doi:10.11887/j.cn.201702028　　http://journal.nudt.edu.cn

多阶段任务系统可靠性分析的二元决策图模型*

孟　礼[1,2]

(1. 国防科技大学 信息系统与管理学院, 湖南 长沙　410073;
2. 装备学院 装备试验系, 北京　101416)

摘　要:为了实现计算机自动生成多阶段任务系统的二元决策图可靠性模型,提出了二元决策图可靠性建模的通用方法。定义了嵌套的二元决策图数据结构 BDD_Element,给出了二元决策图模型的描述和存储方法,提出了与门、或门和 k/n 表决门向二元决策图模型的自动转化算法。给出了建立多阶段任务系统二元决策图可靠性模型的 2 个步骤:基于逻辑门转化算法,将部件逻辑关系结构函数转化为阶段二元决策图模型;基于二元决策图布尔操作规则整合阶段二元决策图模型。卫星姿态调整任务的可靠性分析表明:该方法可以有效应用于多阶段任务系统的可靠性分析。

关键词:多阶段任务系统;可靠性建模;二元决策图;数据结构
中图分类号:N945.17　　**文献标志码**:A　　**文章编号**:1001-2486(2017)02-184-09

Binary decision diagram model for reliability analysis of phased mission system

MENG Li[1,2]

(1. College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China;
2. Department of Equipment Testing, Academy of Equipment, Beijing 101416, China)

Abstract: In order to implement the automatic generation of binary decision diagram (BDD) for phased mission system reliability analysis, a universal approach for establishing BDD reliability model was proposed. A nested BDD data structure named BDD_Element was defined, and the approaches of BDD description and storage were given. Algorithms for transforming AND Gate, OR Gate and k/n Gate into BDD models were proposed. The BDD model for reliability analysis of phased mission system was constructed in 2 steps: based on the logic gate transforming algorithms, structure functions of the component logic relationship were transformed into phase BDD models; based on the BDD manipulation rules, phase BDD models were integrated together. By analyzing the reliability of satellite attitude adjustment mission, it shows that this approach can be effectually applied in the reliability analysis of phased mission system.

Key words: phased mission system; reliability modeling; binary decision diagram; data structure

多阶段任务系统由若干连续、非重叠的任务阶段构成,在每个阶段系统以满足不同的任务需求为目标[1]。例如,民航班机的飞行就是典型的多阶段任务系统,一般可划分为滑行和起飞、爬升、巡航、下降、接近和着陆 5 个阶段。多阶段任务系统的可靠性定义为系统在所有任务阶段满足既定功能的能力。与单阶段系统相比,多阶段任务系统可靠性分析必须正确处理部件的跨阶段依赖关系。假设多阶段任务系统在任意阶段失效都会导致整个任务失败,且不考虑部件可靠性参数随时间退化的因素。

针对多阶段任务系统可靠性建模与分析的方法有很多,例如故障树分析(Fault Tree Analysis, FTA)[2]、马尔可夫链(Markov Chain, MC)[3]等。二元决策图(Binary Decision Diagram, BDD)能够高效执行布尔操作,基于 BDD 的多阶段任务系统逻辑表达与可靠性分析高效、直接[4-5]。BDD 是由 Lee[6] 和 Akers[7] 基于香农扩展式(Shannon's decomposition)[5] 提出的布尔函数数据结构。Bryant[4] 在 BDD 模型中加入变量顺序限制规则,提出了顺序二元决策图(Ordered BDD, OBDD)。Mo[1] 总结了基于 OBDD 的多阶段任务系统可靠性建模与分析的一般方法,给出了 4 种变量排序策略。Zang 等[8] 基于 OBDD 提出了一种不可修多阶段任务系统的可靠性建模与分析方法多阶段任务系统-二元决策图(Phased Mission System-

* **收稿日期**:2015-09-28
基金项目:国家自然科学基金资助项目(71071159,71401172)
作者简介:孟礼(1986—),男,辽宁凌海人,讲师,博士,E-mail:mengli0603@163.com

BDD, PMS-BDD),采用阶段代数法处理部件的跨阶段依赖问题。Wang 等[9]建立了可修多阶段任务系统可靠性分析的层次模型,模型底层采用 MC 方法描述可修部件的状态转移行为,模型顶层采用 OBDD 方法描述部件可靠性逻辑关系。

