Using Faulty Flows in AND/OR Graphs to Model Survivability and Reliability in Distributed Systems

Goce Jakimoski and Mike Burmester [*†]

Abstract

Desmedt, Wang and Burmester [1, 2, 3, 4] have used AND/OR graphs and flows in AND/OR graphs to model both manufacturing processes and problem solving processes, and to analyze the reliability and survivability of critical infrastructures such as cyber-infrastructures and energy infrastructures. In this paper, we revise their model so that the output flow from a faulty vertex is faulty (corresponding to an attack from inside or non-malicious fault) instead of zero (corresponding to a destroyed vertex). We also reconsider the problem of finding critical vertices and show that this problem is **NP**-complete.

Keywords: reliability, survivability, faulty flows, AND/OR graphs, critical vertices

1 Introduction

AND/OR graphs and flows are often used as analysis tools in computer science [5, 6, 7, 8, 9, 10]. For example, minimum-cost solution graphs have been extensively studied in artificial intelligence in order to design algorithms that will provide problem solving that is optimal in terms of resources. Maximum-flow minimum-cut theorem has played a crucial role in network theory and it is used in practice to direct network traffic (e.g., Internet).

Desmedt, Wang and Burmester [1, 2, 3, 4] used AND/OR graphs to model redundant production and computing systems consisting of components which are based on computations with multiple inputs (such as programming of a test flight of an aircraft, construction of a shopping center,

[*]The authors are with the Computer Science Department, Florida State University, Tallahassee, FL 32306-4530, USA. (e-mail:{jakimosk, burmester}@cs.fsu.edu)

[†]This work was supported in part by the National Science Foundation under Grant CCR-0209092.

carrying out of a sequence of manufacturing steps, etc). The AND vertices in the graph correspond to nodes (e.g., processors, machines, chemical processes, engine components, etc.) that need all their inputs in order to produce the output. The OR vertices in the AND/OR graph correspond to the nodes that need only one of their "redundant" inputs. They identified two problems that are important for the analysis of the vulnerability of the system to malicious and non-malicious faults: maximum flow problem and critical vertices problem. Furthermore, they showed that although there are polynomial-time algorithms for solving the problems in the traditional graph theory, it is **NP**-hard to find a non-additive maximum flow in an AND/OR graph, and it is both **NP**-hard and co**NP**-hard to find a set of critical vertices in an AND/OR graph.

In the model analyzed by Desmedt, Wang and Burmester, the value of the flow on the output edges of a faulty vertex is zero. This corresponds to a situation where the node is destroyed (e.g., closed airport due to a bad weather or terrorist threat, a network switch outage, etc.). There are numerous situations that do not fit into this model. For example, in the aforementioned model, a malfunctioning router is modeled as a router that discards all incoming packets and sends no packets in the network. This is not always the case. The malfunctioning router can send incoming packets to a wrong destination or modify the information in the packets. Another example is a malfunctioning factory for car parts. According to their model the factory will not produce car parts at all. However, it is possible that only a small portion of the parts produced by the factory will be faulty. In order to address this problem, we introduce the notion of faulty flow and consider the problem of critical vertices in this new setting. Namely, we define critical vertices to be vertices whose malfunctioning will cause the total faulty flow to be above some given bound and show that the problem of finding fault critical vertices is also hard (i.e., **NP**-complete).

The outline of the paper is following. In Section 2, we give some preliminary definitions and results. The model that we are going to use in our analysis is introduced in Section 3. In Section 4, we show that the problem of finding fault critical vertices is **NP**-complete. The paper ends with the concluding remarks.

2 Preliminaries

In this section, we briefly describe some basic notions and results that have been presented in [1, 2].

Loosely speaking, AND/OR graph is a directed graph whose vertices are labeled either as AND or as OR. The graph has at least one input (source) vertex and exactly one output vertex. The AND vertices represent the components of the system that need all their inputs to produce their output. The OR vertices represent the components of the system that need only one of their "redundant" inputs to produce their output. A more formal definition is given below.

