network $G = N \cup V, E)$. The notation $N[k]$ denotes the k-th equivalent valve in N. The set of candidate ETSs are maintained in a set called `candidate`.

ALGORITHM CHARACTERISE

Input: An equivalent valve network $G = (N \cup V, E)$.

Output: Set T consisting of all ETSs in G.

Steps:

Step 1: begin

Step 2: set $T \leftarrow \emptyset$, `candidate` $\leftarrow \{\emptyset\}$, `level` $\leftarrow 0$;

Step 3: if (`level` $= |N| + 1$) or `candidate` $= \emptyset$) then output T and terminate;

Step 4: for each element $e \in$ `candidate` do begin
4.1: if (e is an ETS) then set $T \leftarrow T \cup \{e\}$, `candidate` $\leftarrow$ *cand* $\setminus \{e\}$;
4.2: else set `candidate` $\leftarrow$ `candidate` $\cup \{e \cup \{N[$`level`$]\}\}$;
end;

Step 5: set `level` $\leftarrow$ `level` $+ 1$ and go to Step 3;

Step 6: end.

The correctness of the CHARACTERISE algorithm is proven in Theorem 2.

Theorem 2 *The set T output by CHARACTERISE algorithm contains all minimal ETSs in the equivalent valve network input to it.*

Proof: In the absence of Step 4.1, each iteration of CHARACTERISE adds new sets to `candidate`, and the algorithm terminates when `level` $= |N| + 1$. At that point, `candidate` contains the power set of N. So in the absence of Step 4.1, CHARACTERISE does generate all ETSs. Step 4.1 ensures that whenever an element of `candidate`is an ETS, the element is moved from `candidate` to T, so that CHARACTERISE does not generate any non-minimal ETSs. Since it does not move any element of `candidate` that is not an ETS to T, the set T contains all minimal ETSs in the equivalent valve network when the CHARACTERISE algorithm terminates. ∎

Given the equivalent valve network in Figure 3 the algorithm outputs the ETSs {A,B}, {D,E}, {A,C,E}, and {B,C,D}. In terms of BOP valves, this means that any VTS in the BOP valve network in Figure 1 will be one of the following types

- VTSs corresponding to {A,B}: Containing one among BOP valves 1 and 2, and one among BOP valves 3 and 4.
- VTSs corresponding to {D,E}: Containing BOP valve 7 and one among BOP valves 8 and 9.
- VTSs corresponding to {A,C,E}: Containing one among BOP valves 1 and 2, one among BOP valves 5 and 6, and one among BOP valves 8 and 9.
- VTSs corresponding to {B,C,D}: Containing one among BOP valves 3 and 4, one among BOP valves 5 and 6, and BOP valve 7.

Therefore the four ETSs mentioned above characterise the 20 VTSs possible in the BOP valve network of Figure 1. Given this characterisation of minimal ETSs, we proceed to the second stage to find out how many ETSs of each type are needed to ultimately generate a set of VTSs to cover the entire set of BOP valves in a BOP valve network.

Stage 2: Choosing an adequate number of ETSs to cover all BOP valves

Recall that no two BOP valves in any minimal VTS can be in series in the BOP valve network. Therefore in order to cover the entire set of valves in the BOP valve network, each equivalent valve j must be a member of at least w_j ETSs in the collection of ETSs to check all BOP valves. This observation leads to the following integer linear program to obtain the cover of all BOP valves in terms of ETSs.

Sets:

J: set of equivalent valves, indexed by j;
T: set of candidate ETSs, indexed by t. (This set is obtained at the end of the first stage described above.)

Data:

w_j: weight of equivalent valve j;
m_{tj}: 1 if equivalent valve j is a member of ETS t, and 0 otherwise.

Decision Variables:

y_t: number of instances of the test set t being used in the cover.

Model:

$$\text{ILP: } \left\{ \text{Minimize} \sum_{t \in T} y_t \text{ subject to } \sum_{t \in T} m_{tj} y_t \geq w_j \ \forall j \in J, \ y_t \text{ integer} \right\}.$$

The objective function in ILP minimizes the total number of ETSs that are required in the cover, and each constraint ensures that the number of occurrences of an equivalent valve in the cover is not less than its weight.

Note that the minimum cardinality cover in terms of ETSs is not unique. For example if we use the equivalent valve network in Figure 3 and use the characterisation of VTSs in terms of ETSs obtained at the end of the first stage to solve ILP, we find that the cardinality of the cover is 4. There are multiple optimal solutions to ILP. For example, both {{A,B}, {D,E}, {A,C,E}, {B,C,D}} and {{A,B}, {A,C,E}, {A,C,E}, {B,C,D}} have cardinality 4, and in both, no equivalent valve appears in a number of ETSs less than its weight.