为了统一表示二元布尔操作,Brace 等提出了三元运算符(If-Then-Else operator, ITE operator)[5]。基于 ITE 算子,产生了能够统一描述 BDD 模型和高效执行基本 BDD 布尔操作的数据结构,例如 CUDD[10]。但是,目前还没有提出专门用于多阶段任务系统可靠性分析的 BDD 数据结构,文献[1,8-9,11]中提出的可靠性分析算法只给出了基本的程序步骤。

Meinel 等[5]列举了 10 种二元逻辑关系,包括或(OR)、与(AND)、等价(Equivalence)等。BDD 的布尔逻辑操作运算规则已经广泛应用于 BDD 程序包中。在多阶段任务系统可靠性建模中,经常出现多个部件或部件组由逻辑门连接的情况,在这种情况下,通过递归使用 BDD 布尔操作会消耗大量计算时间。Sinnamon 等[12]提出了基于 BDD 模型的故障树分析方法,有效解决了大型故障树分析中的"组合爆炸"问题。段珊[13]改进了现有的故障树向 BDD 模型转化算法,提出了一种以构建规模最小的 BDD 模型为目标的逻辑相邻有限组合(Logic Neighbor Priory Connection, LNPC)方法,但是该方法只针对包含与门和或门的故障树,未考虑工程实践中常用的 k/n 表决门。Xing 等[14]针对部件无差异的 k/n 表决门提出了一种基于不完全故障覆盖的精确可靠性分析方法,但是目前还没有提出将 k/n 表决门直接转化为 BDD 模型的简便有效方法。

本文针对多阶段任务系统 BDD 可靠性模型的计算机自动生成问题,改进了目前 BDD 程序包中普遍采用的数据结构,定义了专门用于多阶段任务系统可靠性建模与分析的新数据结构 BDD_Element,并给出其描述 BDD 模型的步骤和基于可扩展标记语言(Extensible Markup Language, XML)[15]的存储方式。BDD_Element 既可以描述一个 BDD 节点,也可以描述一个复杂的 BDD 可靠性模型,并且封装了系统中的部件和阶段信息,可以直接用于 BDD 变量排序。分析了与门、或门和 k/n 表决门的 BDD 模型建立过程,给出了逻辑门向 BDD 模型的转化算法。基于每个阶段的部件逻辑关系,建立单阶段 BDD 可靠性模型,将所有单阶段 BDD 模型整合为用于任务可靠性分析的 BDD 模型。为了说明方法的正确性,基于 BDD_Element实现了 PMS-BDD 算法,并以某卫星姿态调整测控任务为例说明了 BDD 可靠性模型构建过程。

1　多阶段任务系统的 BDD 数据结构

1.1　BDD_Element 定义

关于 BDD 模型,采用如下定义:

1)标识为 1 和 0 的没有输出边的节点分别称为 1 节点(ONE node)和 0 节点(ZERO node),统称为吸收节点(sink nodes)[5]。

2)非吸收节点用变量 x_i 标记,且带有 2 条分别标识为 1 和 0 的输出边,这两条边分别称为 1 边(1-edge)和 0 边(0-edge)[5],分别使用实线和虚线表示。

3)只有一个非吸收节点的 BDD 模型称为简单 BDD 模型,具有多个非吸收节点的 BDD 模型称为复杂 BDD 模型。

4)节点 v 的 1 边指向的后继标记为 $high(v)$,0 边指向的后继标记为 $low(v)$[5]。节点的后继既可以是吸收节点或简单 BDD 模型,也可以是复杂 BDD 模型。

ITE 算子是一个三元布尔函数,设输入变量为 x、y、z,则其计算函数 f:如果 x 成立,那么 y 成立,否则 z 成立,定义为:

$$ITE(x,y,z) = x \cdot y + \bar{x} \cdot z \tag{1}$$

从式(1)可以看出,ITE 算子在形式上反映了 BDD 在某个节点处的香农分解过程,因此十分适于描述 BDD 模型。基于 ITE 算子,产生了一种基于索引的 BDD 数据结构[13],并广泛应用于各种 BDD 程序包中,如图 1 所示。

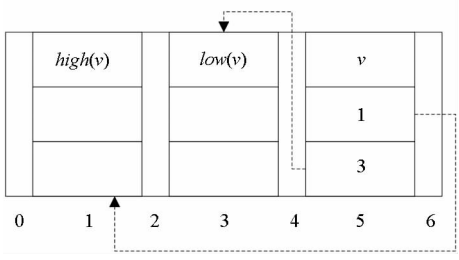

图 1　基于索引的 BDD 数据结构

Fig. 1　BDD data structure based on index

在基于索引的 BDD 数据结构中,BDD 节点被看作基本元素,并存储了变量信息,所有节点以链表的形式连续存储在内存中;节点和后继之间的边用元素索引表示。该数据结构可以高效实现基本的 BDD 布尔操作。但是,基于索引的 BDD 数据结构缺乏描述多阶段任务系统部件跨阶段依赖关系的机制,不能直接用于分析多阶段任务系

