

基于二元决策图的节点不可靠网络可靠度计算

肖宇峰^{a,b}, 张 华^{a,b}

(西南科技大学 a. 信息工程学院; b. 特殊环境机器人技术四川省重点实验室, 四川 绵阳 621010)

摘 要: 针对节点不可靠网络可靠度计算效率较低的问题, 提出一种基于二元决策图的网络可靠度计算方法。通过因子分解得到节点可靠网络的有序二元决策图(OBDD), 根据节点和边的关系对边的变量节点执行边替换操作, 生成节点不可靠网络的 OBDD, 并利用其高效存储结构提高不可靠节点的处理效率。在遍历 OBDD 计算可靠度时, 引入 Hash 表以避免对同一节点的重复访问, 从而减少冗余计算, 进一步提高计算效率。在基准网络中的对比实验结果表明, 该方法不仅能正确计算网络可靠度, 而且能快速分析大型网络。

关键词: 网络可靠度; 二元决策图; 不可靠节点; 因子分解; 布尔变量

中文引用格式: 肖宇峰, 张 华. 基于二元决策图的节点不可靠网络可靠度计算[J]. 计算机工程, 2015, 41(1): 87-91.

英文引用格式: Xiao Yufeng, Zhang Hua. Reliability Computation of Network with Unreliable Nodes Based on Binary Decision Diagram[J]. Computer Engineering, 2015, 41(1): 87-91.

Reliability Computation of Network with Unreliable Nodes Based on Binary Decision Diagram

XIAO Yufeng^{a,b}, ZHANG Hua^{a,b}

(a. Information Engineering School; b. Special Environment Robot Technology Key Laboratory of
Sichuan Province, Southwest University of Science and Technology, Mianyang 621010, China)

[Abstract] According to the inefficient reliability computation of network with unreliable nodes, this paper proposes a network reliability computation method based on Binary Decision Diagram (BDD). After the Ordered Binary Decision Diagram (OBDD) construction with factoring theorem, this method executes the edge replacements to OBDD variables of edges with the relation between nodes and edges, and the OBDD of network with unreliable nodes is constructed. Based on efficient OBDD structure, the computations about unreliable nodes are improved. Furthermore, a hash table is used to avoid the repeated access to same node when the reliability is calculated with OBDD traversal. Therefore, the redundant calculations are reduced, and the reliability computation efficiency of network with unreliable nodes is enhanced. Experiments are arranged on benchmark networks, and data analysis shows that this method can accurately calculate network reliability and quickly analyze some large networks.

[Key words] network reliability; Binary Decision Diagram (BDD); unreliable node; factoring; Boolean variable

DOI: 10.3969/j.issn.1000-3428.2015.01.016

1 概述

对于当前通信量急剧增加的骨干网络^[1-2]、广泛应用的无线传感器网络以及灾变环境下的数据网络^[3], 节点失效带来的网络可靠性问题正被工程人员和学者广泛关注^[4-5]。对节点不可靠网络进行可靠性分析时, 需要解决 2 个问题: (1) 如何处理不可靠节点^[6]; (2) 当网络规模较大时, 如何提高计算效

率^[7-8]。文献[3]应用因子分解来评估无线传感器网络的可靠度, 该方法的计算开销随着网络规模增大快速增长^[9]; 文献[10-11]应用分区等价类减少同构子网带来的重复计算, 大幅提高了网络可靠度计算效率, 但未考虑不可靠节点。文献[12-13]提出应用二元决策图(Binary Decision Diagram, BDD) 计算节点不可靠网络的可靠度, 基于位向量的 Hash 表存储开销随着网络规模扩大快速增加。本文提出一种基

基金项目: 国家核能开发科研基金资助项目([2011]1137); 四川省科技支撑计划基金资助项目(2013GZX0152); 四川省教育厅基金资助重点项目(14ZA0091)。