Once the minimum cardinality collection of ETSs have been obtained through ILP, the final stage of our method assigns BOP valves to the ETSs in the collection to obtain a minimum cardinality set of VTSs that cover all BOP valves.

Stage 3: Generating VTSs out of the ETSs

In this stage, each equivalent valve in the output of ILP is replaced by a BOP valve, so that at the end of the stage, the collection of ETSs is transformed to a set of VTSs that cover all BOP valves in the BOP valve network. In this stage each equivalent valve is taken in order. Consider an equivalent valve j with weight w_j. This means that j corresponds to w_j BOP valves. There are two cases to consider.

Case 1: If equivalent valve j is a member of w_j ETSs in the output of ILP, then each occurrence of j is replaced by a different BOP valve corresponding to it. For example, consider the optimal solution {{A,B}, {A,C,E}, {A,C,E}, {B,C,D}} to ILP for the equivalent valve network in Figure 3. Equivalent valve B has a weight 2, and is a member of two ETSs. BOP valves 3 and 4 correspond to B. So B is replaced by BOP valve 3 in the ETS {A,B} and by BOP valve 4 in the ETS {B,C,D}.

Case 2: If j is a member of more than w_j ETSs, then in the first w_j occurrences of j in the collection of ETSs, it is replaced by a different BOP valve which corresponds to the equivalent valve. In all other occurrences, it is replaced by one of the BOP valves corresponding to it. For example, in the optimal solution taken as example in the previous case, equivalent valve A has a weight 2 and is a member of three ETSs. A consists of BOP valves 1 and 2. So BOP valve 1 replaces A in the ETS {A,B} and BOP valve 2 replaces it in the first copy of ETS {A,C,E}. Either of BOP valves 1 and 2 can replace A in the second copy of ETS {A,C,E}.

Thus corresponding to the optimal solution of ILP, four VTSs that can be checked to verify whether or not all BOP valves in the network in Figure 1 are functioning properly form the set {{1,3}, {2,5,8}, {1,6,9}, {4,5,7}}.

The method described in this section creates a minimum cardinality set of VTSs that cover all BOP valves in a BOP valve network. The set of VTSs can be used to check whether of not all the BOP valves in a BOP valve network are functioning properly. If it is found that one or more BOP valves are malfunctioning, then the next step is to identify all the malfunctioning BOP valves. This identification is the objective of the next section.

3 Identifying malfunctioning BOP valves

In this section assume that one or more BOP valves in a BOP valve network are malfunctioning and present a heuristic method to identify such valves. There are two fundamental rules to determine whether or not a BOP valve is malfunctioning when testing is performed using VTSs.

Rule 1 *If a VTS passes a check, then all its component BOP valves are functioning properly.*

Rule 2 *If a VTS containing a BOP valve v fails a check, and all the other valves in the VTS are known to be functioning properly, then v is malfunctioning.*

Using these rules, the objective of the method described in this section is to add each of the BOP valves in a BOP valve network into one of two sets, GOOD signifying that the valve is functioning properly, and BAD signifying that the valve is malfunctioning. In the process, we have valves which we have included in VTSs that were checked but whose functional status has not been defined, and we have valves which have not been included in any VTS. We collect the former type of valves in a set called SUSPECT and the latter type in a set called UNTESTED.

Notice that VTSs which have failed the check but for which we have not been able to apply Rule 2 may be useful in identifying malfunctioning valves later. To see this, consider a VTS = $\{v_1, v_2, \ldots, v_k\}$ which has failed the check, and for which the functional status of more than one component BOP valve is unknown. We cannot apply either of the rules in this case. However, if through later tests we conclude that all of $v_2, \ldots, v_k$ are functioning properly, then we can apply Rule 2 at that stage to classify v_1 as malfunctioning. However, if any one of $v_2, \ldots, v_k$ are subsequently classified as malfunctioning, then the VTS cannot be used further. Hence we maintain a set called FAILED whose elements are intersections of subsets of VTSs which have failed the physical check and the SUSPECT set. This set does not include sets corresponding to test sets in which one or more of the component BOP valves have already been classified as malfunctioning.

Our method in this section is divided into two stages. In the first stage we create VTSs from a collection of ETSs generated by an enhanced version of ILP, and check them to classify all the valves in one among GOOD, BAD, and SUSPECT. In the second stage we try to move the members of SUSPECT into GOOD or BAD by checking further VTSs.

