# 计算节点不可靠网络可靠度的一种 MDD 算法

# 王泓刚 董荣胜 钱俊彦

(桂林电子科技大学广西可信软件重点实验室 桂林 541004)

摘 要 节点或边不可靠网络的可靠度分析问题是 NP-hard 问题, 网络节点和边都不可靠的假设更接近现实。基于 网络节点和边二元状态的假设, 构建了节点和边不可靠网络的形式化模型, 给出了分析节点和边不可靠网络可靠度的 NEF\_MDD 算法。该算法将单个节点与其未访问邻接边划分为一个集合, 通过枚举节点和边的不同组合, 合并导致子 网同构的冗余状态, 获得简化后的状态向量和可靠度向量, 并用一个多值决策图变量来表述。通过使用自定义的 MDD 操作算子, 构建整个网络的 MDD, 遍历 MDD 节点, 计算网络的可靠度。与二元决策图方法相比, 该方法能够降低决策图层数和节点规模, 有助于节点和边不可靠网络的可靠度分析。

关键词 多值决策图,网络可靠度,不可靠节点,不可靠边

中图法分类号 TP393 文献标识码 A DOI 10.11896/j. issn. 1002-137X. 2016. 1.035

## Novel Reliability Analysis Algorithm Based on MDDs in Networks with Imperfect Nodes

WANG Hong-gang DONG Rong-sheng QIAN Jun-yan

(Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract The reliability of networks with imperfect nodes or edges is an NP-hard problem, and the assumption of networks with imperfect nodes and edges is closer to real life. A formal model of networks with binary state nodes and edges was constructed, and a novel network reliability analysis algorithm was proposed. Any node and its adjacent non-visited edges' combination states are enumerated to merge isomorphic sub-networks. Then, a MDD variable is used to represent the reduced state vector and corresponding probability vector. Finally, the MDD representing for the network is constructed by a custom operation. Experiment shows that the level and size of decision diagram generated by the proposed algorithm are less than the corresponding binary decision diagram.

Keywords Multi-valued decision diagram, Network reliability, Imperfect nodes, Imperfect edge

## 1 引言

网络可靠度分析是网络设计、验证和维护阶段的一个关键问题,广泛应用于无线传感器网络、随机流网络等领域[1-4]。 网络可靠度分析的相关研究通常基于节点可靠、边可靠、节点和边都不可靠这 3 种假设,这 3 种假设下的网络可靠度分析都是 NP-hard 问题<sup>[1]</sup>。现实世界中的随机网络节点和边都可能出现故障,因此,节点和边都不可靠的假设更贴近现实<sup>[5]</sup>。

在节点和边都不可靠的假设下,网络可靠度分析的精确方法以因子分解法、最小路径法和二元决策图(Binary Decision Diagram, BDD)法为代表[6]。文献[7,8]在删除链路边之后处理链路两端不可靠节点,用链路有效和边失效的条件概率公式替换节点可靠的因子分解公式,提出节点和边都不可靠的因子分解与网络缩减算法;但因子分解由于存在大量冗余计算,当网络规模增大时会变得低效[9]。文献[2,3]研究了约束条件下节点不可靠的随机流网络,通过找出满足约束条件的所有下界点集(极小状态),利用容斥原理对下界点集进

行不交化处理求解网络的可靠度。文献[2,3]需要找出网络的所有最小路径,这是一个 NP-hard 问题<sup>[2,10]</sup>。同时,容斥原理在扩展过程中会生成很多相互抵消的项,导致冗余计算<sup>[11]</sup>。文献[5,9]引入 BDD 计算节点和边不可靠网络的可靠度,BDD 由于具有隐式表示状态空间的特性,能够在一定程度上缓解状态空间组合爆炸的问题,提高了计算效率。

多值决策图(Multi-valued Decision Diagram, MDD)是BDD的一般化形式。由于 MDD 变量具有多值(多状态)特性,与等价的 BDD 结构比较, MDD 结构有更少的空间消耗[12,13]。文献[14]基于网络二元状态的假设,通过枚举节点和其未访问邻接边的不同组合,生成节点和其未访问邻接边的状态属性集合,定义多值变量构建网络的 MDD。在定义二元网络的多值变量和构建网络 MDD 的过程中,文献[14]对引入 MDD 计算二元状态网络的可靠度提供了可行依据。

