

Research Article

Qualitative Analysis of Commercial Services in MEC as Phased-Mission Systems

Yuhuan Gong ¹ and Yuchang Mo ²

¹School of Marxism, Huaqiao University, Quanzhou, China

²Fujian Province University Key Laboratory of Computational Science, School of Mathematical Sciences, Huaqiao University, Quanzhou, China

Correspondence should be addressed to Yuhuan Gong; myc@hqu.edu.cn

Received 5 August 2020; Revised 10 September 2020; Accepted 23 September 2020; Published 12 October 2020

Academic Editor: Xiaolong Xu

Copyright © 2020 Yuhuan Gong and Yuchang Mo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, mobile edge computing (MEC) is one of the most popular techniques used to respond to real-time services from a wide range of mobile terminals. Compared with single-phase systems, commercial services in MEC can be modeled as phased-mission systems (PMS) and are much more complex, because of the dependencies across the phases. Over the past decade, researchers have proposed a set of new algorithms based on BDD for fault tree analysis of a wide range of PMS with various mission requirements and failure behaviors. The analysis to be performed on a fault tree can be either qualitative or quantitative. For the quantitative fault tree analysis of PMS by means of BDD, much work has been conducted. However, for the qualitative fault tree analysis of PMS by means of BDD, no much related work can be found. In this paper, we have presented some efficient methods to calculate the MCS encoding by a PMS BDD. Firstly, three kinds of redundancy relations-inclusive relation, internal-implication relation, and external-implication relation-within the cut set are identified, which prevent the cut set from being minimal cut set. Then, three BDD operations, IncRed, InImpRed, and ExImpRed, are developed, respectively, for the elimination of these redundancy relations. Using some proper combinations of these operations, MCS can be calculated correctly. As an illustration, some experimental results on a benchmark MEC system are given.

1. Introduction

Currently, mobile edge computing (MEC) is one of the most popular techniques used to respond to real-time services from a wide range of mobile terminals [1–4]. Edge computing is a distributed computing topology where information processing is placed closer to the things or people that produce and/or consume that information. It means that MEC provides cloud-computing capabilities at the edge of the mobile network in close proximity to mobile subscribers; it is also considered as one of the key pillars for meeting the demanding KPIs of 5G.

Mobile edge computing integrates cloud computing (CC) into mobile networks, prolonging the battery life of mobile users (MUs). However, this mode may cause significant mission complexity. Compared with single-phase

systems, commercial services in MEC can be modeled as phased-mission systems (PMS). A MEC PMS is defined as a commercial service, which is subject to multiple, consecutive, nonoverlapping phases of operation. During each phase, it has to accomplish a specified task. Thus, the MEC configuration, failure criterion, and/or failure behavior can change from phase to phase [5–8].

Much early work has been conducted on the fault tree analysis of PMS. Qualitative analysis of commercial services in MEC as phased-mission systems is much more complex, because of the dependencies across the phases. For instance, the state of a MEC node at the beginning of a new phase is identical to its state at the end of the previous phase [9–11]. Over the past decade, researchers have proposed a set of new algorithms based on the binary decision diagram (BDD) for fault tree analysis of a wide range of PMS with various

mission requirements and failure behaviors [12–15]. Due to the nature of the BDD, cancellation of common components among the phases can be combined with the BDD generation, without additional operations, and the sum of disjoint products (SDP) can be implicitly represented by the final PMS BDD. Several experiments show that BDD-based algorithm is more efficient than the algorithm based on SDP, in both computation time and storage space; this efficiency allows the study of some practical, large MEC PMS [16–18].

The analysis to be performed on a fault tree can be either qualitative or quantitative. Qualitative analysis involves calculating the minimal cut set (MCS), that is, listing all possible smallest combinations of basic events, which cause the top event. Quantitative analysis, on the other hand, involves calculating the probability of the top event occurring from the probabilities of the basic events. For the qualitative fault tree analysis of MEC PMS by means of BDD, no much related work can be found. In this paper, we focus on this line of research, that is, calculating the MCS encoding by a PMS BDD. The major contributions of our work are the following ones:

- (1) Different kinds of redundancy relations within the cut set encoding by a PMS BDD are identified, and it is these relations that prevent the cut set from being minimal cut set.
- (2) Two BDD operations are proposed to eliminate these redundancy relations from a PMS BDD. One operation can only eliminate the external-implication relations relating to one component, and the other operation eliminates all the inclusive relations from the cut set.
- (3) MCS calculation method is developed by combining the proposed two BDD operations for a PMS BDD with forward ordering.

The remainder of the paper is organized as follows. Section 2 introduces the fundamentals of fault tree and BDD for the qualitative analysis of MEC PMS. Section 3 proposes three kinds of redundancy relations within a PMS BDD, several BDD operations eliminating these redundancy relations, and MCS calculation methods based on these BDD operations. Section 4 gives some experimental results on a benchmark MEC PMS. Last, conclusions are given in Section 5.

2. Preliminaries

2.1. Fault Tree. Fault tree analysis is an important technique for reliability and safety analysis. Bell Telephone Laboratories developed the concept in 1962 for the US Air Force for use with the Minuteman system. It was later adopted and extensively applied by the Boeing Company.

Fault tree diagrams are logic block diagrams that display the state of a system (top event) in terms of the states of its components (basic events). It uses a graphic “model” of the pathways within a system that can lead to a foreseeable, undesirable loss event (or a failure). Fault trees are built using gates and events. The two most commonly used gates

in a fault tree are the AND and OR gates. If both events need to occur to cause the top event to occur, they are connected by an AND gate. Alternatively, if the occurrence of either event causes the top event to occur, then these events are connected using an OR gate. Notice that NOT gate is not considered in this paper. Thus, the fault trees analyzed in the following sections are coherent fault trees.

For MEC PMS, there are two kinds of fault tree models: phase-level fault tree, which represents the causal chain between component failure and phase failure, and system-level fault tree, which represents the causal chain between phase failure and service failure. As an illustration example, Figure 1 depicts a small MEC PMS fault tree.

