

文章编号: 1001-0920(2011)01-0032-05

## 基于二元决策图的网络可靠性评估

何明<sup>a</sup>, 权冀川<sup>a</sup>, 郑翔<sup>b</sup>, 赖海光<sup>a</sup>, 杨飞<sup>a</sup>

(解放军理工大学 a. 指挥自动化学院, b. 通信工程学院, 南京 210007)

**摘要:** 提出一种改进二元决策图(BDD)的网络可靠性评估方法. 为了解决BDD构造中有效识别同构子图的问题, 将边收缩/删除法应用于BDD的图分解中, 并提出了BDD的宽度优先搜索算法, 通过遍历BDD图对边进行排序, 为布尔函数的不变化提供了一种新的高效途径. 实验结果表明, 该算法具有精确性高、时间复杂度低的优点, 可以避免常规最短路算法中进行不变化的大量运算, 并可应用于一些大规模的网络.

**关键词:** 二元决策图; 网络可靠性; 评估

中图分类号: TM623

文献标识码: A

## Network reliability evaluation based on binary decision diagrams

HE Ming<sup>a</sup>, QUAN Ji-chuan<sup>a</sup>, ZHENG Xiang<sup>b</sup>, LAI Hai-guang<sup>a</sup>, YANG Fei<sup>a</sup>

(a. Institute of Command Automation, b. Institute of Communication Engineering, PLA University of Science and Technology, Nanjing 210007, China. Correspondent: HE Ming, E-mail: blue\_horse@126.com)

**Abstract:** An improved method for evaluating the network reliability based on binary decision diagrams(BDD) is presented. To solve the structure in the effective identification of BDD subgraph isomorphism problem, the edge contraction/deletion is applied to this decomposition process of network graphs. and the width first search algorithm of BDD is proposed. By traversing BDD graph, the edges are sorted. A new and efficient way is provided for the Boolean function disjoint. The experiment results show that, the algorithm has the merit of high accuracy and low time complexity, and can avoid a large number of disjoint operations in the minimum path algorithm, which can be applied to a number of large-scale networks.

**Key words:** binary decision diagrams; network reliability; evaluation

### 1 引言

“安全、可靠、经济、高效”是网络建设和运行的目标, 其中可靠性是最直接的影响因素, 因此, 网络的可靠性成为一个备受关注的问题. 为了有更好的认识, 对网络可靠性研究的重大理论意义和应用价值也日益凸显. 人们需要关注, 这些网络到底有多可靠? 如何正确地评价网络的可靠性?

目前, 网络可靠性的评价方法可以分为模糊法和精确法两类. 其中精确评估法也大致分为两类: 1) 需要列举出网络中的最短路集或最小割集, 根据容斥原理对集合中的不相关元素进行统计得到可靠性值, 但是这类方法只适用于小型网络分析<sup>[1-3]</sup>; 2) 基于网络拓扑化简, 将网络中的一些特殊结构(如串并结构、多边形结构、三角形结构等)用简单的结构去替代, 得到相应的可靠性值, 直到整个网络缩减为一条链路. 但对于大多数网络, 只有网络中的一部分可以进行拓

扑化简. 为此, 研究者提出了因子分解的方法, 将问题分解为两个互斥的子问题进行分析, 并证明了因子分解方法比最短路集和最小割集方法更有效. 但是随着网络规模的扩大, 因子分解法也无法适用. 文献[4]将二元决策图方法引入了网络化简, 提出二元决策图(BDD)模型, 它是布尔函数表示下的一种图形方式, 已广泛应用于计算机网络、电子电路、逻辑综合和逻辑验证中. [5]对BDD进行了推广, [6]将其引入到可靠性分析中, 并由[7]进行了拓展. 自[8]最早应用于故障树分析以来, BDD技术在故障树分析中显示出计算效率高、结果精确、可方便利用计算机编程来实现等特点, 近年来日益受到重视, 取得不少应用. [9]证明了BDD方法的效率, 并且可以用于某些大型网络的可靠性分析. [10]提出的BDD算法是目前最有效的可靠性精确评估法.