本文在文献[5,14]的基础上,将 MDD 引人节点和边不可靠网络的可靠度分析问题。基于节点和边二元状态的假设,构建了节点和边不可靠网络的形式化模型,给出了分析节

到稿日期,2015-01-05 返修日期,2015-04-03 本文受国家自然科学基金(61363070),广西可信软件重点实验室资助课题(kx201401),广西高等学校高水平创新团队及卓越学者计划,桂林电子科技大学创新团队资助。

**王泓刚**(1987一),硕士生,主要研究方向为无线传感器网络、网络可靠性,E-mail: hgwang8771@hotmail.com;**董荣胜**(1965一),教授,CCF 高级会员,主要研究方向为无线传感器网络、网络可靠性、形式化技术、协议工程;**钱俊彦**(1973一),教授,CCF 会员,主要研究方向为软件工程、模型检验、程序验证。

点和边不可靠网络可靠度的 NEF\_MDD 算法。与因子分解 法和最小路径法相比,NEF\_MDD 算法构建的 MDD 具有隐 式表示状态空间的特征。与等价的 BDD 结构相比,NEF\_ MDD 算法构建的 MDD 层数等于节点数目,简化了决策图的 节点规模。实例分析及实验结果表明,NEF\_MDD 算法构建 的 MDD 层数和节点规模小于 BDD 算法构建的决策图。

本文第2节介绍节点和边不可靠网络的形式化模型、可靠度定义、节点和边的约束条件;第3节介绍多值决策图和控制决策图的操作算子;第4节给出本文的算法和伪代码描述,用实例分析验证算法的正确性;第5节给出与BDD算法对比的实验结果;最后对本文工作做出总结。

# 2 节点和边不可靠网络

#### 2.1 节点和边不可靠网络模型

图 1(a)给出一个有 3 个节点和 3 条边的网络图实例,节点和边无权值,并以给定概率失效。

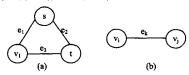


图 1 网络图实例

下面给出节点和边不可靠网络的形式化模型。

**定义 1** 节点和边不可靠网络的形式化模型 NEFN (Node-and-edge fault network)用六元组(*G*, *M*, *P*, *H*, *U*, *W*) 表示,其中:

(1)G=(V,E)表示网络图; $V=\{v_1,\cdots,v_i,\cdots,v_m\}$ 表示网络的节点集,其中  $i\in\{1,2,\cdots,m\}$ ;V 中唯一存在源点 s 和终端节点 t; $E=\{e_1,\cdots,e_j,\cdots,e_n\}$ 表示网络的边集,其中  $j\in\{1,2,\cdots,n\}$ 

 $(2)M = \{M_{11}, \dots, M_{1m}, M_{21}, \dots, M_{2n}\}$ 表示节点和边的状态向量,其中, $M_{1i} = \{0,1\}$ 表示节点  $v_i$  的状态集合,0 表示节点失效,1 表示节点正常工作; $M_{2j} = \{0,1\}$ 表示边  $e_j$  的状态集合,0 表示边失效,1 表示边正常工作。

 $(3)P = \{P_{11}, \dots, P_{1m}, P_{21}, \dots, P_{2n}\}$ 表示节点和边的可靠度向量,与状态向量 M 对应。其中, $P_{ij} = \{p_{ij0}, p_{ij1}\}, i \in \{1, 2\}, j \in \{1, \dots, MAX(m,n)\}, P_{ijk} (k \in \{0,1\})$ 表示节点  $v_{ij}$  或边  $e_{ij}$  对应状态 k 的概率,且  $P_{ij0} + P_{ij1} = 1$ 。

 $(4)H=\{H_1,H_2,\cdots,H_m\}$ 。 $H_i=\{v_i,ve_{i1},\cdots,ve_{ix}\}$ 表示节点与其未访问邻接边的集合, $ve_{ij}$ 表示  $H_i$  内与  $v_i$  相连且未被访问的第 j 条邻接边。任意两个集合满足条件  $H_i\cap H_j=\emptyset(i\neq j)$ ,所有集合满足条件  $H_i\cap H_i=V\cup E$ 。