For the purpose of this paper, MEC PMS fault trees are essentially considered as Boolean functions, that is, terms inductively built over the two constants 0 and 1, a set of variables X , and usual logical connectives \wedge (AND) \vee (OR). The Boolean function F associated with the fault tree in Figure 1 is $F = (A_1 \vee B_1 \vee C_1) \vee (A_2 \vee (B_2 \wedge C_2)) \vee (A_3 \wedge B_3 \wedge C_3)$.

2.2. BDD. BDD is a compact encoding of the truth tables of Boolean function [19]. The BDD representation is based on the Shannon decomposition. Let F be a Boolean function that depends on the variable x ; then the following equality holds:

$$\begin{aligned} F &= (x \wedge F[x \leftarrow 1]) \vee (x \wedge F[x \leftarrow 0]) \\ &= \text{ite}(x, F[x \leftarrow 1], F[x \leftarrow 0]). \end{aligned} \quad (1)$$

BDD has two sink nodes, labeled 0 and 1, representing the two corresponding constants 0 and 1. Each nonsink node is labeled with a Boolean variable x and has two outgoing edges that represent the two corresponding expressions in the Shannon decomposition. These two edges are called E-edge (or 0-edge) and T-edge (or 1-edge), respectively. The node linked by the T-edge represents the Boolean expression $F[x \leftarrow 1]$; E-edge represents the Boolean expression $F[x \leftarrow 0]$. Thus, each nonsink node in a BDD encodes an ite format.

An ordered BDD is a BDD with the constraint that the variables are ordered and every source-to-sink path in the ordered BDD visits the variables in ascending order. A reduced ordered BDD is an ordered BDD where each node represents a distinct Boolean expression.

2.3. Variable Ordering. A variable in a PMS BDD indicates the component that the variable belongs to and the phase in which the component exists. For example, C_i is the state indicator variable of component C at phase i . Variable ordering can be generated in the following steps: component-level ordering and ordering variables on phases.

- (1) Component-level ordering: the components in PMS are s -independent from each other. Therefore, the ordering heuristics for ordinary BDD can be applied to component-level ordering for PMS BDD.
- (2) Phase ordering: there are two classes of phase ordering: forward ordering and backward ordering. In forward ordering, the variable order is the same as

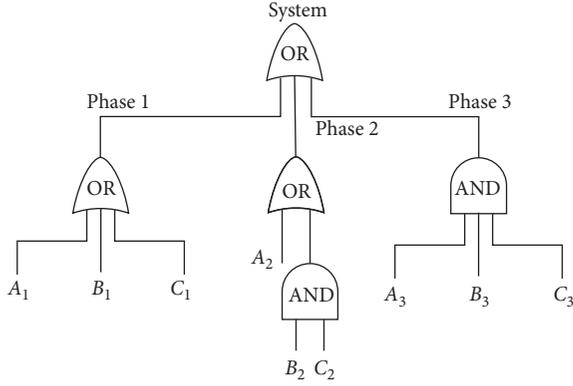


FIGURE 1: An example MEC PMS fault tree.

the phase order. In backward ordering, the variable order is the reverse of the phase order.

To illustrate the above ordering method, the PMS in Figure 1 can be used. Applying DFLM to the fault tree in “do not care” form shown in Figure 2, where phase indexes are unconsidered, the component-level ordering is $A < B < C$. Extending this component-level ordering over different phase indexes, the forward DFLM ordering is $A_1 < A_2 < A_3 < B_1 < B_2 < B_3 < C_1 < C_2 < C_3$, and the backward DFLM ordering is $A_3 < A_2 < A_1 < B_3 < B_2 < B_1 < C_3 < C_2 < C_1$.

2.4. BDD Generation. In order to compute the BDD associated with a Boolean function F , the following principle is applied:

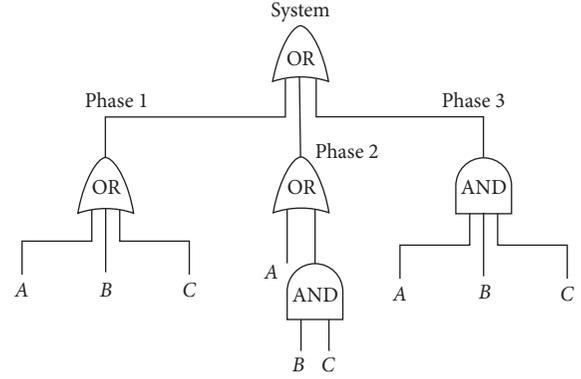


FIGURE 2: The “do not care” form of the tree in Figure 1.

- (1) If F is a constant, then one associates with F the corresponding sink node 0 or 1.
- (2) If $F = x$, where x is a variable, then one associates with F the BDD $\text{ite}(x, 1, 0)$.
- (3) Finally, if $F = G \langle \rangle H$, where $\langle \rangle$ is the binary connective \wedge or \vee and G and H are functions, then one computes the BDDs associated with G and H and then performs the operation $\langle \rangle$ on these two BDDs.

Given two Boolean functions G and H encoded by the BDDs $G = \text{ite}(x, G_1, G_0)$ and $H = \text{ite}(y, H_1, H_0)$, it is possible to compute directly on G and H any logical operation between G and H by means of the following calculation:

$$G \diamond H = \text{ite}(x, G_1, G_0) \diamond \text{ite}(y, H_1, H_0) = \begin{cases} \text{ite}(x, G_1 \diamond H_1, G_0 \diamond H_0), & x = y, \\ \text{ite}(x, G_1 \diamond H, G_0 \diamond H), & x < y, \\ \text{ite}(y, G \diamond H_1, G \diamond H_0), & x > y. \end{cases} \quad (2)$$

Compared with the single-phase system, BDD generation of PMS is much more complex because of the dependencies of component states across the phases. Use G and H to represent phase BDD of phase i and phase j , respectively ($i < j$). When expanded with regard to x_i and x_j ,

they, respectively, can be written as $G = \text{ite}(x_i, G_1, G_0)$; $H = \text{ite}(x_j, H_1, H_0)$.