收稿日期: 2009-10-21; 修回日期: 2010-06-14.

基金项目: 国家自然科学基金项目(60974086).

作者简介: 何明(1978—), 男, 副教授, 博士后, 从事信息安全的研究; 权冀川(1974—), 男, 副教授, 博士, 从事信息融合的研究.

本文利用网络节点分区的思想,对该技术进行改进,提出了采用改进 BDD 求解  $K$ -终端网络可靠性的方法.改进 BDD 的创新之处在于将边收缩/删除法应用于 BDD 的图分解中,解决了 BDD 构造中有效识别同构子图的问题.同时,算法中提出的 BDD 生成顺序的技术,可以很快到达 BDD 的叶结点,为布尔函数的不变化提供了一种新的高效途径.实验结果表明,该算法具有精确性高、时间复杂度低的性能,利用先进数据结构进行编码和操作布尔函数,可以避免常规最小路算法中进行不变化的大量运算,适用于一些大规模的网络.

## 2 理论基础

**定义 1**<sup>[11]</sup> BDD 是有向的、结点具有标号的二叉树  $(V, N)$ . 其中:  $N$  为标号集;  $V$  为结点集, 包含两类结点:

- 1) 具有结点值  $\text{value}(v) \in \{0, 1\}$  的叶结点;
- 2) 具有标号  $\text{index}(v) \in N$  的结点, 这类结点具有两个子结点  $\text{low}(v)$  和  $\text{high}(v) \in V$ , 规定的左子结点为  $\text{high}(v)$ , 右子结点为  $\text{low}(v)$ .

**定义 2**<sup>[11]</sup> 一个 BDD 中以结点  $v$  为根的子树表示的布尔逻辑函数  $f_v(x_1, x_2, \dots, x_n)$  为:

- 1) 当  $v$  是叶结点时, 有  $f_v = \text{value}(v)$ ;
- 2) 当  $v$  是具有标号  $\text{index}(v) = i$  的结点时, 有  $f_v(x_1, x_2, \dots, x_n) = \bar{x}_i f_{\text{low}(v)}(x_1, x_2, \dots, x_n) + x_i f_{\text{high}(v)}(x_1, x_2, \dots, x_n)$ .

## 3 利用改进 BDD 计算 $K$ -终端网络可靠性

### 3.1 $K$ -终端可靠性布尔函数

BDD 可以简洁地描述网络中运行和失效的全部状态, 通过合并等价子状态避免了列举中的冗余.  $K$ -终端网络函数可以由布尔函数  $f: 0, 1^m \rightarrow 0, 1$  描述为

$$f(x_1, x_2, \dots, x_m) = \begin{cases} 1, & k \text{ 个节点连接;} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

通过 BDD 选择函数编码, 设  $f$  为系统结构函数, 网络的  $K$ -终端可靠性  $R_K$  可以递归计算为

$$R_K(p; G) = P_r(f = 1) = p_i P_r(f_{X_i=1} = 1) + q_i P_r(f_{X_i=0} = 1), \quad \forall i \in [1, 2, \dots, m]. \quad (2)$$

通过递归计算, 可以得到  $f_{X_i=0}$  和  $f_{X_i=1}$  的值, 直到最后节点 (即  $f_{X_i=j} = 1, f_{X_i=j} = 0, j \in \{0, 1\}$ ). 通过在 BDD 上应用网络可靠性算法, 可得到  $K$ -终端可靠性表达式为

$$R_K(p; G) =$$

$$p_1(p_2(p_4 + q_4 p_5 p_6) + q_2(p_3(p_4 + q_4 p_5 p_6) + q_3 p_4 p_5 p_6)) + q_1 p_2(p_3(p_4 + q_4 p_5 p_6) + q_3 p_5 p_6). \quad (3)$$

## 3.2 构建基于边收缩/删除的 BDD

### 3.2.1 基于边收缩/删除的网络分解

分解过程可以用一个二叉树来表示. 以图 1 为例, 网络  $G$  的分解过程如下:

**Step 1:** 考虑边  $e_1$ , 并从原始图  $G$  构造出两个子图  $G_{-1}$  和  $G_{*1}$ .

**Step 2:** 考虑边  $e_2$ , 由子图  $G_{-1}$  构造出子图  $G_{-1-2}$  和  $G_{-1*2}$ , 由子图  $G_{*1}$  构造出相应的子图  $G_{*1-2}$  和  $G_{*1*2}$ , 舍弃子图  $G_{-1-2}$ , 因为  $K$ -顶点  $\alpha$  是分离的.