作者简介: 肖宇峰(1978-), 男, 副研究员、博士, 主研方向: 网络通信, 计算机系统可靠性评估; 张 华, 教授、博士。

收稿日期: 2014-01-20 **修回日期:** 2014-03-20 **E-mail:** xiaoyf_swit1@163.com

于二元决策图的节点不可靠网络可靠度计算方法。首先按确定边序分解网络并生成节点可靠时的有序二元决策图(Ordered BDD, OBDD), 然后对 OBDD 变量节点执行边替换操作, 生成节点不可靠时网络的 OBDD, 最后遍历 OBDD 计算网络可靠度。

2 符号表示

本文使用到的符号定义如下:

p_{src} : 网络的源端;

p_{dst} : 网络的终端;

$G(V, E)$: 表示网络拓扑结构的图, 其中, V 是图的点集, E 是图的边集;

$f_{rel}(G)$: 网络 G 的可靠度计算函数;

$O_{obdd}(G)$: 网络 G 的 OBDD 结构;

$f_{r-o}(O_{obdd}(G))$: 根据网络 OBDD 计算可靠度的函数;

G_{+e} : 收缩 G 中 e 边生成的子网;

G_{-e} : 删除 G 中 e 边生成的子网。

此外:

(1) 本文讨论的可靠度是指网络源端和终端之间保持连通的概率。源端和终端之间存在一条连通的路径, 则认为网络可靠; 否则, 网络不可靠。

(2) 上述收缩和删除边的情形可扩展为: 连续收缩边 a 和 b 得到的子网表示为 G_{+a+b} ; 连续删除边 a 和 b 得到的子网表示为 G_{-a-b} ; 连续收缩边 a 后删除边 b 得到的子网表示为 G_{+a-b} 。

3 网络可靠度计算方法

本文在计算效率和不可靠节点处理方面增强了基于 OBDD 的网络可靠度计算方法: 首先利用边的因子分解创建节点可靠网络的 OBDD 结构; 其次根据边和节点的逻辑联系, 对边的 OBDD 变量节点执行边替换操作, 创建节点不可靠网络的 OBDD 结构; 最后遍历 OBDD 来计算网络可靠度, 并利用 Hash 表减少对同一 OBDD 节点的重复访问, 进一步提高计算效率。

3.1 基于因子分解方法的 OBDD 创建

3.1.1 因子分解的基本概念

边的因子分解方法通常假设网络边不可靠而节点可靠, 根据边的 2 种情况进行分解: 边可靠则收缩边, 把边的 2 个端点合并成一个点; 边不可靠则删除边, 把边的 2 个端点保留在网络中。基于因子分解的网络可靠度计算可用如下公式表示^[4]:

$$f_{rel}(G) = p \times f_{rel}(G_{+e}) + (1 - p) \times f_{rel}(G_{-e}) \quad (1)$$

其中, e 是网络的某条边; p 是该边可靠的概率; $f_{rel}(G_{+e})$ 是收缩 e 后网络的可靠度; $f_{rel}(G_{-e})$ 是删除

e 后网络的可靠度。

图 1 给出了利用边的因子分解创建 OBDD 的实例, 网络 G 及其分解得到的子网按边序 $e_0 \sim e_5$ 进行 6 次分解。其中, 灰色点表示包含或者合并了源端(终端)的节点, 图中省略了源端和终端分离的子网, 所有的分解过程最后聚焦为源端和终端合并为一个节点子网。显然, 根据确定边序执行因子分解把网络逐层分为多个子网, 每执行一次收缩边和删除边操作就得到更小的子网。图 1 中, 除了第一次外的每次分解都是在上次分解得到的子网之上做进一步分解, 得到更小的子网。图中的实线表示收缩边, 虚线表示删除边, L_6 层的灰色点并为一, 表示源端和终端之间存在连通路。