 $(5)U = \{U_1, U_2, \cdots, U_m\}$  是对应 H 的状态向量, $U_i = \{f_{io}, f_{i1}, \cdots, f_{iy}\}$ ,y 的取值满足  $y \leq 2^{|H|} - 1$ 。  $f_{ij}$  ( $j \in \{0, \cdots, y\}$ )表示  $H_i$  处于第 j 个状态。  $f_{ij}$  取值为 0 表示处于第 j 个状态的  $H_i$  中不存在一条从 s 到 t 的连通路径, $f_{ij}$  取值为 1 表示处于第 j 个状态的  $H_i$  中存在一条从 s 到 t 的连通路径, $f_{ij}$  取值为其他表示无法判定当前状态下  $H_i$  中是否存在连通路径

 $(6)W = \{W_1, W_2, \cdots, W_m\}$ 是对应 H 的可靠度向量, $W_i = \{p_{i0}, p_{i1}, \cdots, p_{iy}\}$ ,y 的取值满足  $y \leqslant 2^{|Hi|} - 1$ 。  $P_{ij} (0 \leqslant j \leqslant y)$ 表示  $H_i$  处于状态 j 的概率,且  $\sum_{i=1}^{N} p_{ij} = 1$ 。

在 NEFN 模型中,对于一个给定的网络 G、状态向量 M

和可靠度向量P,则可以将网络G划分为若干个节点和边的集合H,并由此计算出H对应的状态向量U和可靠度向量W。具体方法在第4节介绍。

# 2.2 节点和边不可靠网络可靠度

网络可靠度的定义基于如下假设:

- (1)网络的节点和边都为二元状态,任意节点和边的状态 随机且独立分布;
  - (2)所有节点和边正常工作的概率为 0.9,失效概率为 0.1;
  - (3) 网络有唯一源点 s 和唯一终端节点 t。

定义 2 给定网络模型 NEFN,节点和边不可靠网络的可靠度是 s 和 t 之间至少存在一条连通路径的概率 [5] 。用 h 表示一条连通路径,函数 H(x,y)表示元素 x 在 y 内,则节点和边不可靠网络的可靠度  $NEF_REL$  可形式化描述为  $NEF_REL = Pr\{\exists h((\forall v_i)(H(v_i,h) \land (M_{1i}=1)) \land (\forall e_i)(H(e_i,h) \land (M_{2i}=1)))\}$ 。

## 2.3 节点和边的约束条件

给定如图 1(b)所示链路模型,节点  $v_i$  和  $v_j$  以及边  $e_k$  满足约束条件:如果  $v_i$  或  $v_j$  失效,则边  $e_k$  失效。结合 NEFN 模型,把该约束条件形式化描述为式(1):

$$((M_{1i}=0) \lor (M_{1j}=0)) \rightarrow (M_{2k}=0)$$
 (1)

# 3 多值决策图

#### 3.1 多值决策图

定义 3 多值决策图可以用一个具有多个终端节点的有向无环图来表示,表示具有 n 个变量的离散函数,形式化描述为式(2):

$$f: U_1 \times \cdots \times U_m \to S$$
 (2)  
其中:

- $(1)U_i = \{f_{i0}, f_{i1}, \dots, f_{iy}\}$ 表示多值变量的状态集合;
- (2)S为 MDD 终端节点的有限值域;
- (3) MDD 的节点包括终端节点和非终端节点,终端节点从 S 中取值,非终端节点用  $d_i$  及 y+1 个指针表示,指向y+1个其他 MDD 节点。

MDD 的具体表现形式如图 2 所示。

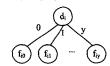


图 2 MDD 结构图

MDD 的终端节点取值为 0 和 1,0 表示 s 和 t 之间不存在一条连通路径,1 表示 s 和 t 之间存在一条连通路径。

## 3.2 MDD的自定义操作

计算网络可靠度,首先构造网络的 MDD,然后遍历 MDD 计算可靠度的值。下面给出 MDD 操作的定义。

定义 4 给定离散函数  $A = case(a, A_1, A_2, \dots, A_n)$  和  $B = case(a, B_1, B_2, \dots, B_n)$ ,根据 a 和 b 索引值的不同,A 和 B 的操作形式化描述为式(3):