**Step 3:** 对于剩下的  $m-2$  层, 应用同样的分解, 直到  $K$  中所有顶点均已连通, 或者至少其中一个是分离的. 当  $K$ -顶点在相应的子图中被合并成一个独立顶点时, 可确认它们是连通的.

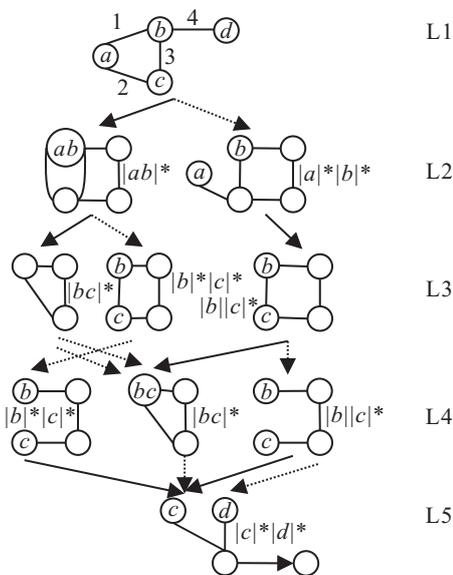


图 1 基于边删除/收缩的网络分解示意图

### 3.2.2 共享同构子图

文献 [12] 通过一些顶点分割来描述子图拓扑结构, 并通过共享同构子图, 拓展树被修订成为根无循环图 (即 BDD). 假设原始的边排序为  $e_1 < e_2 < \dots < e_k < \dots < e_m$ , 定义为  $E_k = \{e_1, e_2, \dots, e_{k-1}\}$  和  $\bar{E}_k = \{e_k, \dots, e_m\}$ . 在 BDD 的第  $k$  层,  $E_k$  中每个边均有固定的状态, 因此相应的子图只由边集  $\bar{E}_k$  构成.

**定义 3**<sup>[12]</sup> (边界集) 令  $F_k$  为顶点集, 其中的每个顶点都至少有一个边在  $E_k$  中, 一个边在  $\bar{E}_k$  中.  $F_k$  由第  $k$  层的边界集命名, 记  $F_{\max}$  是分解过程中边界集的最大尺寸, 且  $F_{\max} = \max\{|F_i|, i = 1, 2, \dots, m\}$ . 子图可以用  $F_k$  的分割来表示.

**定义 4**<sup>[12]</sup> (分割) 为了用分割来描述第  $k$  层的子图, 将  $F_k$  中节点根据以下规则分块:

- 1)  $F_k$  中两个顶点  $x$  和  $y$  在同一个块内, 当且仅当

$x$  和  $y$  在边集  $E_k$  收缩/删除过程中已经被合并成为一个顶点.

2) 如果至少一个  $K$ -顶点已与块内某个顶点合并, 则该块用星号标记. 存在一条由  $E_k$  中可运行的边组成的路径, 连通  $K$ -顶点且至少有一个顶点在该块中.

**定义 5**<sup>[12]</sup>(同构图) 令  $G_1$  和  $G_2$  是  $G$  由同一个边集  $E_k$  通过收缩/删除得到的子图, 如果相应的分割相同, 则子图  $G_1$  和  $G_2$  称为同边排序同构<sup>[13]</sup>.