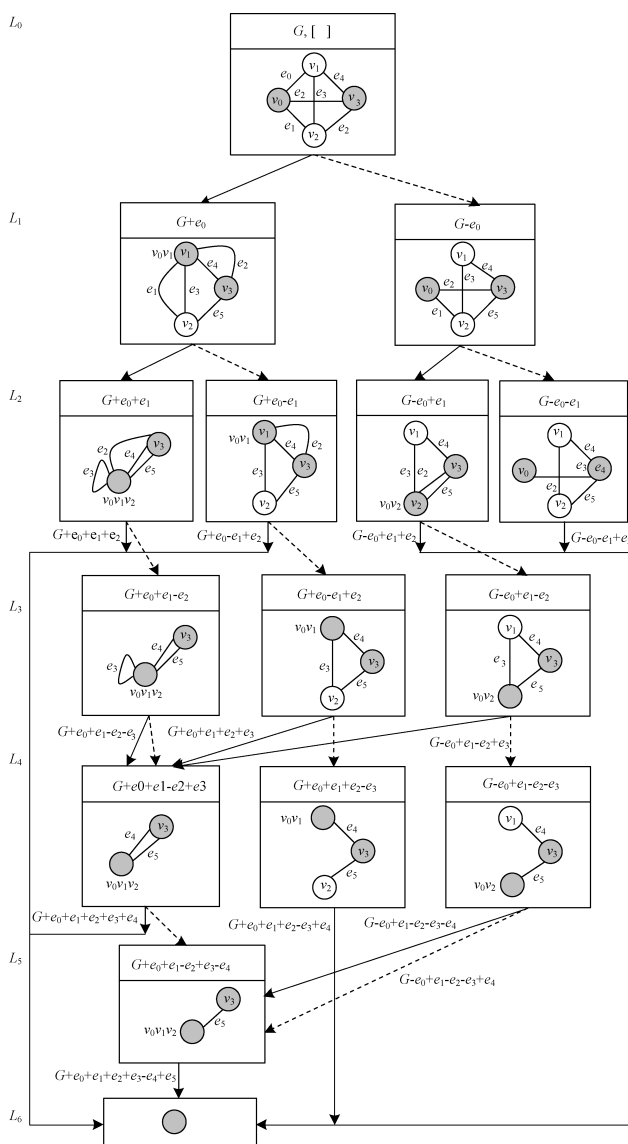


图 1 网络的因子分解

3.1.2 节点可靠网络的 OBDD 创建

创建 OBDD 时, 首先要确定边序 e_0, e_1, \dots, e_{m-1} , 然后根据边的逻辑(是否可靠)状态来分析网

络是否连通。利用 OBDD 的 ITE 运算可以把每次分解对应的逻辑判断记录下来, 最后得到描述整个网络的状态图^[7]。网络 G 的 OBDD 结构可通过下式描述:

$$bdd_ite(f, g, h) = (f \wedge g) \vee (\bar{f} \wedge h) \quad (2)$$

其中, f 是边序中某条边的 OBDD 表达式; g 是收缩边得到子网的 OBDD 结构; h 是删除边得到子网的 OBDD 结构。显然, g 和 h 需要对边序中其他边递归执行相同 ITE 运算得到。图 2 给出了图 1 中实例网络的 OBDD, 创建 OBDD 的具体步骤可参考文献[12-13]。

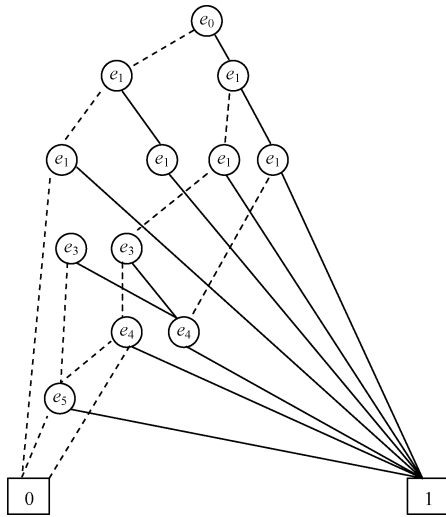


图2 节点可靠网络的 OBDD

3.1.3 节点不可靠网络的 OBDD 创建

前述 OBDD 创建过程假设网络节点可靠, 下文介绍一种应用逻辑运算处理不可靠节点的方法。

假设网络 G 的节点可靠时, OBDD 布尔变量 V_e 表示边 e , 则 G 的 OBDD 结构可表示为:

$$O_{obdd}(G) = \{V_e \wedge O_{obdd}(G) \mid_{V_e=1}\} \vee \{\bar{V}_e \wedge O_{obdd}(G) \mid_{V_e=0}\} \quad (3)$$