Following special BDD operation, to deal with the cross-phase dependencies associated with the operation $\langle \rangle$ on G and H ,

$$G \diamond H = \text{ite}(x_i, G_1, G_0) \diamond \text{ite}(x_j, H_1, H_0) = \begin{cases} \text{ite}(x_i, G_1 \diamond H_1, G_0 \diamond H_0), & x_i = x_j, \\ \text{ite}(x_j, G \diamond H_1, G_0 \diamond H_0), & x_j < x_i, \\ \text{ite}(x_i, G_1 \diamond H_1, G_0 \diamond H), & x_i < x_j. \end{cases} \quad (3)$$

With the help of BDD operations (2) and (3), PMS BDD can be generated from PMS fault tree. When different orderings are used, the sizes of PMS BDD are different. As shown in Figure 3, the size of BDD with forward DFLM ordering is 15.

3. Qualitative Analysis

3.1. Cut Set and Minimal Cut Set. In order to introduce formally the notion of minimal cut set, we need the following definitions.

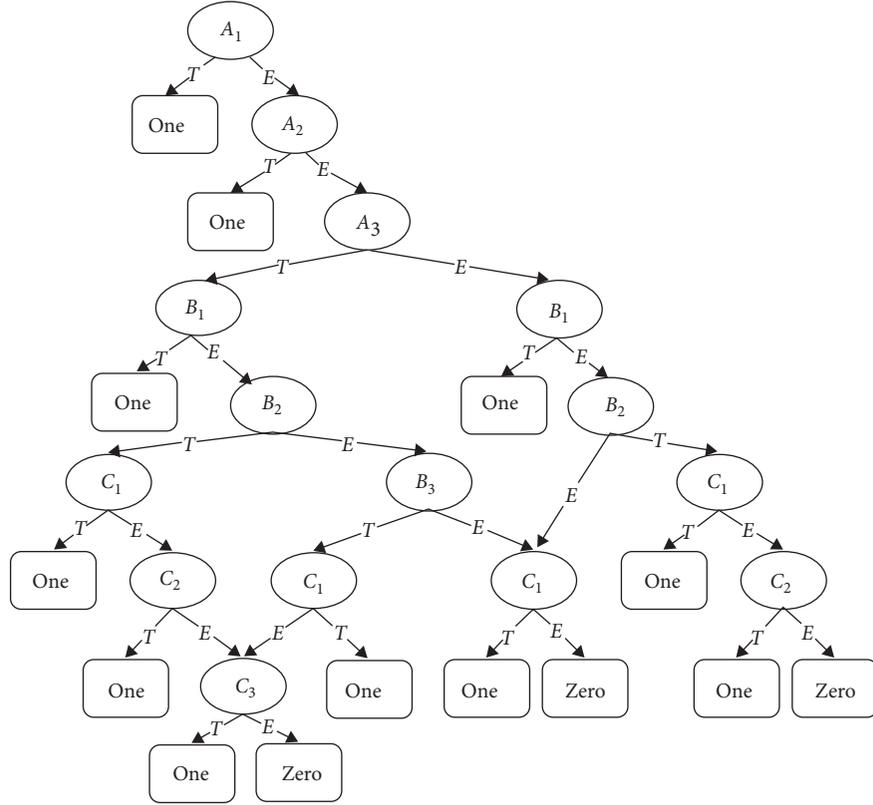


FIGURE 3: PMS BDD with forward DFLM ordering.

Definition 1. Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of Boolean variables. An assignment of X is a mapping from X into $\{0, 1\}$.

Assignment and subset of X are equivalent objects. A unique assignment σ corresponds to a subset of X , $X(\sigma)$, where

$$\sigma(x) = 1 \text{ iff } x \in X(\sigma). \quad (4)$$

Definition 2. Let σ be an assignment of X and F be a monotonic Boolean function on X . $X(\sigma)$ is a cut if $F(\sigma) = 1$.

The set of cuts of F is denoted by $CS[F]$.

Definition 3. For any $\rho \in CS[F]$, ρ is a minimal cut if for any assignment σ ,

$$X(\sigma') \subset \rho \text{ implies that } F(\sigma') = 0. \quad (5)$$

The set of minimal cuts of F is denoted by $MCS[F]$.

According to Definition 3, the following property holds.

Property 1. Let $MCS[F]$ be the minimal cut set of F ; then

$$MCS[F] \subseteq CS[F] \wedge \forall \rho \in CS[F] - MCS[F], \exists \delta \in MCS[F], \delta \subset \rho. \quad (6)$$

Here, set $CS[F] - MCS[F]$ includes all the elements in $CS[F]$, but not in $MCS[F]$.

Let F be the BDD associated with F . Each path from the root of F to sink node 1 corresponds to an assignment σ and defines a cut of F , ρ , as follows: $x \in \rho$ iff the path goes through a node labeled by x and goes out of this node on the T-edge.

The set of cuts encoded by F is denoted by $CS[F]$. Due to the fact that F compactly encodes the truth table of F in Shannon's form by means of subtree sharing, $CS[F]$ is a cut set of F .

What is more, PMS BDD F is minimal if for any path from the root of F to sink node 1, the cut identified by it is minimal. The set of cuts encoded by minimal F , denoted by $MCS[F]$, is still a minimal cut set of F .

Consider the PMS BDD with forward DFLM ordering shown in Figure 3. The cut set encoded by this BDD is

$$\{\{A_1\}, \{A_2\}, \{A_3, B_1\}, \{A_3, B_2, C_1\}, \{A_3, B_3, C_2\}, \{A_3, B_2, C_3\}, \{A_3, B_3, C_1\}, \{A_3, B_3, C_3\}, \{A_3, C_1\}, \{B_1\}, \{B_2, C_1\}, \{B_2, C_2\}, \{C_1\}\}. \quad (7)$$

According to forward ordering, the following property holds.

Property 2. Let F be a PMS BDD with forward ordering. Then,

$$\forall C \in \text{CS}[F], \quad |C \cap X| = 1, \quad (8)$$

where $X = \{x_1, x_2, \dots, x_n\}$ includes all variables relating to component X and n is the maximum of phase number.

Proof. Due to the fact that PMS considered in this paper is not maintained, a component remains down for the remaining phases if it is down in one phase; that is, x_j will never appear at the T-edge of node x_i .

In the following section, we try to show how to calculate the MCS[F] from PMS BDD F or CS[F]. \square

3.2. Inclusive Relation Elimination. Let $F = \text{ite}(x, F_1, F_0)$ be a BDD. If ω is a cut of F_0 , then ω does not contain x and ω is also a cut of F . If δ is a cut of F_1 , δ augmented of x , $\rho = \delta U \{x\}$, is a cut of F . If ρ includes ω , we say that F has inclusive relation between ω and ρ . Notice that ω will never include ρ . More formally, we have the following definition.