### 3.2.3 分割的分类<sup>[10]</sup>

将同一层  $k$  的分割排序, 以便更有效地识别存储它们, 分割的分类(每个分割均有编号)可以在由  $\text{BDDPart}()$  构造 BDD 的过程中用来合并同构节点, 每个内部节点用分割的号码编号. 分割的排序由第 2 类斯特灵数产生, 有以下递归公式:

$$A_{ij} = jA_{i-1,j} + A_{i-1,j-1}, \quad 1 < j \leq i. \quad (4)$$

其中:  $A_{i,j} = 1, A_{i,j} = 0, 0 < i < j$ .  $A_{i,j}$  是  $j$  个块的分割数, 可以产生  $i$  个元素, 对于给定的元素数  $i$ , 排序伴随块数的增长, 利用递归生成. 首先将由  $i-1$  个元素的分割和  $j$  个块(每个块加上第  $i$  个元素)生成的分割排序; 然后将由  $i-1$  个元素的分割和  $j-1$  个块(将第  $i$  个元素加到新块)生成的分割排序, 依此类推. 边界集  $F_k$  的每个分割都对应一个号码, 介于  $1 \sim B_{|F_k|}$  之间. 这些分割的  $B_{|F_k|}$  可以由 Bell 数得到, 定义为

$$B_{|F_k|} = \sum_{j=1}^{|F_k|} A_{|F_k|,j}. \quad (5)$$

如果  $W_i$  是第  $i$  层节点的数目, 则 BDD 的宽度  $W$  定义为  $W = \max_{i=1,2,\dots,n} W_i$ .

**命题 1**  $W_{\text{th}}$  的理论宽度, 即对于给定的边界集  $F$ , 理论上被标记分割的最大数目为

$$W_{\text{th}}(|F|) = \sum_{j=1}^{|F|} \left( A_{|F|,j} \sum_{l=0}^{\min(|K|,j)} \binom{j}{l} \right). \quad (6)$$

### 3.2.4 BDDPart() 算法<sup>[10]</sup>

在构造解决  $K$ -终端可靠性问题的 BDD 过程中, 用标记分割和编号来代替图, 由此, 每个 BDD 节点用一个分割号码标识. 由于图可以用标记分割描述, 通过合并同构节点来构造 BDD. 利用 Hash 表记录每个共享同构子图的分割数, 由第  $k-1$  层的分割构造第  $k$  层分割时, 核对 Hash 表中是否存在新生分割对应的分割, 以此识别同构子图, 避免计算中的冗余. 算法  $\text{BDDPart}()$  构造描述  $K$ -终端可靠性函数的 BDD, 设用宽度优先搜索算法遍历图来对边进行排序. 图 2 展示了网络  $G$  利用分割分解的过程:

**Step 1:** 考虑边  $e_1 = (a, b)$ , 边界集等于  $\{a, b\}$ , 从

空分割开始构造分割  $[a, b]^*$ . 因为  $e_1$  可用, 所以顶点  $a$  和  $b$  在同一块中, 且由于  $a$  是  $K$ -顶点, 该块被标记. 同时构造分割  $[a]^*[b]$ ,  $e_1$  失效, 只有  $K$ -顶点  $a$  在标记的块中.

**Step 2:** 顶点  $c$  在新的边界集  $F_2$  中(考虑边  $e_2 = (a, c)$ ), 顶点  $a$  没有出现(邻接  $a$  的边  $e_1$  和  $e_2$  有固定状态), 因此新边界集  $F_2$  等于  $\{b, c\}$ . 由  $[ab]^*$  可以产生分割  $[bc]^*$  和  $[b]^*[c]^*$ , 只有分割  $[b][c]^*$  来自  $[a]^*[b]$ , 其他分割对应失效状态(顶点  $a$  非连通).