 $A \lozenge B = case(a, A_1, A_2, \dots, A_n) \lozenge case(b, B_1, B_2, \dots, B_n)$   $= \begin{cases} case(a, A_1 \lozenge B_1, A_2 \lozenge B_2, \dots, A_n \lozenge B_n), & index(a) = index(b) \\ case(a, A_1 \lozenge B_1, A_2 \lozenge B_2, \dots, A_n \lozenge B_n), & index(a) < index(b) \\ case(b, A_1 \lozenge B_1, A_2 \lozenge B_2, \dots, A_n \lozenge B_n), & index(a) > index(b) \end{cases}$ 

(3)

其中,◇代表如布尔操作、算术操作以及对离散函数的自定义

操作等操作。为了方便构造和操作 MDD,本节自定义关于 MDD 的 And\_MDD 操作算子,And\_MDD 操作算子的取值如表 1 所列。

表 1 And\_MDD 操作算子的取值

Α	В	And_MDD
0	f <sub>B0</sub>	
•••	•••	0
0	$f_{By}$	
fAx	1	
•••	•••	. 1
$f_{\mathbf{A}\mathbf{x}}$	1	
fAy	fBo	f <sub>Ay</sub> U f <sub>B0</sub>
		•••
fAy	$f_{\mathrm{By}}$	f <sub>Ay</sub> U f <sub>By</sub>

其中,A 和 B 的取值  $f_{ij}$  参考定义 1, U 表示并运算。

下面基于图 3(a) 和图 3(b) 来说明  $And_MDD$  操作算子,假设  $s_1 \cup s_2$  内存在一条从 s 到 t 的可行路径。图 3(c) 和图 3(d)分别是在 index(a) < index(b)和 index(a) > index(b)的情况下由  $And_MDD$  操作算子构建的 MDD。

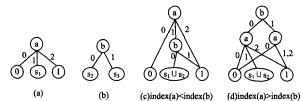


图 3 And\_MDD 操作

## 3.3 遍历 MDD 计算网络可靠度

计算网络可靠度需要递归遍历表示网络的 MDD,如式 (4)所示,生成的 MDD 对应式 (2)的函数 f。

$$NEF\_Rel = P(f) = \begin{cases} 0, & f = 0 \\ 1, & f = 1 \\ \sum_{j=0}^{y} p_{j} P(f_{j}), & \text{else} \end{cases}$$
 (4)

从上式可以看出,计算当前 MDD 可靠度只与生成 MDD 的节点数目有关,只需要遍历一次 MDD 就可以计算出网络的可靠度。

#### 4 算法设计

#### 4.1 算法描述

本节提出计算节点和边不可靠网络可靠度的 NEF\_MDD 算法。NEF\_MDD 算法的输入为节点和边不可靠网络 G、节点和边状态向量 M 和可靠度向量 P,输出为网络图 G 的连通可靠度  $NEF\_REL$ 。NEF\_MDD 算法分两步进行:

步骤 1 把网络图 G 的节点集 V 和边集 E 划分为集合  $H_1 \cong H_m$ ,枚举并化简  $H_i$  中的状态,得到化简后的状态向量  $U_i$  和概率向量  $W_i$ 。

步骤 1 通过调用 EncodeGraph(V, E) 函数实现, EncodeGraph(V, E) 函数的伪代码如图 4 所示。

EncodeGraph (V,E)

- 1. for (i←1 to |V|)
- 2. for  $(j\leftarrow 1 \text{ to } |E|)$
- 3. if (e; 是 v; 的邻接边) E; ←E; +e;
- 4.  $V \leftarrow V v_i$ ,  $E \leftarrow E E_i$ ,  $H_i \leftarrow v_i + E_i$
- 5. for (i  $\leftarrow$ 1 to |H|)
- 6. if (v<sub>i</sub>是s或t)