Definition 4. Let F be a PMS BDD; F has inclusive relation across different cuts within CS[F] if $\exists \rho_1, \rho_2 \in \text{CS}[F], \rho_1 \subset \rho_2$.

From the point of view of MCS, inclusive relation is one kind of redundancy relation. That is to say, in order to make CS[F] be MCS, ρ_{11} should be dropped.

It is obvious that the cut sets encoded by PMS BDD shown in Figures 4 and 5 have inclusive relations across different cuts. For example, $\{A_3, C_1\}$ is a cut, but its subset $\{C_1\}$ is still a cut.

The following theorem gives an inductive principle to eliminate the inclusive relations across cuts.

Theorem 1. Let $F = \text{ite}(x, F_1, F_0)$ be a PMS BDD. Then,

$$\begin{aligned} \text{IncRed}(F) = & \{ \rho \mid (\rho = \delta \cup \{x\}) \wedge (\delta \in \text{IncRed}(F_1) \setminus \text{CS}[F_0]) \} \\ & \cup \text{IncRed}(F_0), \end{aligned} \quad (9)$$

where $\text{IncRed}(F)$ is the cut set of F which has no inclusive relation across different cuts.

Proof

$$\text{CS}[F] = \{ \rho \mid (\rho = \delta \cup \{x\}) \wedge (\delta \in \text{CS}[F_1]) \} \cup \text{CS}[F_0]. \quad (10)$$

For CS[F₀], by eliminating the inclusive relations within it, we obtain IncRed(F₀).

For CS[F₁], by eliminating the inclusive relations within it, we obtain IncRed(F₁).

In order to eliminate the inclusive relations between $\{ \rho \mid (\rho = \delta \cup \{x\}) \wedge (\delta \in \text{IncRed}[F_1]) \}$ and CS[F₀], IncRed(F₁) \setminus CS[F₀] should be calculated. The without

operation “\” is defined in [9] and can remove from IncRed(F₁) all those cuts that include a cut of CS[F₀]. For example, if $\{A, B\}, \{B, C\} \in \text{IncRed}(F_1)$ and $\{A\} \in \text{CS}[F_0]$, then $\{A, B\} \notin \text{IncRed}(F_1) \setminus \text{CS}[F_0]$.

According to Theorem 1, the following property holds. \square

Property 3. Let F be a PMS BDD. Then,

$$\begin{aligned} & (\text{IncRed}(F) \setminus \text{CS}[F]) \wedge (\forall \rho \in \text{IncRed}(F) - \text{CS}[F], \\ & \quad \exists \delta \in \text{IncRed}(F), \delta \subset \rho). \end{aligned} \quad (11)$$

Using Theorem 1, an operation (IncRed) can be derived to eliminate all the inclusive relations within CS[F]. Reference [20] has implemented a similar operation called “Minsol” under the context of fault tree analysis of single-phase system. Thus, the operation implementation would not be given here.

As an illustration, IncRed operation is used to transform BDD in Figure 3 into the one as shown in Figure 4. The cut set encoded by the PMS BDD shown in Figure 4 is $\{\{A_1\}, \{A_2\}, \{A_3, B_2, C_3\}, \{A_3, B_3, C_3\}, \{B_1\}, \{B_2, C_2\}, \{C_1\}\}$.

3.3. External-Implication Relation Elimination. Let $F = \text{ite}(x_i, F_1, F_0 = \text{ite}(x_j, G_1, G_2)), i < j$, be a PMS BDD with forward ordering. If ρ_1 is a cut of G_1 , ρ_1 contains x_j and $\rho_1 U \{x_i\}$ is also a cut of F . If ρ_2 is a cut of F_1 , $\rho_2 U \{x_i\}$ is a cut of F . If ρ_1 includes ρ_2 , we say that F has external-implication relation between cut $\rho_1 U \{x_j\}$ and cut $\rho_2 U \{x_i\}$. More formally, we have the following definition.

Definition 5. Let F be a PMS BDD with forward ordering. F has external-implication relation across different cuts within CS[F] if $\exists \rho_1, \rho_2 \in \text{CS}[F]$:

$$\begin{aligned} & (\rho_1 \cap \{x_i, x_j\} = \{x_i\}) \wedge (\rho_2 \cap \{x_i, x_j\} = \{x_j\}) \\ & \quad \wedge (\rho_1^{\subset} = \rho_2^{\subset}) \wedge (\rho_1^{\supset} \supset \rho_2^{\supset}), \end{aligned} \quad (12)$$

where $\rho^{\subset} = \{y \mid (y \in \rho) \wedge (\text{order}(y) < \text{order}(x_i))\}$. Notice that ρ_1, ρ_2 will never contain variables within $X - \{x_i, x_j\}$ according to Property 2.

From the point of view of MCS, external-implication relation is one kind of redundancy relation. Notice that x_i implies x_j , and $\{x_i, x_j\}$ and $\{x_i\}$ are equivalent objects. In order to make CS[F] be MCS, ρ_2 should be dropped. Consider the PMS BDD shown in Figure 5. The cut set encoded by this BDD is $\{\{A_1\}, \{A_2\}, \{A_3, B_2, C_3\}, \{A_3, B_3, C_3\}, \{B_1\}, \{B_2, C_2\}, \{C_1\}\}$. Due to the external-implication relation, $\{A_1\}$ should be eliminated from the cut set, and so does $\{A_2, B_2, C_3\}$.

The following theorem gives an inductive principle to eliminate the external-implication relation across different cuts.