Stage 1: Using VTSs obtained from the cover of BOP valves

One way of obtaining a cover of all BOP valves in a BOP valve network has been described in the second and third stages of the method described in Section 2. However, in the second stage of the method, it was pointed out that ILP may admit multiple optimal solutions. All the optimal solutions are not equally effective when it comes to identifying malfunctioning BOP valves.

It is clear that Rule 2 becomes applicable for more VTSs when the cardinality of GOOD is large. Hence in order to make our heuristic method more effective, we need to choose an optimal solution to ILP which is likely to increase the cardinality of GOOD. Since by our assumption, all BOP valves have an equal chance of malfunctioning, the chance that a VTS passes the physical check decreases with increasing cardinality of the VTS. Thus we have the following preference rule.

Preference Rule 1 *If two VTSs are available for checking at any time, the VTS with lower cardinality is given preference over the other.*

In order to incorporate this preference in ILP, we devise a secondary objective for the model. This objective ensures that an ETS of a particular cardinality is chosen by the model only when it is impossible to create a cover of the same cardinality using ETSs with lower cardinality. Assume that the optimal objective function value of ILP is known to be K. We attach values $WT(t)$ to each ETS $t \in T$ depending on its cardinality. The value attached to an ETS whose cardinality is lowest among all ETSs in T is 1. For all other ETSs in T the value attached to an ETS with cardinality k is $KW + 1$ where W is the maximum value attached to an ETS with cardinality less than k. With this value assignment, the sum of $WT(\cdot)$ values of ETSs in a set of K ETSs which include an ETS with cardinality k will always be more than the sum of $WT(\cdot)$ values for a set of K ETSs in which each member has a cardinality less than k. Our secondary objective thus is to minimize the sum $\sum_{t \in T} WT(t) y_t$.

Another problem in assigning BOP valves to ETSs is that if a malfunctioning valve is a part of multiple VTSs then it causes all the VTSs that it is a part of to fail the check. In order to avoid such situations whenever possible, we have the following preference rule.

Preference Rule 2 *If at any stage, one among several BOP valves which have already been part of VTSs which have been checked needs to be included in a VTS, then the one which has the lowest chance of malfunctioning should be chosen.*

At the beginning of our method of classifying BOP valves, we do not know which of the valves are functioning properly. So a way of implementing this preference is to discourage the use of an equivalent valve j in more than w_j ETSs. This can be achieved by adding a secondary objective to minimize the sum $\sum_{j\in J}(\sum_{t\in T} m_{tj}y_t - w_j)$ to ILP.

Using these two secondary objectives we can enhance ILP and create the following goal program to obtain an optimal set of ETSs to help us identify malfunctioning BOP valves.

$$
\begin{array}{llr}
\text{ILGP: Minimize} & \sum_{t\in T} y_t & \text{Primary objective} \\
\text{Minimize} & \sum_{t\in T} WT(t) y_t & \text{Secondary objective} \\
\text{Minimize} & \sum_{j\in J}(\sum_{t\in T} m_{tj}y_t - w_j) & \text{Secondary objective} \\
\text{subject to} & \sum_{t\in T} m_{tj}y_t \geq w_j & \forall j \in J, \\
 & y_t \text{ integer.} &
\end{array}
$$

The assignment of BOP valves to ETSs in order to form VTSs as described in the third stage of the method in Section 2 did not take the consequences of checking other VTSs into account. However, this becomes important if we use the checks to identify malfunctioning valves. We therefore suggest a different procedure for assigning BOP valves to ETSs to create VTSs. A pseudocode for this assignment and related checking is given below.

ALGORITHM ASSIGN-VALVES-AND-CHECK

Input: $G = (N \cup V, E)$, the collection C of ETSs output by ILGP, the set Q of BOP valves.

Output: Sets GOOD, BAD, SUSPECT, and FAILED at the end of checking VTSs corresponding to ETSs in C.

Steps:

Step 1: begin

Step 2: arrange ETSs in C in non-decreasing order of cardinality;

Step 3: set GOOD = BAD = SUSPECT = FAILED = $\leftarrow \emptyset$, UNTESTED $\leftarrow Q$;

Step 4: while $(C \neq \emptyset)$ do begin

4.1: Check if $\exists t \in C$ such that each equivalent valve in e has a corresponding BOP valve in UNTESTED. If so set $C \leftarrow C \setminus \{t\}$. Create a copy `to-test` of t, and replace each equivalent valve in `to-test` with a BOP valve from UNTESTED corresponding to it. Go to Step 7.

4.2: If no appropriate ETS is obtained at the end of Step 4.1 then check if $\exists t \in C$ such that each equivalent valve in e has a corresponding BOP valve in UNTESTED$\cup$GOOD. If so set $C \leftarrow C \setminus \{t\}$. Create a copy `to-test` of t, and replace each equivalent valve in `to-test` with a BOP valve from UNTESTED corresponding to it if possible, else replace it with a BOP valve from GOOD corresponding to it. Go to Step 7.