- 7. if  $(|H_i|=1)$
- 8.  $f_{i0} \leftarrow 0$ ,  $p_{i0} \leftarrow p_{1i0}$
- 9.  $f_{i1} \leftarrow \{v_i\}, p_{i1} \leftarrow p_{1i1}$
- 10. else if  $(|H_i| \geqslant 2)$
- 11.  $f_{i0} \leftarrow \{\overline{v_i}\}, p_{i0} \leftarrow p_{1i0} + p_{1i1} * p_{2i0}^{|Hi|-1}, k \leftarrow 2$
- 12. for  $(m-1 \text{ to } 2^{|Hi|-1}-1)$
- f<sub>ik</sub>←{v<sub>i</sub>,处于(1&m)状态的 ve<sub>ii</sub>, …,处于(2<sup>|Hi|-2</sup>&m)状态的 ve<sub>i(Hi|-1)</sub>}

- 15. k**←**k+1
- 16. else if (v<sub>i</sub> 不是 s 或 t)
- 17.  $f_{i0} \leftarrow {\{\overline{v_i}\}}, p_{i0} \leftarrow p_{1i0} + p_{1i1} * p_{2i0}^{|Hi|-1}, k \leftarrow 2$
- 18. for  $(m-1 \text{ to } 2^{|H_i|-1})$
- 19. f<sub>ik</sub> ← {v<sub>i</sub>, 处于(1&m)状态的 ve<sub>il</sub>, ····, 处于(2<sup>|Hi|-2</sup>&m)状态的 ve<sub>i(|Hi|-1)</sub>}
- 20.  $p_{ik}$ ←  $p_{lil} \prod_{m=1}^{|Hi|-1} Pr(处于(2^{m-1} \& m)状态的 ve_{in})$
- 21. k**←**k+1
- 22. return H,U,W

#### 图 4 EncodeGraph 函数伪代码

步骤 2 确定一个较优的变量序能够有效地减小决策图规模,然而,找出最优变量序是一个 NP 完全问题 [16],本文采用深度优先搜索的方法遍历网络图来确定变量序。在确定变量序集合 index[|U|]后,按照变量序构建 MDD 的一般过程为:根据定义 3 构建  $U_i$  的  $MDD_i$ ,并对  $MDD_i$  叶子节点进行化简,化简规则满足式(1)。遍历并判定叶子节点的取值,判定规则为:如果节点表示的集合内不存在一条从 s 到 t 的路径,则把该节点赋值为 0;如果节点表示的集合内存在一条从 s 到 t 的路径,则把该节点赋值为 1;如果叶子节点在哈希表中,则合并该节点,否则将该节点加入哈希表。然后,利用定义 3 生成  $U_{i+1}$  的  $MDD_{i+1}$ ,对  $MDD_i$  和  $MDD_{i+1}$ 进行  $And_{i+1}$  的  $MDD_{i+1}$ ,对  $MDD_{i+1}$  进行  $And_{i+1}$  的  $MDD_{i+1}$  计  $MDD_{i+1}$  是  $MDD_{i+1}$ 

在 And\_MDD操作后,不同叶子节点可能会出现同构的子网。目前还没有多项式复杂度的方法来判定子网同构,而哈希表由于能够有效减少存储开销和计算开销,常被用来识别和处理子网同构问题。本文采用文献[5,9]提出的网络位矢量的表示方法来设计哈希表。

步骤 2 通过调用 GenerateMdd(G,U,W)函数实现,函数 伪代码如下。

GenerateMdd(G,U,W)

- 1.  $i \leftarrow index[1]$
- 2. MDDA=CreateMdd(Ui, Wi)
- 3. nodeA←MDD; 的根节点
- 4. QUEUE, PUSH(nodeA)
- 5. for( $i \leftarrow 2$  to index[|U|])
- 6.  $MDDB = CreateMdd(U_i, W_i)$
- 7. for (k←1 to QUEUE, LENGTH)
- 8. tempnode←QUEUE. FRONT()
- 9. ncs←tempnode 的孩子节点数目
- 10. childnode[ncs]← tempnode 的所有孩子节点
- 11. for  $(j \leftarrow 1 \text{ to nes})$
- 12. if (childnode[j]=0 或 childnode[j]=1) continue
- 13. else