Theorem 2. Let $F = \text{ite}(x_i, F_1, F_0 = \text{ite}(y_j, G_1, G_0)), i < j$, be a PMS BDD with forward ordering. Then,

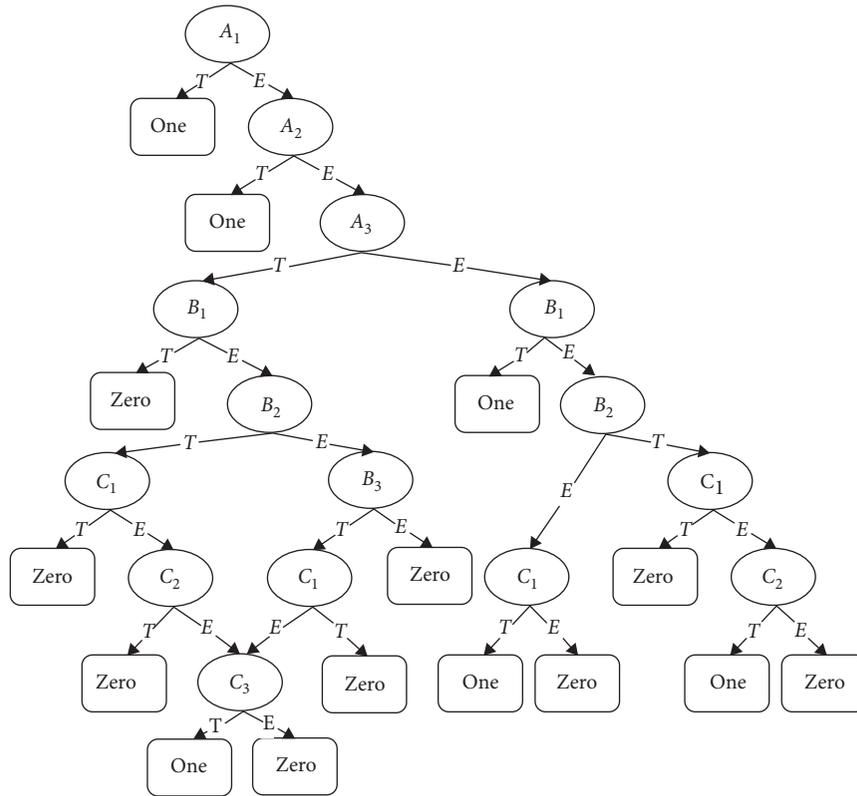


FIGURE 4: Results of IncRed operation for the PMS BDD in Figure 3.

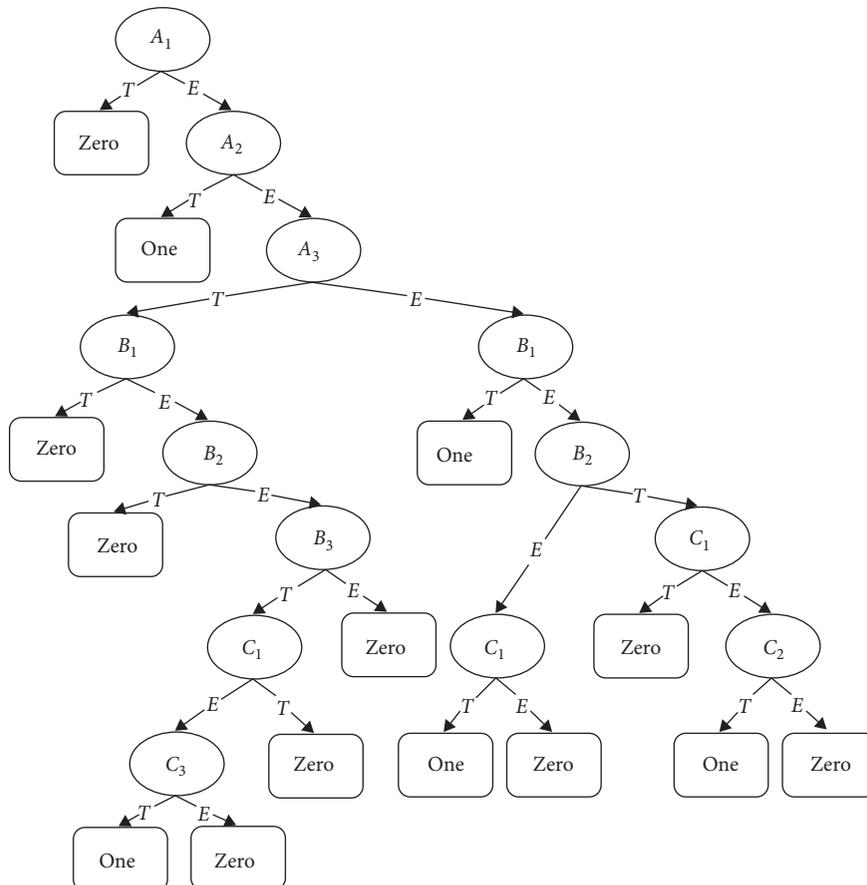


FIGURE 5: Results of ExImpRed operation for the PMS BDD in Figure 4.

$$\text{ExImpRed}(F) = \begin{cases} \{\rho \mid (\rho = \delta \cup \{x_i\}) \wedge \delta \in \text{ExImpRed}(F_1)\} \cup \text{ExImpRed}(F_0), & x \neq y, \\ \{\rho \mid (\rho = \delta \cup \{x_i\}) \wedge \delta \in \text{ExImpRed}(F_1) \setminus \text{CS}[G_1]\} \cup \text{ExImpRed}(F_0), & x = y, \end{cases} \quad (13)$$

where $\text{ExImpRed}(F)$ is the cut set of F which has no external-implication relation across different cuts.

Proof. Two cases are possible:

- (1) $x \neq y$. F_0 does not contain variables relating to component X .

If ρ is a cut of F , ρ is either a cut of F_0 (ρ does not contain x_i) or a cut of F_1 augmented of x_i (ρ only contains one variable x_i relating to component X). Thus, there is no external-implication relation relating to x_i within the cut set, and we obtain that

$$\{\rho \mid (\rho = \delta \cup \{x_i\}) \wedge \delta \in \text{ExImpRed}(F_1)\} \cup \text{ExImpRed}(F_0). \quad (14)$$

- (2) $x = y$. There might be some cuts that have external-implication relation relating to x_i .