4.3: If no appropriate ETS is obtained at the end of Step 4.2 then check if $\exists t \in C$ such that each equivalent valve in e has a corresponding BOP valve in UNTESTED $\cup$ GOOD $\cup$ SUSPECT. If so set $C \leftarrow C \setminus \{t\}$. Create a copy `to-test` of t, and replace each equivalent valve in `to-test` with a BOP valve from UNTESTED corresponding to it if possible, else replace it with a BOP valve from GOOD corresponding to it if possible, else replace it with a BOP valve from SUSPECT corresponding to it. Go to Step 7.

4.4: If no ETS is obtained at the end of Step 4.3 then choose an ETS $t \in C$ with smallest cardinality. Create a copy `to-test` of t. Replace each equivalent valve in `to-test` with a BOP valve from UNTESTED corresponding to it if possible, else replace it with a BOP valve from GOOD corresponding to it if possible, else replace it with a BOP valve from SUSPECT corresponding to it if possible, else replace it with a BOP valve from BAD corresponding to it. Go to Step 7.

Step 5: end;

Step 6: output GOOD, BAD, SUSPECT, FAILED, and terminate;

Step 7: if (`to-test` includes a BOP valve from BAD) then set `result`$\leftarrow$FAIL;

Step 8: else set `result` $\leftarrow$ result of checking `to-test`;

Step 9: set UNTESTED $\leftarrow$ UNTESTED $\setminus$ `to-test`;

Step 10: if (`result`= FAIL and `to-test` does not contain valves in BAD) then begin

Step 11: set `to-test` $\leftarrow$ `to-test` $\setminus$ GOOD;

Step 12: set SUSPECT $\leftarrow$ SUSPECT $\cup$ `to-test`, FAILED $\leftarrow$FAILED $\cup\{$`to-test`$\}$;

Step 13: end;

Step 14: else if (`result`= PASS) then begin

Step 15: set GOOD $\leftarrow$ GOOD $\cup$ `to-test`, SUSPECT $\leftarrow$ SUSPECT $\setminus$ `to-test`;

Step 16: remove any occurrence of BOP valves in `to-test` from all sets in FAILED;

Step 17: if (any set in FAILED is a singleton) then add the member of the set to FAILED;

Step 18: remove any set in FAILED which include members of BAD;

Step 19: end;

Step 20: go to Step 4;

Step 21: end.

Steps 2–5 in ASSIGN-VALVES-AND-CHECK are responsible for choosing an ETS to check, and Steps 10–21 modify the sets GOOD, BAD, and SUSPECT based on the results of the check.

The candidate ETSs are ordered in Step 1 in accordance to Preference Rule 1. Step 4 guides the choice of the next VTS to be checked. In Step 4.1, a VTS is selected if none of its component valves have been part of earlier VTSs. If this is not possible, then the selection is done in Steps 4.2–4.4. A VTS that is created in Steps 4.2–4.4 includes BOP valves that have been part of earlier VTSs. In accordance to Preference Rule 2, the preference order of the sets from which such BOP valves are selected is GOOD $\succ$ SUSPECT $\succ$ BAD. If the VTS created in Step 4 includes a member of BAD, then we know that the VTS will fail the check, hence Step 7 prevents the check to be performed. Otherwise the check is carried out in Step 8.

If the VTS fails the check, then all BOP valves in the VTS are suspect. This does not include any member of the VTS which is already known to be functioning properly. In such cases, SUSPECT and FAILED need to be augmented. This is done in Steps 10–13. If however the VTS passes the check then by Rule 1 all its members are functioning properly and need to be added to GOOD. If some of them are members of SUSPECT, then they have to be removed from the set since we already know their functional status. In addition, since we know the functional status of the members of the VTS as functioning properly, any occurrence of the valves in FAILED need to be deleted. Based on these changes, if a member of FAILED becomes a singleton, then we know that in the corresponding VTS, all BOP valves except the valve still in the set were functioning properly. So we can use Rule 2 to classify this BOP valve as malfunctioning and add it to BAD. If such valves appear in other sets in FAILED, then no further information can be obtained from those sets, and they are removed from FAILED. This process is achieved in Steps 14–19.

ASSIGN-VALVES-AND-CHECK terminates when C is empty. At termination, UNTESTED is empty since all BOP valves in the network have been assigned to at least one test set in Step 4. If SUSPECT is empty when ASSIGN-VALVES-AND-CHECK terminates, then all malfunctioning BOP valves in the network have been identified. If SUSPECT is not empty then we proceed to the second stage of the identification method.