**Step 3:** 边界没有改变, 由本层的 3 个分割可以产生新分割  $[b]^*[c]^*$ ,  $[bc]^*$ , 每个分割的编号均存储在 Hash 表中, 且发现分割  $[bc]^*$  可以共享.

**Step 4:** 边界集等于  $\{c, d\}$ , 同理, 由  $[bc]^*$  产生一个新分割, 对应正常运行的子图可以分出叶子 1.

**Step 5:** 描述  $K$ -终端网络  $G$  的 BDD 构建成功.

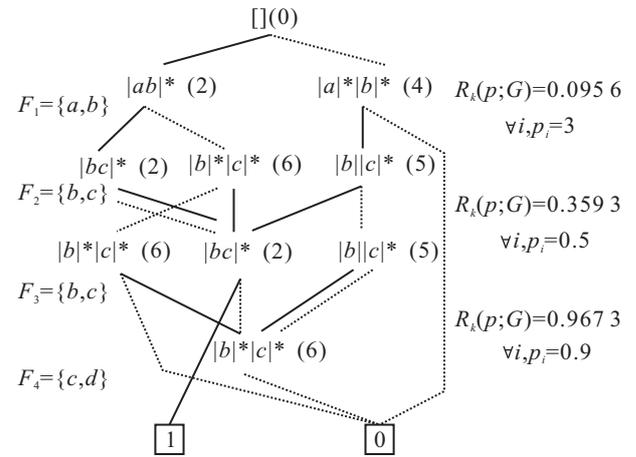


图 2 BDD 计算过程

### 3.3 计算复杂度

**定理 1**  $K$ -终端可靠性计算的复杂度为  $O(2^{F_{\max}} m F_{\max} B_{F_{\max}})$ .

**证明** BDD 的计算时间和所用内存依赖每层产生和列举的分割数目,  $F_{\max}$  表示边界集的最大尺寸, 由  $W_{\text{th}}(F_{\max})$  界定. 因为 BDD 层深为  $m$ , 分割总数上界为  $mW_{\text{th}}(F_{\max})$ , 由命题 1 可得

$$W_{\text{th}}(|F_k|) \leq \sum_{j=1}^{|F|} \left( A_{|F|,j} \sum_{l=0}^j \binom{j}{l} \right) \leq 2^{|F_k|} B_{|F_k|}. \quad (7)$$

在第  $k$  层,  $\theta(|F_k|)$  中最多产生  $W_{\text{th}}(|F_k|)$  个分割. 每个分割在  $k+1$  层有两个对应的分割需计算, 一个是删除  $\theta(|F_k|)$  中  $e_{k+1}$  产生的, 另一个是收缩  $\theta(|F_k|)$  中  $e_{k+1}$  产生的, 均可用  $\theta(|F_k|)$  中的函数进行编码. 因此, 算法时间复杂度为  $O(2^{F_{\max}} m F_{\max} B_{F_{\max}})$ , 所需内存复杂度为  $O(2^{F_{\max}} m B_{F_{\max}})$ .  $\square$

$F_{\max}$  与网络拓扑的线性宽度有关, 表示边界集合的最大值. 对于大部分网络, 除了网格网络等为简单结构复制形成的网络外,  $F_{\max}$  随着网络规模的增

加而增加.  $B_{F_{\max}}$  是  $F_{\max}$  节点集的 Bell 数,  $B_{F_{\max}}$  随  $F_{\max}$  的增长呈指数增加.

#### 4 实验结果分析

算法采用 Matlab (版本 7.0.0) 编写, 在 1.85 GHz 主频的 AMD 处理器的微机上运行. 实验中启发式的边排序和 BDD 中的变量, 使用的是宽度优先搜索排序, 计算速度依赖于  $F_{\max}$ .

将 BDD 算法用于图 3 和图 4 所示的 4 个网络, 因为 4 个网络的节点数目相同, 节点的可靠性值并不影响全网可靠性的结果. 假设节点完全可靠, 链路可靠概率均为 0.9, 得到的全网可靠性结果见表 1. 全网可靠性  $R$  越高, 则网络越可靠. 从表 1 可知, 网络可靠性的关系为  $R_c > R_b > R_a > R_{G_1}$ .

图 5 展示了全连通网络  $K_n$  的统一全终端可靠性, 该网络有  $n$  个顶点和  $n(n-1)/2$  条边, 其中所有的

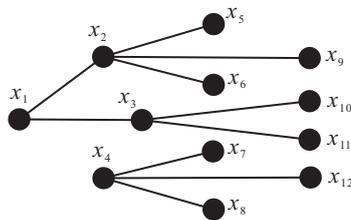
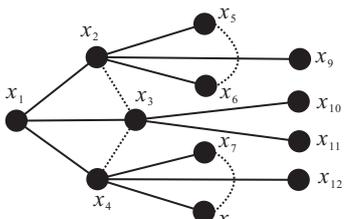
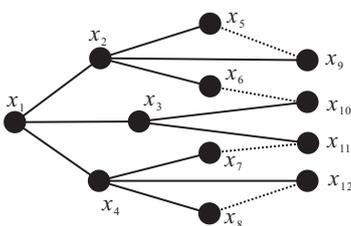