If ρ is a cut of F_0 and ρ contains x_j , ρ is also a cut of F . If δ is a cut of F_1 , δ augmented of x_i is a cut of F . If $(\delta \supseteq \rho / \{x_j\})$, ρ and $\delta \cup \{x_i\}$ have external-implication relation. That is to say, δ should be eliminated. According to the fact that ρ contains x_j , $(\rho / \{x_j\})$ is a cut of G_1 . Thus, we obtain that

$$\begin{aligned} & \{\rho \mid (\rho = \delta \cup \{x_i\}) \wedge (\delta \in \text{ExImpRed}(F_1) \setminus \text{CS}[G_1])\} \\ & \cup \text{ExImpRed}(F_0). \end{aligned} \quad (15)$$

According to Theorem 2, the following property holds. \square

Property 4. Let F be a PMS BDD with forward ordering. Then,

$$\begin{aligned} & (\text{ExImpRed}(F) \subseteq \text{CS}[F]) \wedge (\forall \rho \in \text{ExImpRed}(F) - \text{CS}[F], \\ & \exists \delta \in \text{ExImpRed}(F), \delta \subset \text{Extend}(\rho)), \end{aligned} \quad (16)$$

where extend operation means adding variables with higher index to a cut. For example, $\text{extend}(\{A_2, B_2, C_3\}) = \{A_3, A_2, B_3, B_2, C_3\}$.

As an illustration, the ExImpRed operation is used to transform BDD in Figure 4 into the BDD shown in Figure 5. The cut set encoded by this BDD is $\{\{A_2\}, \{A_3, B_3, C_3\}, \{B_1\}, \{B_2, C_2\}, \{C_1\}\}$. This cut set has no external-implication relation across different cuts.

3.4. MCS Calculation

Theorem 3. Let F be a PMS BDD with forward ordering. Then,

$$\text{IncRed}(F) \cap \text{ExImpRed}(F) = \text{MCS}[F]. \quad (17)$$

Proof. We have known that $\text{CS}[F]$ might have inclusive relations and external-implication relations across the included cuts. According to Property 2, we know that x_i and x_j will not be simultaneously included by a cut. Thus, there is no internal-implication relation.

According to Theorem 2, we know that ExImpRed operation can only eliminate the external-implication relations relating to one component. However, $\text{CS}[F]$ might have external-implication relations relating to more than one component. For example, consider a $\text{CS}[F]$ has two cuts (one is $\{A_2, B_2\}$ and the other is $\{A_1, B_1\}$). Notice that x_i implies x_j and $\{A_1, B_1, A_2, B_2\}$ and $\{A_1, B_1\}$ are equivalent objects. Thus, $\{A_1, B_1\}$ should be dropped. However, our ExImpRed operation cannot use $\{A_2, B_2\}$ to eliminate $\{A_1, B_1\}$. Now, we show that the ExImpRed operation can do this elimination indirectly. If both $\{A_2, B_2\}$ and $\{A_1, B_1\}$ belong to $\text{CS}[F]$, then F must have a structure as shown in Figure 6. There are two points: (1) according to the BDD generation process, B_1 will appear at the T-edge of node A_2 under the condition that the T-edge of node A_1 has a B_1 and the T-subBDD of node A_2 has a B_2 ; (2) the T-edge of this B_1 is 1 due to the fact that $\{A_2, B_2\}$ is a cut. Thus, $\{A_2, B_1\}$ must be included by $\text{CS}[F]$. Now, the ExImpRed operation can use $\{A_2, B_1\}$ to eliminate $\{A_1, B_1\}$ and then use $\{A_2, B_2\}$ to eliminate $\{A_2, B_1\}$. Thus, our ExImpRed operation can eliminate all kinds of external-implication relations within $\text{CS}[F]$.

ExImpRed operation eliminates all the external-implication relations from the cut set $\text{CS}[F]$. However, it will disturb some inclusive relations. For example, consider the PMS BDD F shown in Figure 7, where cut set $\text{CS}[F] = \{\{A_1\}, \{A_2, B_1, C_1\}, \{A_2, B_2, C_1\}, \{B_1, C_1\}\}$. After the ExImpRed operation, we get $\text{ExImpRed}(F) = \{\{A_1\}, \{A_2, B_2, C_1\}, \{B_1, C_1\}\}$. Here, the inclusive relation between cut $\{B_1, C_1\}$ and cut $\{A_2, B_1, C_1\}$ is destroyed. Thus, after another IncRed operation, we get $\text{IncRed}(\text{ExImpRed}(F)) = \{\{A_1\}, \{A_2, B_2, C_1\}, \{B_1, C_1\}\}$ instead of $\{\{A_1\}, \{B_1, C_1\}\}$, the $\text{MCS}[F]$.

On the other hand, IncRed operation eliminates all the inclusive relations from the cut set $\text{CS}[F]$. However, it will disturb some external-implication relations. For the PMS BDD F shown in Figure 7, after the IncRed operation, we get $\text{IncRed}(F) = \{\{A_1\}, \{A_2, B_2, C_1\}, \{B_1, C_1\}\}$. Here, the external-implication relation between cut $\{A_2, B_1, C_1\}$ and cut $\{A_2, B_2, C_1\}$ is destroyed. Thus, after another ExImpRed operation, we get $\text{ExImpRed}(\text{IncRed}(F)) = \{\{A_1\}, \{A_2, B_2, C_1\}, \{B_1, C_1\}\}$ instead of $\{\{A_1\}, \{B_1, C_1\}\}$, the $\text{MCS}[F]$. \square

4. Application

With edge, compute and storage systems reside at the edge as well, as close as possible to the component, device,

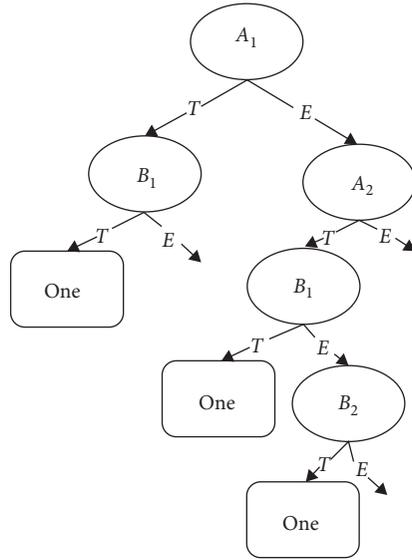


FIGURE 6: The PMS BDD encoding $\{A_2, B_2\}$ and $\{A_1, B_1\}$.