Stage 2: Reclassifying BOP valves in SUSPECT

In this stage of our method, we try to remove valves from SUSPECT and add them to either GOOD or BAD, signifying that they are functioning properly or malfunctioning, respectively. Note that at the beginning of this stage, both C and UNTESTED are empty.

The algorithm for this stage is identical to the ASSIGN-VALVES-AND-CHECK algorithm except for Steps 1 and 4. Step 1 is irrelevant since C is empty. In the remainder of this section, we will describe the modified version of Step 4.

In each iteration of the algorithm for this stage, we choose a BOP valve from SUSPECT, create a VTS including it using ETSs in T, and check the VTS to see if the chosen BOP valve can be classified as functioning properly or malfunctioning using Rules 1 and 2. Recall that the effectiveness of Rule 2 increases with the number of equivalent valves which have a corresponding BOP valve which is a member of GOOD, and hence while choosing a BOP valve to examine, we give preference to those valves which correspond to an equivalent valve which does not have any corresponding BOP valves in GOOD.

The following is a pseudocode of the modified version of Step 4 which is required for this stage.

Step 4a: for each BOP valve v in SUSPECT corresponding to equivalent valve e, compute the number of BOP valves corresponding to e that are members of GOOD. Arrange the BOP valves in SUSPECT in non-decreasing order of these numbers.

Step 4b: while (SUSPECT $\neq \emptyset$) do begin

4b.1 Examine valves in SUSPECT in the order described in Step 4a. For each valve v belonging to equivalent valve e, try to find a smallest cardinality ETS in T which includes e and other equivalent valves each of which have a BOP valve corresponding to them in GOOD. If such an ETS is found, copy the ETS to `to-test`, replace equivalent valve e in `to-test` with v, and replace all other equivalent valves in `to-test` with BOP valves corresponding to them that are members of GOOD. Go to Step 7.

4b.2 If an appropriate valve has not been found in Step 4b.1, again examine valves in SUSPECT in the order described in Step 4a. For each valve v belonging to equivalent valve e, find a smallest cardinality ETS in T which includes e and other equivalent valves each of which have a BOP valve corresponding to them in GOOD ∪ SUSPECT. If such an ETS is found, copy the ETS to `to-test`, replace equivalent valve e in `to-test` with v. For other equivalent valves in `to-test` for which there are corresponding BOP valves in GOOD, replace the equivalent valves in `to-test` with corresponding valves in GOOD. Replace the remaining equivalent valves in `to-test` with BOP valves corresponding to them that are members of SUSPECT. Go to Step 7.

4b.3 If an appropriate valve has not been found in Step 4b.2, then output GOOD, BAD, and SUSPECT, and terminate.

Step 4c: end;

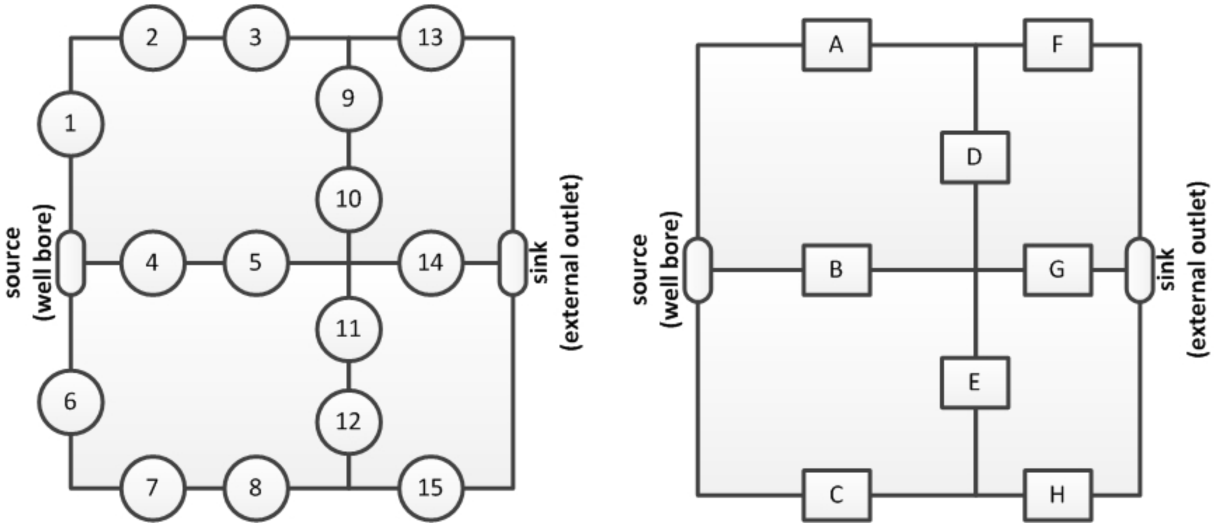
There are several differences in between Step 4 in ASSIGN-VALVES-AND-CHECK algorithm presented in Section 2 and the modified Step 4 that we present here. First, the VTS chosen to be tested in the former was based on a choice of an ETS in C while it is based on a BOP valve in SUSPECT in the latter. Second, we do not make use of BOP valves in BAD in the modified Step 4. This is because a VTS including a BOP valve from BAD is certain to fail the test, and since it includes a valve from BAD, the failed test set cannot be added to FAILED for future use. Hence a VTS including a BOP valve from BAD does not offer us any useful information. Third, there is a possibility of failing to obtain a VTS to check in an iteration in the modified version, which was not possible in the ASSIGN-VALVES-AND-CHECK algorithm. In such situations, the algorithm terminates without reclassifying all BOP valves in SUSPECT.