其中, $O_{obdd}(G) \mid_{V_e=1}$ 表示 e 处于可靠状态时 G 的 OBDD 结构; $O_{obdd}(G) \mid_{V_e=0}$ 表示 e 处于不可靠状态时 G 的 OBDD 结构。节点不可靠时需要考虑边的 2 个端点的逻辑状态, 用布尔变量 V_a 和 V_b 分别表示 e 的端点 a, b 。那么, e 的完整布尔表示式为 $V_a \wedge V_e \wedge V_b$ 。为得到节点不可靠网络 G 的 OBDD, 需要进行如下运算:

$$obdd1 = O_{obdd}(a) \text{ and } O_{obdd}(e)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(b)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e) = obdd2}$$

其中, $obdd2$ 是 $V_a \wedge V_e \wedge V_b$ 的 OBDD 结构。上述运算的含义是用端点不可靠的 e 替换 OBDD 中原有的 e , 执行了 OBDD 的 composition 运算^[10]。本文将该

运算过程称为边替换。

节点不可靠网络的 OBDD 创建算法伪码如下:

```
//get_terminals: 返回边的 2 个端点
//bdd_var: BDD 返回 BDD 节点对应的变量
//bdd_and: BDD 的逻辑 and 运算
//bdd_compose: BDD 的逻辑替换运算
//bdd_G 输入节点可靠网络的 OBDD
//newbdd_G 输出节点不可靠网络的 OBDD
BDDConstruct_urn(bdd_G)
{ //a, b 表示边 e 的 2 个端点
  newbdd_G = bdd_G;
  for (each node v of e in bdd_G)
  { e = bdd_var(v);
    get_terminals(e, a, b);
    obdd1 = bdd_and(bdd_ithvar(a), v);
    obdd2 = bdd_and(obdd1, bdd_ithvar(b));
    newbdd_G = bdd_compose(newbdd_G, obdd2,
    node); }
  return newbdd_G; }
```

对图 2 各边执行下列边替换操作后, 得到图 3 中节点不可靠时网络的 OBDD 结构。

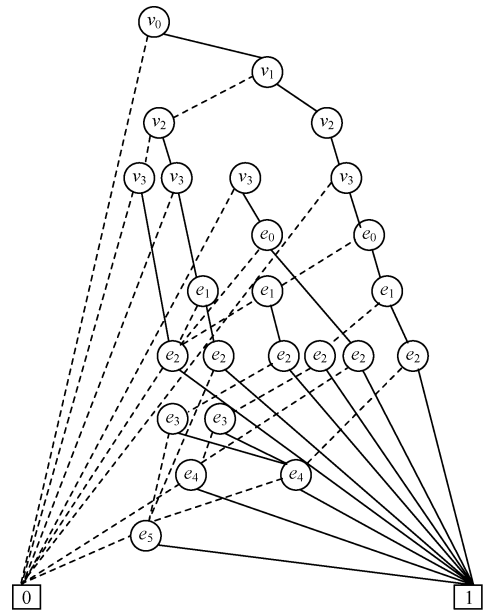


图3 节点不可靠网络的 OBDD

(1) $\langle v_0, e_0, v_1 \rangle$ 的布尔变量替代 $\langle e_0 \rangle$ 的变量:

$$obdd1 = O_{obdd}(v_0) \text{ and } O_{obdd}(e_0)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(v_1)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e_0) = obdd2}$$

(2) $\langle v_0, e_1, v_2 \rangle$ 的布尔变量替代 $\langle e_1 \rangle$ 的变量:

$$obdd1 = O_{obdd}(v_0) \text{ and } O_{obdd}(e_1)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(v_2)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e_1) = obdd2}$$

(3) $\langle v_0, e_2, v_3 \rangle$ 的布尔变量替代 $\langle e_2 \rangle$ 的

变量:

$$obdd1 = O_{obdd}(v_0) \text{ and } O_{obdd}(e_2)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(v_3)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e_2) = obdd2}$$

(4) $\langle v_1, e_3, v_2 \rangle$ 的布尔变量替代 $\langle e_3 \rangle$ 的

变量:

$$obdd1 = O_{obdd}(v_1) \text{ and } O_{obdd}(e_3)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(v_2)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e_3) = obdd2}$$