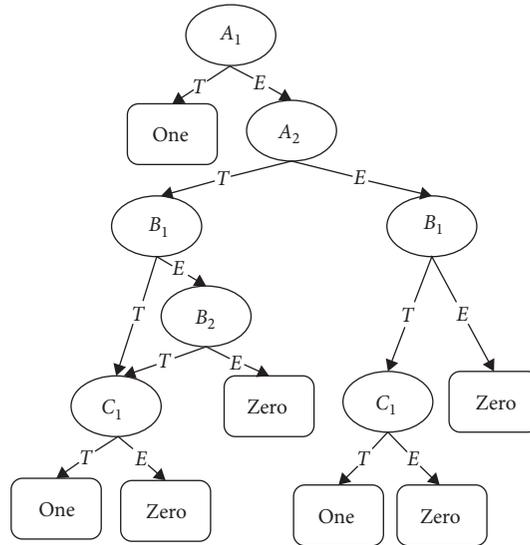


FIGURE 7: An example PMS BDD with forward ordering.

application, or human that produces the data being processed. The purpose is to remove processing latency, because the data need not be sent from the edge of the network to a central processing system and then back to the edge.

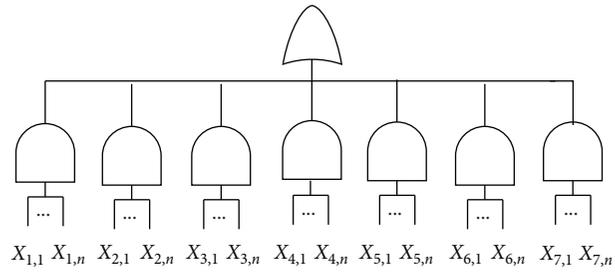
The applications for edge make sense: Internet of things-connected devices are a clear use for edge computing architecture. With remote sensors installed on a machine, component, or device, they generate massive amounts of data. If that data is sent back across a long network link to be analyzed, logged, and tracked, that takes much more time than if the data is processed at the edge, close to the source of the data.

There are patterns of failure. (1) If you place too much at the edge, it is easy to overwhelm the smaller processor and storage platforms that exist there. In some cases, storage could be limited to a few gigabytes and processing using a

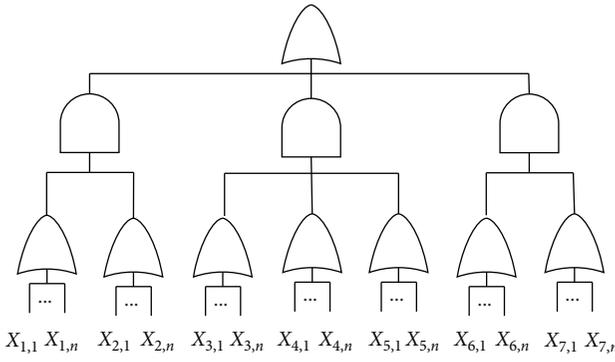
single CPU. Power and size restrictions are really what set the limits. (2) Another pattern is failure to integrate security from concept to production. Security is systemic to edge computing architectures and centralized processing. Security needs to span both and use mechanisms such as identity and access management. Encryption is not a nice-to-have, but rather a requirement for device safety.

Consider a benchmark MEC PMS borrowed from [21]. This MEC PMS has the following:

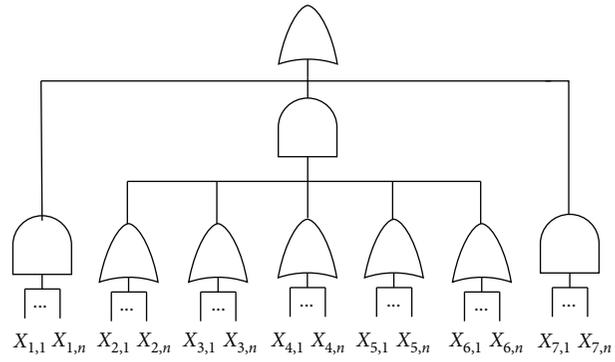
- (1) It has 7 groups of components.
- (2) Each group has 5 components.
- (3) Five MEC node configurations are shown in Figure 8.
- (4) Four MEC service configurations are listed in Table 1.



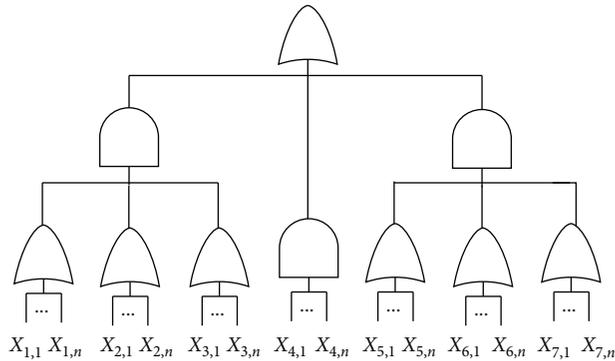
(a)



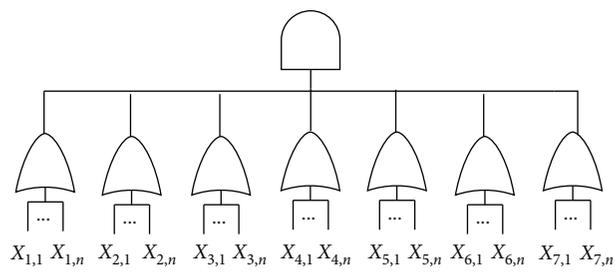
(b)



(c)



(d)



(e)

FIGURE 8: Five MEC node configurations.

TABLE 1: Four MEC service configurations.

Name	Phase number	Configuration
M1	2	A, E
M2	3	A, B, E
M3	4	A, B, C, E
M4	5	A, B, C, D, E

TABLE 2: Results of MCS calculation for PMS BDD with forward ordering.