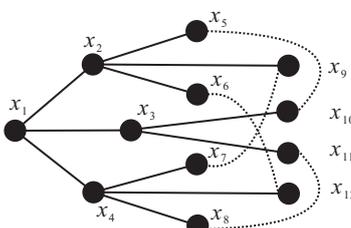
图 3 树状网络  $G_1$



(a) 改进网络 1



(b) 改进网络 2



(c) 改进网络 3

图 4 树状网络  $G_1$  增加 4 条链路的 3 种改进网络

表 1 树状网络  $G_1$  及 3 种改进网络的 BDD 算法可靠性结果

拓扑	树状网络 $G_1$	改进网络 a	改进网络 b	改进网络 c
全网可靠性 $R$	0.3138	0.6055	0.8401	0.8826

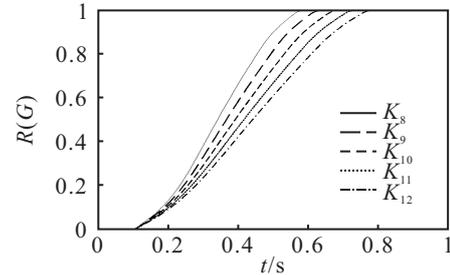


图 5 全连接网络中计算全终端可靠性

边均有相同的失效概率.

表 2 为计算 BDD 时的相应特性. 显然该拓扑结构具有最佳可靠性, 对于给定的可靠性  $p$ , 规模越大, 网络可靠性越好. 可以看到因为  $F_{\max} = |V| - 1$ , 随着全连通图规模的增长, 可靠性是单调增长的, 算法的指数因子是网络规模.

表 2 全连接网络基准

type	$ V $	$ E $	time	BDD	$F_{\max}$
$K_{08}$	8	28	0.03	2745	7
$K_{09}$	9	36	0.06	10265	8
$K_{10}$	10	45	0.13	39856	9
$K_{11}$	11	55	0.52	160793	10
$K_{12}$	12	66	2.14	673934	11

表 3 为 BDD 算法用于全连通网络的计算时间复杂度. 从表 3 的结果可以看出, 本文算法更适用于一般性的大型网络可靠性分析, 且对于大部分网络, 有

$$2^{F_{\max}} m F_{\max} B_{F_{\max}} > (m+n)n^{2.376}.$$

表 3 BDD 算法用于全连通网络的计算时间复杂度

网络	$F_8$	$F_{10}$	$F_{12}$
BDD 算法时间复杂度	$1 \times 10^7$	$8 \times 10^8$	$9.7 \times 10^8$

网络	$G_{40000 \times 4}$	$G_{1000 \times 7}$	$G_{10 \times 10}$
BDD 算法时间复杂度	$1.3 \times 10^9$	$2.6 \times 10^{10}$	$9.7 \times 10^8$

#### 5 结 论

本文提出的改进二元决策图的网络可靠性评估方法与改进前的方法相比, 能够提高数据结构的效率, 有效降低算法的复杂度, 提高算法的时间效率. 此外, 算法提出的 BDD 生成顺序的技术, 能够很快到达 BDD 的叶结点. 该方法可应用于多阶段任务系统的可靠性分析中, 且能够将其引入到故障树分析中, 有效地自动化分析与处理大型的故障树<sup>[13-16]</sup>.