(5) $\langle v_1, e_4, v_3 \rangle$ 的布尔变量替代 $\langle e_4 \rangle$ 的

变量:

$$obdd1 = O_{obdd}(v_1) \text{ and } O_{obdd}(e_4)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(v_3)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e_4) = obdd2}$$

(6) $\langle v_2, e_5, v_3 \rangle$ 的布尔变量替代 $\langle e_5 \rangle$ 的

变量:

$$obdd1 = O_{obdd}(v_2) \text{ and } O_{obdd}(e_5)$$

$$obdd2 = obdd1 \text{ and } O_{obdd}(v_3)$$

$$O_{obdd}(G) = O_{obdd}(G) \mid_{O_{obdd}(e_5) = obdd2}$$

3.2 可靠度计算

3.2.1 可靠度计算原理及算法

按照上述改进方法创建网络 G 的 OBDD 结构后,可通过下面的递归公式来计算网络 G 的可靠度:

当 $O_{obdd}(G) = bddtrue$ 时, $f_{r,o}(O_{obdd}(G)) = 1$;

当 $O_{obdd}(G) = bddfalse$ 时, $f_{r,o}(O_{obdd}(G)) = 0$;

否则:

$$f_{r,o}(O_{obdd}(G)) = \Pr(x_k = 1) \times f_{r,o}(O_{obdd}(G) \mid_{x_k=1}) + \Pr(x_k = 0) \times f_{r,o}(O_{obdd}(G) \mid_{x_k=0})$$

其中, $0 \leq k < n$; $O_{obdd}(G) \mid_{x_k=1}$ 是 $x_k = 1$ 时网络 G 的 OBDD, 即 x_k 对应节点或边可靠时 G 的 OBDD 结构; $O_{obdd}(G) \mid_{x_k=0}$ 是 $x_k = 0$ 时网络 G 的 OBDD 结构, 即 x_k 对应节点或边不可靠时 G 的 OBDD 结构; $\Pr(x_k = 1)$ 是 x_k 对应节点或边的可靠概率, $\Pr(x_k = 0)$ 是 x_k 对应节点或边不可靠概率; $bddtrue$ 和 $bddfalse$ 分别表示 OBDD 结构中的逻辑真和逻辑假; k 的初始值为 0。显然, 完成上述公式的计算需要沿 OBDD 结构遍历其节点, 而且某些节点被多次访问。在图 4 中, e_2, e_4 和 e_5 在计算过程中被多次访问, 从而带来重复计算。为减少这种冗余计算, 设计算法时建立一个 hash 表, 以当前子网的 OBDD 根节点标号为索引记录当前子网的可靠度数值, 当再次访问该节点时直接返回已保存的数值。

下面的 NetRel_urn() 给出了节点不可靠网络的可靠度计算伪码。其中, G 代表网络; BDDConstruct() 是因子分解创建节点可靠网络 OBDD 的算法, 详细实现可参考文献[4]; BDDConstruct_urn() 是上文创建节点不可靠网络的 OBDD 算法; BDDRel() 是本小节上述计算方法的

实现。

//OrderEdge: 确定边序

//BDDConstruct: 创建节点可靠网络 OBDD

//BDDConstruct_urn: 节点不可靠时网络 OBDD

//BDDRel: 遍历 OBDD 计算网络可靠度

NetRel_urn(G)

{ OrderEdge(G);

bdd_G = BDDConstruct(G);

newbdd_G = BDDConstruct_urn(bdd_G);

reliability = BDDRel(newbdd_G);

return reliability; }

//Root: 返回 OBDD 的根节点

//rel_hash: 可靠度 hash 表

//LookupREHash: 查找 hash 表

//InsertintoREHash: 插入 hash 表

double OBDDRel(bdd bdd_graph)

{ if(bdd_graph is bddtrue) return 1.0;

else if(bdd_graph is bddfalse) return 0.0;

Root(bdd_graph);

p = operation probability of node v;

if (bdd_graph exists in the bdd_hash)