Mission	CS[Mi]	ExImpRed (Mi)	IncRed (Mi)	IncRed (Mi) \cap ExImpRed (Mi)
M1	3399661	104492	279943	78132
M2	109574034	5802547	1944295	78307
M3	1616724670	60025676	8222890	81432
M4	1224665026	121374302	17953442	81682

Using Theorem 4, MCS is calculated with forward ordering. The results are shown in Table 2. Here, we only present the numbers of cuts in the set CS [Mi], IncRed (Mi), ExImpRed (Mi), and IncRed (Mi) \cap ExImpRed (Mi). From the results in Table 2, the following conclusions can be derived: (1) Different phase combination will result in different sizes of cut sets. Some more complex combination may have smaller cut sets as illustrated by M3 and M4. (2) For the MCS calculation IncRed \cap ExImpRed of this PMS, forward ordering can generate extremely smaller cut sets for the intermediate BDD operations compared with the original CS.

5. Conclusion

Based on the mobile edge computing techniques, commercial service providers, such as video content providers, can benefit from low-latency edge resources to provide their users with more efficient service acquisition, thereby improving the quality of experience [22–24]. Compared with single-phase systems, fault tree analysis of MEC PMS is much more complex, because of the dependencies across the phases.

The analysis to be performed on a fault tree can be either qualitative or quantitative. For the qualitative fault tree analysis of PMS by means of BDD, no much related work can be found. In this paper, we have presented some efficient methods to calculate the MCS encoding by a PMS BDD. The basic idea is to eliminate different kinds of redundancy relations from a cut set encoding by a PMS BDD. These methods are based on several BDD operations, such as IncRed and ExImpRed. The IncRed can eliminate all inclusive relations. The ExImpRed can eliminate all external-implication relations. Using some proper combinations of these operations, MCS can be calculated correctly.

Smart city is a fast-developing system enabled by the Internet of things (IoT) with massive collaborative services (e.g., intelligent transportation and collaborative diagnosis) [25, 26]. Therefore, one direction of our future work is to consider the qualitative analysis of commercial services in MEC smart city. As an extension of this research work, we will also improve the presented BDD-based analysis

methodology for more generalized MEC service systems [27].

Data Availability

The MCS calculation data used to support the findings of this study may be released upon application to the Fujian Province University Key Laboratory of Computational Science, who can be contacted at CSLab@hqu.edu.cn.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the NSF of China under Grant no. 61972156 and Program for Innovative Research Team in Science and Technology in Fujian Province University, Quanzhou High-Level Talents Support Plan under Grant no. 2017ZT012.

References

- [1] X. Zhou, W. Liang, K. I.-K. Wang, H. Wang, L. T. Yang, and Q. Jin, “Deep-Learning-enhanced human activity recognition for Internet of healthcare things,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.
- [2] X. Zhou, Y. Li, and W. Liang, “CNN-RNN based intelligent recommendation for online medical pre-diagnosis support,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.
- [3] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Bhuiyan, “Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.
- [4] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. R. Chang, “Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, 2020.
- [5] H. Yu and M. Tang, “Reliability assessment for systems suffering competing degradation and random shocks under fuzzy environment,” *Science Progress*, vol. 103, no. 1, Article ID 003685041988108, 2020.

- [6] K. Yan, J. Huang, W. Shen, and Z. Ji, "Unsupervised learning for fault detection and diagnosis of air handling units," *Energy and Buildings*, vol. 210, Article ID 109689, 2020.
- [7] K. Yan, W. Shen, Q. Jin, and H. Lu, "Emerging privacy issues and solutions in cyber-enabled sharing services: from multiple perspectives," *IEEE Access*, vol. 7, pp. 26031–26059, 2019.
- [8] Q. Zhai, L. Xing, R. Peng, and J. Yang, "Aggregated combinatorial reliability model for non-repairable parallel phased-mission systems," *Reliability Engineering & System Safety*, vol. 176, pp. 242–250, 2018.
- [9] R. Peng, Q. Zhai, L. Xing, and J. Yang, "Reliability analysis and optimal structure of series-parallel phased-mission systems subject to fault-level coverage," *IIE Transactions*, vol. 48, no. 8, pp. 736–746, 2016.
- [10] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, 2020.
- [11] Y. Liu, Y. Chen, and T. Jiang, "Dynamic selective maintenance optimization for multi-state systems over a finite horizon: a deep reinforcement learning approach," *European Journal of Operational Research*, vol. 283, no. 1, pp. 166–181, 2020.
- [12] A. Shrestha, L. Xing, and Y. Dai, "Reliability analysis of multistate phased-mission systems with unordered and ordered states," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 4, pp. 625–636, 2011.
- [13] Y. Mo, L. Xing, and S. V. Amari, "A multiple-valued decision diagram based method for efficient reliability analysis of non-repairable phased-mission systems," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 320–330, 2014.
- [14] Y. Mo, "A multiple-valued decision-diagram-based approach to solve dynamic fault trees," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 81–93, 2014.
- [15] A. Shrestha, L. Xing, and Y. Dai, "Decision diagram-based methods, and complexity analysis for multistate systems," *IEEE Transactions on Reliability*, vol. 59, no. 1, pp. 145–161, 2010.
- [16] Y. Mo, L. Xing, and J. B. Dugan, "Performability analysis of k-to-l-out-of-n computing systems using binary decision diagrams," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 126–137. in press, 2015.
- [17] X. Zhou, W. Liang, S. Huang, and M. Fu, "Social recommendation with large-scale group decision-making for cyber-enabled online service," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1073–1082, 2019.
- [18] X. Zhou, W. Liang, K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, 2018.
- [19] R. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, 1986.
- [20] A. Rauzy, "New algorithms for fault trees analysis," *Reliability Engineering & System Safety*, vol. 40, no. 3, pp. 203–211, 1993.
- [21] Z. Xu, Y. Mo, Y. Liu, and T. Jiang, "Reliability assessment of multi-state phased-mission systems by fusing observation data from multiple phases of operation," *Mechanical Systems and Signal Processing*, vol. 118, pp. 603–622, 2019.
- [22] C. Zhou, A. Li, A. Hou, Z. Zhang, Z. Zhang, and F. Wang, "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Systems with Applications*, 2020.
- [23] K. Yan, L. Liu, Y. Xiang, and Q. Jin, "Guest editorial: AI and machine learning solution cyber intelligence technologies: new methodologies and applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6626–6631, 2020.
- [24] X. Zhou, W. Liang, K. I. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE Transactions on Computational Social Systems*, 2020.
- [25] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. Khosravi, and L. G. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, 2020.
- [26] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.
- [27] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, Article ID 106196, 2020.