Definition 1 (AND/OR graph) *An* AND/OR graph $G(V_\wedge, V_\vee, \text{INPUT}, output; E)$ *is a graph with a set* $V_\wedge$ *of* AND-vertices*, a set* $V_\vee$ *of* OR-vertices*, a set* $\text{INPUT} \subset V_\wedge$ *of* input vertices*, an* output vertex $output \in V_\vee$*, and a set of* directed edges E*. The input vertices have no incoming edges and the output vertex has no outgoing edges.*

When an AND/OR graph is used to model a production system or a computation system, then there is a flow (e.g., information flow, traffic, production rate, etc.) associated with the edges of the graph. Given an AND/OR graph $G(V_\wedge, V_\vee, \text{INPUT}, output; E)$, a *capacity function* c associated with G is a positive integer function defined on edges of G. A *flow* f in G is an integer function defined on the edges of G that satisfies the following properties:
for all $e \in E$,

$$0 \leq f(e) \leq c(e),$$

for all $v \in V_\vee \backslash \{output\}$,

$$\sum_{e \in v^-} f(e) = \sum_{e \in v^+} f(e),$$

and for all $v \in V_\wedge \backslash \text{INPUT}$,

$$\forall e_1, e_2 \in v^-(f(e_1) = f(e_2))$$

$$\forall e_1 \in v^- \forall e_2 \in v^+(f(e_2) \leq f(e_1))$$

where v^- is the set of incoming edges of v and v^+ is the set of outgoing edges of v.

The *amount of flow* into a vertex $v \in V_\wedge \bigcup V_\vee$, is defined to be the value $\sum_{e \in v^-} f(e)$. Given a flow f in the AND/OR graph G, the *total flow* $F_f(G)$ is the amount of flow into the output vertex *output*.

There are numerous applications of the theory of flows in AND/OR graphs including optimal design of computation systems with multiple inputs, and optimal design of a production manufacturing process, where optimization is in terms of maximizing the total flow. The maximum flow problem for AND/OR graphs is defined as follows.

MFAO (i.e., Maximum Flows for AND/OR Graphs).
Instance: An AND/OR graph $G(V_\wedge, V_\vee, \text{INPUT}, \textit{output}; E)$, a capacity function c associated with G, and a positive integer k.
Question: Does there exist a flow f in G such that the total flow $F_f(G)$ is at least k?

Although there are polynomial time algorithms for finding maximum flows in traditional graph theory, the equivalent problem for AND/OR graphs is **NP**-complete.

Theorem 1 *MFAO is* **NP***-complete.*

Theorem 2 *MFAO is* **NP***-complete for* $k = 1$.

Now, let us consider the following setting: A complex system is modeled using an AND/OR graph G. Let c be the capacity function associated with G. The adversary has the power to destroy k nodes of the system and he wants to select the k vertices so that the damage is maximal. That is the adversary wants to remove k vertices from G so that the total flow of the remaining graph is minimal. In order to state the problem of critical vertices more precisely, we need to define what a set of critical vertices is. Given an AND/OR graph G with a capacity function c and a set of vertices U different from the input and output vertices, the capacity function is defined as

$$c_U(e) = \begin{cases} 0 & \text{if } e \text{ is an outgoing edges of some vertex in } U, \\ c(e) & \text{otherwise} \end{cases}$$

Given a number $k > 0$, a *set of critical vertices with respect to* c *and* k is a vertex set U with the following properties:

- No element of U is an input or output vertex.
- $|U| \leq k$.
- If $F_{c_U}(G)$ is the maximum of all total flows, then for any other vertex set U' with $|U'| = k$, $F_{c_U}(G) \leq F_{c_{U'}}(G)$.

The critical vertices problem is defined as follows.

CV (i.e., Critical Vertices).

Instance: An AND/OR graph $G(V_\wedge, V_\vee, \text{INPUT}, output; E)$ with a capacity function c, and a vertex set U.

Question: Is U a set of critical vertices with respect to c and $|U|$?

The following results hold.

Theorem 3 *CV is* **NP***-hard.*

Theorem 4 *CV is co*$\mathbf{NP}$*-hard.*

Other variants of the CV problem (e.g., Critical Vertices with a given Bound and Strictly Critical Vertices) are shown to be **NP**-hard too.

3 Faulty flows

In the model described in the previous section, the adversary was only allowed to destroy vertices and his task was to maximize damage given the number of vertices that can be destroyed. As previously mentioned, there are numerous scenarios that don't fit in this model. Some of these scenarios can be analyzed using the notions presented in this section. Namely, we allow the malicious or non-malicious faults to be introduced in the components of the system so that a portion of the components' output flow becomes faulty.