{ r = LookupREHash(bdd_graph);

return r; }

bdd bdd_high = high(bdd_graph);

bdd bdd_low = low(bdd_graph);

reliability = p * OBDDRel(bdd_high) + (1 - p) *

OBDDRel(bdd_low);

InsertintoREHash(bdd_graph, reliability);

return reliability; }

3.2.2 算法复杂度分析

本文算法由三部分组成: BDDConstruct, BDDConstruct_urn 和 BDDRel。其复杂度由其计算复杂度决定。文献[9]通过优化节点可靠网络的分解过程, 给出 BDDConstruct 和 BDDRel 复杂度的上界:

$$O_1 \leq O(m \times F_{\max}^3 \times B_{F_{\max}})$$

其中, O_1 为 BDDConstruct 和 BDDRel 的算法复杂度, m 为网络的边数, F_{\max} 为最大边界集的元素个数, $B_{F_{\max}}$ 是最大边界集的 Bell 数。这个值是现有研究中比较实用的上界值, 本文按文献[9]的方法优化网络分解后, BDDConstruct_urn 的复杂度就可由 BDD 宽度和深度计算。BDD 深度由网络的边数确定, BDD 宽度上界可根据文献[9]确定:

$$W_{bdd} \leq O((F_{\max}^2 + 1) \times B_{F_{\max}})$$

其中, W_{bdd} 是 BDD 宽度上界。由算法结构, BDDConstruct_urn 的复杂度上界为:

$$O_2 \leq O(m \times F_{\max}^2 \times B_{F_{\max}})$$

综上所述, 整个算法复杂度的上界为:

$$O_1 + O_2 \leq O(m \times F_{\max}^3 \times B_{F_{\max}}) + O(m \times F_{\max}^2 \times B_{F_{\max}}) = O(m \times (F_{\max}^3 + F_{\max}^2) \times B_{F_{\max}})$$

可见, F_{\max} 决定了算法的复杂度, 数百节点的网络一般都有较小的 F_{\max} 值。根据文献[9]优化后, 本

文算法的计算效率将得到进一步改善。

4 实验与结果分析

本文采用 Buddy 版 BDD 实现本文方法,并将其 C 语言实现程序命名为 NetRel_urn^[14]。为方便对比分析,同时实现了文献[3,13]的方法,并分别命名为 NetRel_yz 和 NetRel_x。实验计算机的 CPU 工作频率为 2.4 GHz,内存容量为 2 GB,操作系统为 2.6 内核的红帽 Linux。图 4 列出的 9 个网络被较多文献使用^[9-13],因此,本文将它们作为基准网络来检验计算的正确性和效率。网络中的黑点分别表示源端和终端,点和边可靠度是 0.9。