At the termination of the algorithm for this stage, the SUSPECT set may or may not be empty. If it is empty, then we conclude that all malfunctioning valves in the BOP valve network have been identified. If it is not empty, then the classification of BOP valves in terms of their functional status is incomplete, but it is known that given the functional status of the other BOP valves, it is impossible to determine the functional status of the valves in SUSPECT using any test set. This is likely to happen if a large number of BOP valves in the BOP valve network malfunction simultaneously. Such a situation is rare given that the functional status of valves in a BOP valve network is checked frequently.

In the next section, we illustrate the performance of the algorithms described in Sections 2 and 3 using a realistic network.

4 An example to illustrate our methods

Consider the 15 BOP valve network shown on the left hand side of Figure 4. This is a schematic representation of a BOP valve network for an oil rig in Queensland, Australia. The BOP valves in the network are are numbered 1 through 15. The equivalent valve network for this BOP valve network is shown on the right hand side of Figure 4. The equivalent valve network has eight equivalent valves labeled A through H, where the BOP valves corresponding to each equivalent valve are tabulated in Table 1. The CHARACTERISE algorithm described in Section 2 obtains the set T = {{A,B,C}, {F,G,H}, {A,B,E,H}, {A,D,G,H}, {B,C,D,F}, {A,C,D,E,G}, {B,D,E,F,H}, {C,E,F,G}} of all ETSs for the equivalent valve network. These eight ETSs correspond to 99 possible VTSs for the BOP network. The 15 BOP valves can give rise to 32767 possible combinations of BOP valves that could generate VTSs, and CHARACTERISE only evaluates 220 possibilities. ILP outputs the optimal cover for this network as {{A,B,C}, {A,B,C}, {A,C,D,E,G}, {B,D,E,F,H}}. From this collection of ETSs, the third stage of the method in Section 2 generates the set {{1,4,6}, {2,5,7}, {3,8,9,11,14}, {4,10,12,13,15}} of VTSs that cover all BOP valves in the BOP valve network



A BOP valve network   Its equivalent valve network

Figure 4: A 15 node BOP valve network and its equivalent valve network

Table 1: Equivalent valves and their composition

Equivalent valve	Weight	Corresponding BOP valves
A	3	1 2 3
B	2	4 5
C	3	6 7 8
D	2	9 10
E	2	11 12
F	1	13
G	1	14
H	1	15

in Figure 4. Interestingly, the drilling company uses five test sets to check the functioning of the 15 valves. These test sets are {1,4,6}, {1,5,6}, {2,7,10,11,14}, {3,8,9,12,14}, and {13,14,15}. So ILP reduces the checking effort by 20%. Also notice that the test sets used by the drilling company results in three BOP valves, 1, 6, and 14, being used in multiple test sets, while the solution to ILP results in only one valve in equivalent valve B being used in multiple test sets.

Next assume that valves 4, 10, and 13 have failed, and we use the method described in Section 3 to identify these malfunctioning valves.