Given an AND/OR graph $G(V_\wedge, V_\vee, \text{INPUT}, output; E)$, a flow f and a *set of faulty vertices* $W \subset (V_\wedge \bigcup V_\vee) \backslash (\text{INPUT} \bigcup \{output\})$, a *faulty flow* f_W with respect to f and W is an integral function defined on edges of G that satisfies the following properties:

for all $e \in E$,

$$0 \leq f_W(e) \leq f(e),$$

for all $v \in \text{INPUT}$

$$\forall e \in v^+ (f_W(e) = 0),$$

for all $v \in V_\vee \backslash (W \bigcup \{output\})$

$$\sum_{e \in v^-} f_W(e) = \sum_{e \in v^+} f_W(e),$$

for all $v \in V_\wedge \backslash (W \bigcup \text{INPUT})$

$$\forall e \in v^+ (\max\{0, f(e) - (f_v - m_v)\} \leq f_W(e) \leq \min\{M_v, f(e)\}),$$

where $f_v = f(e), e \in v^-$; $m_v = \max\{f_W(e) : e \in v^-\}$ and M_v is the ymin$\{\sum_{e \in v^-} f_W(e), f_v\}$.

For a vertex $v \in V_\wedge \bigcup V_\vee$, the *amount of faulty flow* into v is defined to be the value $\sum_{e \in v^-} f_W(e)$. For a faulty flow f_W in the AND/OR graph G, the *total faulty flow* $F_{f_W}(G)$ is defined to be the amount of flow into the output vertex *output*.

4 Fault critical vertices

We consider the following scenario: A redundant computation or production system is modeled as an AND/OR graph G and there is a flow f associated with G. The adversary has power to introduce faults in at most K components (vertices) of the system[1]. Then, the adversary wants to know how to choose the K vertices so that the damage to the system is largest, that is the faulty flow is maximized. This question is also interesting for designer of the system since he wants to know how robust his system is. The formal description of the problem is given below.

FCV (i.e., Fault Critical Vertices).
Instance: An AND/OR graph $G(V_\wedge, V_\vee, \text{INPUT}, \textit{output}; E)$, a flow f in G, and two positive integers K and B.
Question: Are there a set of faulty vertices $W \subset (V_\wedge \bigcup V_\vee) \backslash (\text{INPUT} \bigcup \{\textit{output}\})$ and faulty flow f_W with respect to f and W such that $|W| \leq K$ and the total faulty flow is $F_{f_W}(G) \geq B$.

As it was case in the previous models, it turns out that the problem of finding fault critical vertices is also hard.

Theorem 5 *FCV is* **NP***-complete.*

Proof. Given an AND/OR graph G and a flow f, we can guess a set of faulty vertices $W(|W| \leq K)$ and f_W and then verify in polynomial time whether f_W is a faulty flow with respect to f and W such that $F_{f_W}(G) \geq B$. Hence, FCV $\in$ **NP**.

Next, we will show how **NP**-complete problem VC (i.e., Vertex Cover) can be reduced to FCV. VC is defined as:
Instance: A graph $G = (V, E)$ and a positive integer $K \leq |V|$.
Question: Is there a vertex cover of size K or less, that is a subset $V' \subset V$ such that $|V'| \leq K$ and for each edge $\{u, v\} \in E$, at least one of u and v belongs to V'.

[1]The faults can also be non-malicious (not introduced by an adversary).