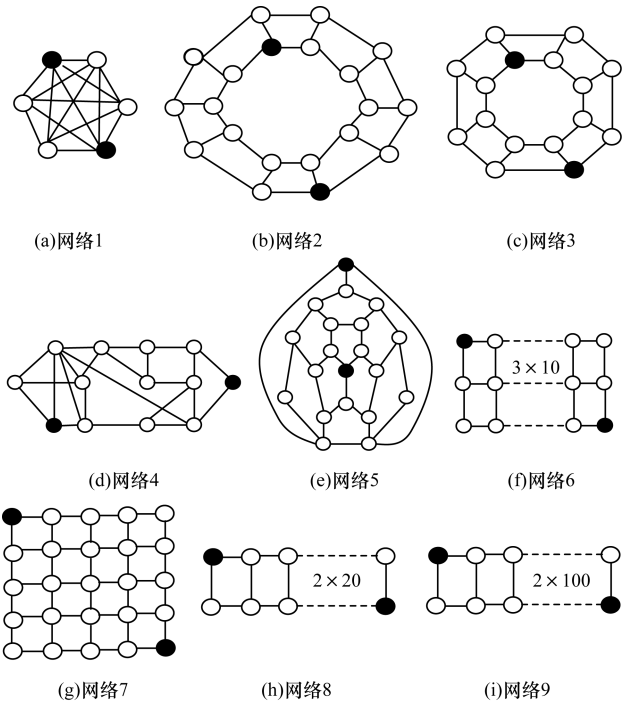


图 4 实验基准网络

实验结果如表 1 所示,其中,“-”表示开销超过 1 h。

表 1 计算时间比较

网络	网络可靠度	NetRel_yz	NetRel_x	NetRel_urn
1	0.809 79	11.265 10	5.142 67	0.831 77
2	0.748 17	32.174 60	12.831 4	2.436 91
3	0.769 78	7.513 79	7.397 11	1.122 17
4	0.752 26	337.431 00	94.149 10	1.425 14
5	0.784 79	1 431.270 00	103.752 00	5.416 68
6	0.609 63	-	529.624 00	23.413 50
7	0.711 99	-	335.028 00	8.017 31
8	0.230 16	-	-	57.031 20
9	0.001 11	-	-	416.231 00

每个程序被执行 10 次后,其平均计算时间被记录为表中的时间开销。可以看出,NetRel_yz 和 NetRel_x 的开销高于 NetRel_urn。从网络 6、网络 7、网络 8 和网络 9 的计算结果可以发现:随着网络规模增大,NetRel_yz 的计算开销显著上升,甚至在 1 h 内也无法

得到计算结果;NetRel_x 也无法计算网络 8 和网络 9 的可靠度;相比之下,NetRel_urn 完成了计算任务。实验结果表明,本文方法可有效评估节点不可靠网络的可靠度,当网络规模较大时,其计算开销较低。

5 结束语

为高效处理不可靠节点,本文提出了一种网络可靠度计算方法,利用 OBDD 变量节点的边替换操作和 OBDD 高效存储结构提高计算效率。实验结果表明,该方法能以较快的速度计算规模较大网络的可靠度。但本文未考虑节点和边失效的统计相关性,下一步将对其进行深入研究。

参考文献

[1] Ball M, Thomas M. Hand Books in Operations Research and Management Science, Network Models [M]. [S. l.]: Elsevier, 1995.

[2] Lin Y K, Chang P C. A Novel Reliability Evaluation Technique for Stochastic-flow Manufacturing Networks with Multiple Production Lines [J]. IEEE Transactions on Reliability, 2013, 62(1): 92-104.

[3] AboElFotouh H M F, Iyengar S S, Chakrabarty K. Computing Reliability and message delay for cooperative Wireless Distributed Sensor Networks Subject to Random Failures [J]. IEEE Transactions on Reliability, 2005, 54(1): 145-155.

[4] 肖宇峰. 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.

[5] 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.

[6] 吴 俊, 段东立, 赵 娟. 网络系统可靠性研究现状与展望 [J]. 复杂系统与复杂性科学, 2011, 8(2): 77-86.

[7] 肖宇峰, 李 昕, 李玉宏, 等. 用改进的 OBDD 方法计算通信网可靠度 [J]. 计算机应用研究, 2010, 27(3): 114-117.

[8] Huang N, Chen Y, Hou D, et al. Application Reliability for Communication Networks and Its Analysis Method [J]. Journal of Systems Engineering and Electronics, 2011, 22(6): 1030-1036.

[9] Hardy G, Lucet C, Limnios N. K-terminal Network Reliability Measures with Binary Decision Diagrams [J]. IEEE Transactions on Reliability, 2007, 56(3): 506-515.

[10] Imai H, Sekine K, Imai K. Computational Investigations of all-terminal Network Reliability via BDDs [J]. IEICE Transactions on Fundamentals, 1999, E82-A(5): 714-721.

[11] Carlier J, Lucet C. A Decomposition Algorithm for Network Reliability Evaluation [J]. Discrete Applied Mathematics, 1996, 65(1): 141-156.

[12] Kuo S Y, Yeh F M, Lin H Y. Efficient and Exact Reliability Evaluation for Networks with Imperfect Vertices [J]. IEEE Transactions on Reliability, 2007, 56(2): 198-211.

[13] Xing Liudong. An Efficient Binary Decision Diagram Based Approach for Network Reliability and Sensitivity Analysis [J]. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2008, 38(6): 105-115.

[14] Andersen H. An Introduction to Binary Decision Diagrams [M]. [S. l.]: Elsevier, 1997.

编辑 金胡考