## 参考文献(References)

- [1] Ahmad S H. Simple enumeration of minimal cutsets of acyclic directed graph[J]. IEEE Trans on Reliability, 1998, 27(5): 484-487.
- [2] Locks M O. A minimizing algorithm for sum of disjoint products[J]. IEEE Trans on Reliability, 1987, 36(4): 436-445.
- [3] Hariri S, Raghavendra C S. SYREL: A symbolic reliability algorithm based on path and cutset methods[J]. IEEE Trans on Computers, 1987, 36(10): 1224-1232.
- [4] Akers B. Binary decision diagrams[J]. IEEE Trans on Computers, 1978, 27(7): 509-516.
- [5] Bryant R E. Symbolic Boolean manipulation with ordered binary-decision diagrams[J]. ACM Computing Surveys, 1992, 24(3): 293-318.
- [6] Coudert O, Madre J C. Implicit and incremental computation of primes and essential primes of Boolean functions[C]. Proc of the 29th ACM/IEEE Design Automation Conf. IEEE Computer Society Press, 1992: 36-39.
- [7] Odeh K. New algorithms for probabilistic and logic analysis of fault trees[D]. Université de Technologie de Compiègne, 1995.
- [8] Rauzy A. A new methodology to handle Boolean models with loops[J]. IEEE Trans on Reliability, 2003, 52(1): 96-105.
- [9] Yeh F M, Lu S K, Kuo S Y. OBDD-based evaluation of  $k$ -terminal network reliability[J]. IEEE Trans on Reliability, 2002, 51(4): 443-451.
- [10] Hardy G, Lucet C, Linnios N.  $K$ -terminal network reliability measures with binary decision diagrams[J]. IEEE Trans on Reliability, 2007, 56(3): 506-515.
- [11] 武小悦, 沙基昌. 网络系统可靠度的BDD算法[J]. 系统工程与电子技术, 1999, 21(7): 72-73.  
(Wu X Y, Sha J C. A BDD algorithm for network reliability[J]. Systems Engineering and Electronics, 1999, 21(7): 72-73.)
- [12] Yeh F M, Lu S K, Kuo S Y. OBDD-based evaluation of  $k$ -terminal network reliability[J]. IEEE Trans on Reliability, 2002, 51(4): 443-451.
- [13] Gebre B, Ramirez-Marquez J. Element substitution algorithm for general two-terminal network reliability analyses[J]. IIE Trans, 2007, 39(3): 265-275.
- [14] Cancela H, Khadiri M E, Rubino G. Rare event analysis by monte Carlo techniques in static models[C]. Rare Event Simulation using Monte Carlo Methods. John Wiley and Sons Ltd., 2009: 145-170.
- [15] Reed D A, Lu C D, Mendes C L. Reliability challenges in large systems[J]. Future Generation Computer Systems, 2006, 22(3): 293-302.
- [16] Rauzy A. A new methodology to handle Boolean models with loops[J]. IEEE Trans on Reliability, 2003, 52(1): 96-105.

(上接第31页)

- [8] Boudia M, Prins C. A memetic algorithm with dynamic population management for an integrated production-distribution problem[J]. European J of Operational Research, 2009, 195(3): 703-715.
- [9] Chan Felix T S, Chung S H. Multi criterion genetic optimization for due date assigned distribution network problems[J]. Decision Support Systems, 2005, 39(4): 661-675.
- [10] Barrie M Baker, Ayechev M A. A genetic algorithm for the vehicle routing problem[J]. Computers and Operations Research, 2003, 30(5): 787-800.
- [11] 邢文训, 谢金星. 现代优化计算方法[M]. 北京: 清华大学出版社, 2005.  
(Xing W X, Xie J X. The modern optimization methodology[M]. Beijing: Tsinghua University Press, 2005.)
- [12] 彭春林, 梁春华, 周泓. 求解同时取货和送货车辆路径问题的改进遗传算法[J]. 系统仿真学报, 2008, 20(9): 2266-2270.  
(Peng C L, Liang C H, Zhou H. Improved genetic algorithm for vehicle routing problem with simultaneous pickups and deliveries[J]. J of System Simulation, 2008, 20(9): 2266-2270.)
- [13] 周泓, 冯允成. 一种启发式混合遗传算法及其在车间作业排序问题中的应用[J]. 航空学报, 1998, 19(1): 74-77.  
(Zhou H, Feng Y C. A kind of hybrid heuristic genetic algorithm with application to job shop scheduling[J]. Acta Aeronautica et Astronaut Ica Sinica, 1998, 19(1): 74-77.)