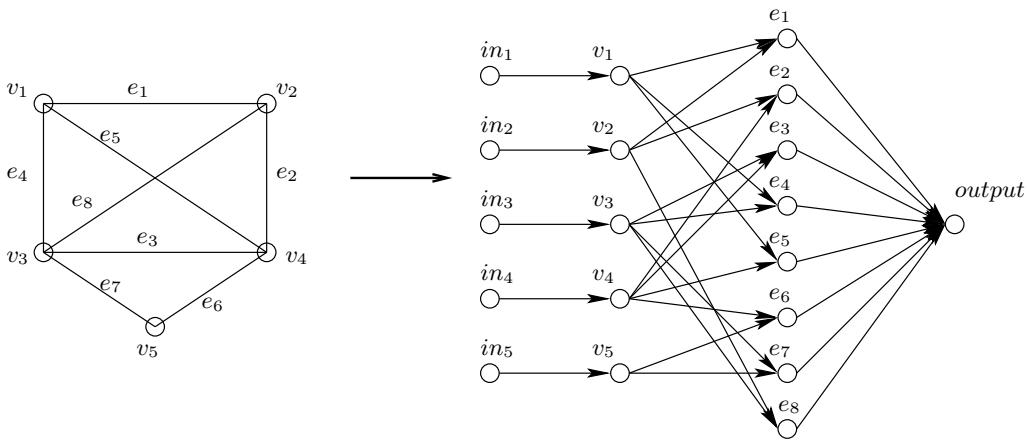

Figure 1: Reducing an instance of VC to an instance of FCV

Given a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_m\}$ and $E = \{e_1, \ldots, e_n\}$, and a positive integer $K \leq |V|$, we will construct an AND/OR graph G' and determine a flow f such that there is a vertex cover of size K or less if and only if there is a faulty set W of size K or less and a faulty flow f_W with respect to f and W such that the total faulty flow is $F_{f_W}(G) \geq n$. The set of input vertices is INPUT $= \{in_1, \ldots, in_m\}$. All vertices except *output* are AND vertices and $V_\wedge =$ INPUT $\bigcup \{v_1, \ldots, v_m\, e_1, \ldots, e_n\}$, that is to each vertex and edge in G corresponds a vertex in G'. Furthermore, the set of edges of G' consists of directed edges $(in_i, v_i), i = 1, \ldots, m$, directed edges $(e_j, output), j = 1, \ldots, n$, and directed edges (v_i, e_j), where (v_i, e_j) is an edge of G' iff e_j is incident upon v_i in G. The flow f is 1 for all edges of G'. It is obvious that G' and f can be constructed in time polynomial of the size of the corresponding VC instance. An example is given in Figure 1.

Suppose that V' is a vertex cover of size K or less. Let W be the set of vertices in G' that correspond to the vertices in V' and let $f_W(e) = 1$ if $e = (e_j, output), j = 1, \ldots, n$ or $e = (v_i, e_j)$ where $v_i \in V'$ and $f_W(e) = 0$ for the rest of the edges. It is not difficult to verify that f_W is faulty flow with respect to f and W, and $F_{f_W}(G) \geq n$. Now, suppose that W is a faulty set of size K or less and f_W is a faulty flow with respect to f and W such that $F_{f_W}(G) \geq n$. Let V' be a subset of $\{v_1, \ldots, v_m\}$ constructed in a following manner:

- If $v_i \in W$, then put v_i in V'.

- If $e_k \in W$ and $e_k = \{v_i, v_j\}$ in G, then put either v_i or v_j in V'.

Obviously, $|V'| \leq K$. We will show that, V' is vertex cover. Assume that V' is not a vertex cover, then there is vertex e_k in G' such that e_k, v_i and v_j do not belong to W, where $e_k = \{v_i, v_j\}$ in G. In that case, for any faulty flow with respect to f and W we have $f_W((e_k, output)) = 0$ and $F_{f_W}(G) \leq n-1$, which is in contradiction to the assumption that $F_{f_W}(G) \geq n$. Therefore, the set V' is a vertex cover of size K or less for the graph G. ■

5 Conclusion

AND/OR graphs and flows are often used in computer science to describe and analyze real life system. In the reliability and survivability theory, AND/OR graphs are used to describe redundant computation and production systems, and to analyze the vulnerability (or robustness) of the systems in presence of malicious or non-malicious faults. The adversary in the existing models based on AND/OR graphs is only allowed to destroy the components of the system. We consider the scenario where malicious or non-malicious faults are introduced into some of the components of the systems. We formally describe this scenario using the notion of faulty flows and fault critical vertices, and show that the problem of finding fault critical vertices is **NP**-complete.

References

[1] Y.Desmedt and Y.Wang, "Maximum Flows and Critical Vertices in AND/OR Graphs," COCOON '02 2002, Lecture Notes in Computer Science 2387, pages 238-248, Springer-Verlag, 2002.

[2] Y.Desmedt and Y.Wang, "Analysing Vulnerabilities of Critical Infrastructures Using Flows and Critical Vertices in AND/OR Graphs," International Journal of Foundations of Computer Science, 15(1), pp. 107-125, February 2004.

[3] M. Burmester, Y. Desmedt and Y. Wang,"Using approximation hardness to achieve dependable computation," In: Proc. the 2nd International Conference on Randomization and Approximation Techniques in Computer Science, Springer Verlag, 1998.

[4] Y.Wang, Y.Desmedt and M.Burmester, "Models for dependable computation with multiple inputs and some hardness results," Fundamenta Informaticae, 42(1):61–73, 2000.

[5] A. Bagchi and A. Mahanti, "Admissible heuristic search in AND/OR graphs," Theoret. Comput. Sci., 24:207

[6] P. Chakrabarti, S. Ghose and S. DeSarkar, "Admissibility of AO* when heuristics overestimate," Artificial Intelligence, 34:97-113,1988.

[7] C. L. Chang and J. R. Slagle, "An admissible and optimal algorithm for searching AND/OR graphs," Artificial Intelligence, 2:117-128, 1971.

[8] S. Even, "Graph Algorithms," Computer Science Press, 1979.

[9] L.R. Ford and D. R. Fulkerson, "Flows in Networks," Princeton University Press, Princeton, NJ, 1962.

[10] M. Gondran and M. Minoux, "Graphs and Algorithms," John Wiley and Sons Ltd., New York, 1984.