- 14. S←childnode[j]对应的状态
- 15. S←ReduceUnion(S)
- 16. if (S 中 s 或 t 失效或者 s 或 t 与 G 中邻接节点不连通) childnode[j]←0
- 17. else if (S中存在—条从 s 到 t 的可行路径) childnode[j] ← 1
- 18. else if (S在 hash 表中)

合并 childnode[j]与其同构的节点

- 19. else 把 childnode[j]和 S加入 hash 表
- 20. QUEUE. PUSH(childnode[j])
- 21. QUEUE. POP()
- 22. MDDA ← And MDD(MDDA, MDDB)
- 23. NEF\_REL= CompRel (nodeA)
- 24. return NEF\_REL

#### 图 5 GenerateMdd 函数伪代码

函数 ReduceUnion(S) 是按照式(1) 化简节点和边子集合的操作;函数 CreateMdd(U,W) 是根据 U 和 W 创建 MDD 的操作,满足定义 3;函数  $And\_MDD(MDDA,MDDB)$  是合并两个 MDD 的操作,满足定义 4;函数 CompRel(node) 是按照式(4) 计算 MDD 节点可靠度数值的递归函数, CompRel(node) 份代码如下。

CompRel (node)

- 1. if (node 的概率值=0) return 0
- 2. if 当前节点为终端节点
- 3. if (node=1) return 1
- 4. else return 0
- 5. ncs ← tempnode 的孩子节点数目
- 6. childnode[ncs] ← tempnode 的孩子节点
- 7. for i←1 to ncs
- 8. value ← 当前节点的概率值
- 9. value ← value \* CompRel (childnode[i])
- 10. result ← value+result
- 11. return result

图 6 CompRel 函数伪代码

#### 4.2 实例分析

本节所用测试网络取自文献[5]的图 1,如图 1(a)所示,节点和边正常工作的概率都是 0.9,失效概率为 0.1。下面基于图 1(a)对 NEF\_MDD算法展开说明,具体过程如下:

步骤 1 依据算法步骤 1,将网络划分为 3 个集合,分别 是  $H_1 = \{s, e_1, e_2\}$ ,  $H_2 = \{v_1, e_3\}$ ,  $H_3 = \{t\}$ ; 枚举并合并  $H_1$  至  $H_3$  3 个集合的状态空间,得到  $H_1$  至  $H_3$  的状态集合  $U_1$  至  $U_3$ ,计算得到  $U_1$  至  $U_3$  的可靠度向量  $W_1$  至  $W_3$ ,如表 2 一表 4 所列。

表 2 集合  $H_1$  的  $U_1$  和  $W_1$ 

s	0	0	0	0	1	1	1	1
$e_1$	0	0	1	1	0	0	1	1
$e_2$	0	1	0	1	0	1	0	1
			$\{\bar{s}\}$			$\{s,e_2\}$	$\{s,e_1\}$	$\{\mathbf{s},\mathbf{e}_1,\mathbf{e}_2\}$
状态 fi			0			$f_{11}$	$f_{12}$	f <sub>13</sub>
概率 Pij		(	0.109	9		0.081	0.081	0.729

表 3 集合  $H_2$  的  $U_2$  和  $W_2$ 

v <sub>1</sub>	0	0	1	1
$e_3$	0	1	0	1
	$\{\overline{\mathbf{v_1}}\}$		$\{\mathbf{v}_1\}$	$\{\mathbf{v}_1$ , $\mathbf{e}_3\}$
状态 f <sub>i</sub>	f <sub>20</sub>		$f_{21}$	$f_{22}$
概率 P <sub>3j</sub>	0.1	l	0.09	0.81

表 4 集合  $H_3$  的  $U_3$  和  $W_3$ 

t	0	1
	$\{\overline{\mathbf{t}}\}$	{t}
状态 fi	0	f <sub>31</sub>
概率 P <sub>4j</sub>	0.1	0.9