The ETS cover generated by ILGP for this network is {{A,B,C}, {A,B,E,H}, {B,C,D,F}, {A,C,D,E,G}}. The first stage of the method arranges these ETSs in non-decreasing order of cardinalities, i.e., in the order {{A,B,C}, {A,B,E,H}, {B,C,D,F}, {A,C,D,E,G}}. At the beginning of this stage GOOD = BAD = SUSPECT = {}, UNTESTED = {1,2,...,15}, and FAILED = {}. The first VTS constructed from ETS {A,B,C} is {1,4,6} which fails since it includes BOP valve 4. However the test itself does not allow us to classify any single valve as malfunctioning. After this test GOOD = BAD = {}, SUSPECT = {1,4,6}, UNTESTED = {2,3,5,7,...,15}, and FAILED = {{1,4,6}}. The second VTS to be tested is {2,5,11,15} constructed from the ETS {A,B,E,H}. This VTS passes the test, and so after this second test GOOD = {2,5,11,15}, BAD = {}, SUSPECT = {1,4,6}, UNTESTED = {3,7,8,9,10,12,13,14}, and FAILED = {{1,4,6}}. The third VTS to be tested is {5,7,9,13} corresponding to the ETS {B,C,D,F}. (Notice that for equivalent valve B we have a choice between BOP valves 4 and 5, since both have been tested once already. We chose BOP valve 5 since it is the only valve which corresponds to B and which we know that it is functioning properly.) This VTS fails the test since it contains BOP valve 13, so that after this test GOOD = {2,5,11,15}, BAD = {}, SUSPECT = {1,4,6,7,9,13}, UNTESTED = {3,8,10,12,14}, and FAILED = {{1,4,6},{7,9,13}}. Notice that the BOP valve 5 is not included in the set of valves that were added to FAILED since we know that BOP valve 5 is functioning properly. The fourth VTS to be tested is {3,8,10,12,14} corresponding to the ETS {A,C,D,E,G}. This VTS fails the test since it contains BOP valve 10, so that after this test which ends the first stage GOOD = {2,5,11,15}, BAD = {}, SUSPECT = {1,3,4,6,7,8,9,10,12,13,14}, UNTESTED = {}, and FAILED = {{1,4,6},{7,9,13},{3,8,10,12,14}}.

In the second stage we consider BOP valves in SUSPECT and try to classify them. Note that at the end of the first stage, equivalent valves A, B, E, and H have one BOP valve corresponding to each in GOOD, so that BOP valves 6, 7, 8, 9, 10, 13, and 14 get preference at this stage. Also, since based on Observation 4 lower cardinality test sets have higher chances of passing the test, we prefer lower cardinality test sets over higher cardinality test sets.

In the first iteration of the second stage, let us check if BOP valve can be chosen for selection in the first pass. Valve 6 corresponds to equivalent valve C. Now there is ETS {A,B,C} in T in which all equivalent valves except C have BOP valves corresponding to them that are members of GOOD. So valve 6 can be chosen for this iteration. The ETS {A,B,C} is converted to the VTS {2,5,6} which is tested. This VTS passes the test, so that BOP valve 6 can be classified as GOOD. The four sets GOOD, BAD, SUSPECT, and FAILED are updated to GOOD = {2,5,6,11,15}, BAD = {}, SUSPECT = {1,3,4,7,8,9,10,12,13,14}, and FAILED = {{1,4},{7,9,13},{3,8,10,12,14}}. at the end of this iteration, equivalent valve C has a BOP valve corresponding to it which is a member of GOOD, and hence the valves in SUSPECT that have preference in the second iteration are 9, 10, 13, and 14. In the second iteration none of the preferred valves can be chosen in the first pass. However, BOP valve 3 can be chosen as a part of the ETS {A,B,C} leading to the VTS {3,5,6}. This VTS passes the test, so that the sets are updated as GOOD = {1,2,5,6,11,15}, BAD = {}, SUSPECT = {3,4,7,8,9,10,12,13,14}, and FAILED = {{4},{7,9,13},{3,8,10,12,14}}. After the update the set {4} in FAILED becomes a singleton, which leads us to conclude that the BOP valve 4 is malfunctioning. This causes a further update in the sets to yield GOOD = {1,2,5,6,11,15}, BAD = {4}, SUSPECT = {3,7,8,9,10,12,13,14}, and FAILED = {{7,9,13},{3,8,10,12,14}}. The details of the steps in the second stage of the heuristic are shown in Table 2.

We see that the method requires a total of 11 tests to be performed to identify all defective valves in the network when three BOP valves 4,10, and 13 malfunction simultaneously. This illustrates that our heuristic is capable of identifying defective valves even when there are several defective ones. However, we believe that having three BOP valves among 15 malfunctioning simultaneously is a rare occurrence, especially since testing is typically carried out quite frequently. The number of tests required to identify all defective valves is smaller if there are fewer defective valves.

5 Summary and future research directions

In this paper we address the problem of testing blowout preventer (BOP) valve networks for malfunctioning valves, and if such valves are present, we provide a heuristic method to isolate malfunctioning valves. This problem is an important one arising in practice which, to the best of our knowledge has not been addressed in the optimization literature. In the introductory section we present the problem both in the practical context as well as in terms of a network problem. In the second section, we provide a mathematical formulation based approach to test whether or not a BOP valve network has one or more malfunctioning valves in it. In the third section, we present a heuristic to identify malfunctioning valves should they exist in a BOP valve network. Finally, we illustrate our methods in the fourth section using a BOP valve network from an oil rig in Queensland, Australia.