步骤 2 结合定义 1 + U 内元素  $f_{ij}$  的取值范围,依据定义 3 构造  $U_1$  至  $U_3$  的 MDD,如图 7 所示。根据深度优先搜索确定节点的变量序为  $\pi:U_3 < U_2 < U_1$ ,按照算法步骤 2,构建图 1(a)的 MDD,如图 8 所示。

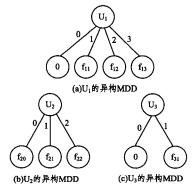


图 7  $U_1-U_3$  的 MDD

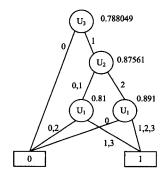


图 8 图 1(a)的 MDD

根据式(4)遍历 MDD, 计算出图 1(a)的可靠度 NEF\_REL 为 0.788049, 具体过程已在图 8 上标出。该数值与文献 [5]对图 1(a)计算的可靠度数值完全相同,且文献 [5]构建的 BDD有 6 层 8 个节点(不包含终端节点 0 和 1), NEF\_MDD 算 法构建的决策图有 3 层 4 个节点, 优于文献 [5] 构建的 BDD。

## 5 实验结果

本节用 C++语言实现 NEF\_MDD 算法,所用到的 MDD 函数库是爱荷华州立大学开发的 Meddly-0.7,程序运行环境为 CentOS-5.5,内核版本为 2.6.18。所有实验网络节点和边的可靠度数值均设置为 0.9,网络的黑实点表示源端和终端。

从表 5 的实验结果可以看出,对于同一网络图,NEF\_MDD 算法和 BDD 算法计算出的可靠度数值相同。BDD 内单个决策图变量表示一个节点或边,因而构建的二元决策树层数等于节点数边数的总和。NEF\_MDD 算法单个决策图变量表示单个节点及其未访问邻接边,构建的 MDD 层数等于节点数,因而从决策图层数上讲,NEF\_MDD 算法优于 BDD类算法。在图 9 中,网络 1 和 4 分别取自文献[5]的图 4 和图 9 的网络 1,网络 2 取自文献[9]的图 3一图 17,对应的 BDD节点数目分别为 16,423,24。对比表 5 中 MDD 的节点数目可以看出,NEF\_MDD 算法构建的 MDD 节点规模上优于BDD。

表 5 实验结果

网络	节点数	边数	可靠度	BDD 层数	MDD 层数	MDD 节点数
1	4	5	0.76008	9	4	8
2	4	6	0.80559	10	4	15
3	6	8	0.78082	14	6	56
4	6	15	0.80979	21	6	295
5	8	12	0.79270	20	8	339
6	10	15	0.78902	25	10	1521

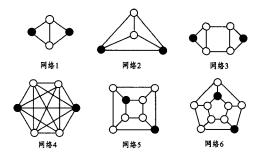


图 9 实验基准网络

结束语 本文引人 MDD 计算节点和边不可靠网络的可靠度,构造了节点和边不可靠网络的形式化模型,给出了分析节点和边不可靠网络可靠度的 NEF\_MDD 算法。由于每个MDD 变量被用来描述单个节点和其未访问的邻接边,因此,NEF\_MDD 算法构建的决策图层数等于网络节点数目。实例分析及实验结果表明,NEF\_MDD 算法构建的 MDD 层数和节点规模小于 BDD 算法构建的决策图。

MDD 的构建过程中会生成同构子网,同构子网会增加算法的时间和空间开销。设计一个较优的同构子网处理方法是未来的研究方向。

# 参考文献

- [1] Ball M O. Complexity of network reliability computations[J]. Networks, 1980, 10(2):153-165
- [2] Lin Y K. A simple algorithm for reliability evaluation of a stochastic-flow network with node failure[J]. Computers & Operations Research, 2001, 28(13): 1277-1285
- [3] Lin Y K, Huang C F. Assessing reliability within error rate and time constraint for a stochastic node-imperfect computer network[J]. Proceedings of the Institution of Mechanical Engineers Part O—Journal of Risk and Reliability, 2013, 227:80-85
- [4] Yan Zong-shuai, Nie Chen-hua, Dong Rong-sheng, et al. A Novel OBDD-Based Reliability Evaluation Algorithm for Wireless Sensor Networks on the Multicast Model [J]. Mathematical Problems in Engineering, 2015, 2015; 1-14