Table 2: Steps in the second stage of the heuristic

Iteration	Valve chosen	ETS	VTS	Result of test	Sets
Initial					GOOD = {2,5,11,15}, BAD = {}, SUSPECT = {1,3,4,6,7,8,9,10,12,13,14} FAILED = {{1,4,6},{7,9,13},{3,8,10,12,14}}
1	6	{A,B,C}	{2,5,6}	Pass	GOOD = {2,5,6,11,15}, BAD = {}, SUSPECT = {1,3,4,7,8,9,10,12,13,14} FAILED = {{1,4},{7,9,13},{3,8,10,12,14}}
2	1	{A,B,C}	{1,5,6}	Pass	GOOD = {1,2,5,6,11,15}, BAD = {4}, SUSPECT = {3,7,8,9,10,12,13,14} FAILED = {{7,9,13},{3,8,10,12,14}}
3	3	{A,B,C}	{3,5,6}	Pass	GOOD = {1,2,3,5,6,11,15}, BAD = {4}, SUSPECT = {7,8,9,10,12,13,14} FAILED = {{7,9,13},{8,10,12,14}}
4	7	{A,B,C}	{2,5,7}	Pass	GOOD = {1,2,3,5,6,7,11,15}, BAD = {4}, SUSPECT = {8,9,10,12,13,14} FAILED = {{9,13},{3,8,10,12,14}}
5	8	{A,B,C}	{2,5,8}	Pass	GOOD = {1,2,3,5,6,7,8,11,15}, BAD = {4}, SUSPECT = {9,10,12,13,14} FAILED = {{9,13},{10,12,14}}
6	12	{A,B,E,H}	{2,5,12,15}	Pass	GOOD = {1,2,3,5,6,7,8,11,12,15}, BAD = {4}, SUSPECT = {9,10,13,14} FAILED = {{9,13},{10,14}}
7	9	{A,D,G,H}	{2,9,14,15}	Pass	GOOD = {1,2,3,5,6,7,8,11,12,14,15}, BAD = {4,10,13}, SUSPECT = {} FAILED = {}

The work presented in this paper can be extended in several ways. First, as a part of our approach to detect the presence of malfunctioning valves in a BOP valve network, we have used an enumeration algorithm that generates all possible test sets for a network. This enumeration algorithm can require a long time to execute for large networks. Although it has to be executed only once during the lifetime of a BOP valve network which needs to be checked several times, a more efficient way of generating all test sets will be welcome. We believe that a modification of the algorithm presented in Loulou (1992) can be used to generate a superset of all possible test sets which will speed up enumeration. Secondly, the heuristic that is presented in Section 3 is not guaranteed to isolate all malfunctioning valves by checking the minimum number of test sets. Since normal drilling operations are suspended in oil rigs when checks are performed, performing a larger number of checks than are absolutely necessary causes the efficiency of drilling operations to decrease. An interesting direction of research will be to devise a method which is guaranteed to isolate all malfunctioning valves in a BOP valve network using the minimum number of tests.

Acknowledgements

The author thanks Nilotpal Chakravarti for suggesting this problem, for helping the author to understand the practical problem, and going through initial drafts of the paper. The author also thanks Chris Gregory, Tony Sucandy, Yosua J. Setyobudhi, Yenny Subhawani, and Peter Serhalawan of New Frontier Solutions Pte. Ltd.; Andre Mol of PI International BV; and Chris Hindmarsh, drilling supervisor and freelance performance consultant.

References

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

T. Azwell, M.J. Blum, A. Hare, S. Joye, S. Kubendran, A. Laleian, G. Lane, D.J. Meffert, E.B. Overton, J. Thomas III, L.E. White, The Macondo Blowout Environmental Report, Deepwater Horizon Study Group Environmental Report, 2011.

D. Du, and F. Hwang, Combinatorial Group Testing and Its Applications, World Scientific, Singapore, (2000).

D. Ghosh, On the blowout preventer testing problem: An approach to checking for leakage in BOP networks IIMA W.P.No.2012-01-02 (Unpublished manuscript), Indian Institute of Management Ahmedabad, (2012).

R. Loulou, Minimal cut cover of a graph with an application to the testing of electronic boards, Operations Research Letters, 12 (1992), pp. 301–305.

R. Westergaad, All About Blowout, Norwegian Oil Review, 1987 ISBN 82-991533-0-1, (1987).