- [5] Kuo S Y, Yeh F M, Lin H Y, Efficient and exact reliability evaluation for networks with imperfect vertices [J]. IEEE Trans. Reliability, 2007, 56(2):288-300
- [6] Jiang Yi-nan, Li Rui-ying, Huang Ning, et al. Survey on network reliability evaluation methods [J]. Computer Science, 2012, 39 (5);9-13(in Chinese)

  江逸楠,李瑞莹,黄宁,等. 网络可靠性评估方法综述[J]. 计算机
  科学,2012,39(5);9-13
- [7] Theologou O R, Carlier J G. Factoring and reductions for networks with imperfect vertices [J]. IEEE Trans. Reliability, 1991,40(2):210-217
- [8] Sun Yan-rui, Cui Li-yan, Zhang De-xiang. A factoring algorithm for reliability evaluation of distributed networks with imperfect nodes[J]. Computer Science, 2002, 29(4), 111-113(in Chinese) 孙艳蕊,崔立彦,张祥德. 计算具有不可靠结点分布式网络可靠 度的一个因子分解算法[J]. 计算机科学, 2002, 29(4), 111-113
- [9] Xiao Yu-feng. Reliability analysis of two terminals network based on discrete probability model [D]. Beijing, Beijing University of Posts and Telecommunications, 2009 (in Chinese) 肖字峰. 基于离散概率模型的二端网络可靠性分析[D]. 北京:北京邮电大学,2009
- [10] Yeh W C. A simple heuristic algorithm for generating all minimal paths[J]. IEEE Transactions on Reliability, 2007, 56(3): 488-494
- [11] Sun Yan-rui, Reliability evaluation of stochastic-flow network under both time and cost constraints[J]. Journal of Northeastern University (Natural Science), 2013, 11(34): 1537-1541(in Chinese)
  - 孙艳蕊. 带时间和成本约束的随机流网络可靠度的计算[J]. 东北大学学报(自然科学版),2013,11(34);1537-1541
- [12] Nagayama S, Sasao T. On the optimization of heterogeneous MDDs[J]. IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(11):1645-1659
- [13] Xing L, Dai Y. A new decision-diagram-based method for efficient analysis on multistate systems[J]. IEEE Transactions on Dependable and Secure Computing, 2009, 6(3):161-174
- [14] Herrmann J U, Soh S, West G, et al. Using Multi-valued Decision Diagrams to Solve the Expected Hop Count Problem[C]//
  The IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA 2009). Bradford, United Kingdom, 2009; 419-424
- [15] Bollig B, Wegener I, Improving the variable ordering of OBDDs is NP-complete[J], IEEE Trans. Computers, 1996, 45:993-1002

#### (上接第132页)

- [13] Wang Wei, Kim D, Kyung AM, et al. On Construction of Quality Fault-Tolerant Virtual Backbone in Wireless Networks[J]. IEEE/ACM Transactions on Networking, 2013, 21 (5): 1499-1510
- [14] Schleich J, Danoy G, Bouvry P, et al. Blackbone2, an efficient deterministic algorithm for creating 2-connected m-dominating set-based backbones in ad hoc networks[C]//Proc. of the 7th ACM International Symposium on Mobility Management and Wireless Access. New York, USA, 2009:91-98
- [15] Zheng Chan, Yin Ling, Sun Shi-xin, Constructing 2-connected k-dominating sets for fault-tolerant backbone in wireless sensor

- networks[J]. Control and Decision, 2013, 28(5): 650-656(in Chinese)
- 郑婵,尹令,孙世新. 无线传感器网络中 2-连通 k-支配的容错连通支配集构造[J]. 控制与决策,2013,28(5),650-656
- [16] Li Ying-shu, Wu Yi-wei, Ai Chun-yun, et al. On the construction of k-connected m-dominating sets in wireless networks[J]. Combinatorial Optimization, 2012, 23(1); 118-139
- [17] Wightman P M, Labrador M A. Atarraya; a simulation tool to teach and research topology control algorithms for wireless sensor networks [C] // Proc. of 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, 2009; 26-35