统的可靠性问题。

为了自动生成用于多阶段任务系统可靠性分析的 BDD 模型,定义了名为 BDD_Element 的嵌套数据结构,如图 2 所示。

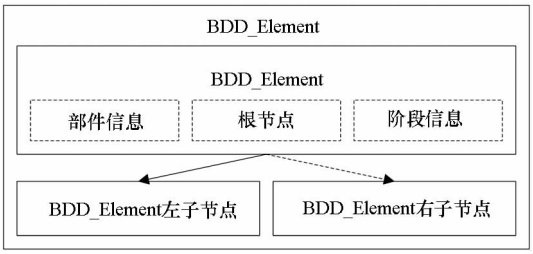

图 2 针对 PMS 的新 BDD 数据结构

Fig. 2 New BDD data structure for PMS

算法 1 给出了详细定义,该数据结构包括 6 个元素:

1) 3 个 BDD_Element 元素:*root*、*high*、*low*。BDD 模型分解为 3 部分:根节点和根节点的两个后继,分别用 *root*、*high* 和 *low* 表示。

2) *component*:记录了 BDD 根节点代表部件的排序信息,便于在可靠性分析算法中进行变量顺序比较。

3) *phase*:记录根节点代表部件的阶段信息,与 *Component* 元素一起用于变量顺序比较。

4) *value*:记录 BDD 构建和可靠性分析过程的计算结果。

算法 1 BDD_Element 定义

Alg. 1 Definition of BDD_Element

```
structure BDD_Element {
    BDD_Element root;
    BDD_Element high;
    BDD_Element low;
    int component;
    int phase;
    double value;
}
```

1.2 运用 BDD_Element 的模型描述方法

例如,图 3 为某多阶段任务系统第 1 阶段的 BDD 可靠性模型 $F^* = a_1 \cdot b_1 + c_1$。以 F^* 为当前目标 BDD,分为 3 个步骤将 F^* 定义为 BDD_Element 结构。

Step 1:将 F^* 的每个节点看作独立的 BDD 模型,分别将它们定义为 BDD_Element。

对于吸收节点 ONE 和 ZERO,元素 *root*、*high* 和 *low* 定义为空;元素 *component* 和 *phase* 定义为 0;元素 *value* 分别定义为 1 和 0。

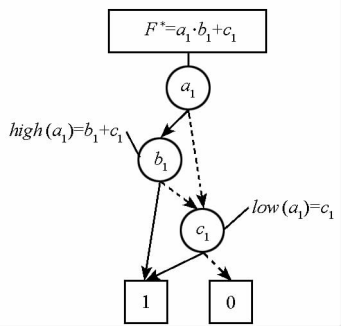

图 3 算例 BDD 模型 F^*

Fig. 3 ABDD example F^*

ONE = structure ("", "", "", 0, 0, 1)

ZERO = structure ("", "", "", 0, 0, 0)

非吸收节点的 *root*、*high* 和 *low* 元素分别定义为空、ONE 和 ZERO。假设变量顺序为 $a < b < c$(其中 $x < y$ 表示变量 x 排在 y 之前),则部件 a、b、c 的 *component* 值分别为 1、2、3。*phase* 值与当前 F^* 的阶段相同。以 b_1 为例,它的 *component* 值为 2,*phase* 值为 1。非吸收节点 *value* 元素的初始值定为 −1。

a1 = structure ("", ONE, ZERO, 1, 1, −1)

b1 = structure ("", ONE, ZERO, 2, 1, −1)

c1 = structure ("", ONE, ZERO, 3, 1, −1)

Step 2:将当前目标 BDD 模型分解为三个部分,分别为根节点 *root* 及其两个后继 *high* 和 *low*,并确定 *root* 的值。

对于 F^*,*root* 元素是 a_1,*high* 和 *low* 分别为以 b_1 和 c_1 为根节点的子 BDD 模型,即 $high(a_1) = b_1 + c_1$,$low(a_1) = c_1$。

Step 3:区分当前 BDD 模型的后继是简单 BDD 模型还是复杂 BDD 模型。若后继是简单 BDD 模型,以 Step 1 中非吸收节点的形式定义该后继;否则,若该后继是复杂 BDD 模型,将其作为当前目标 BDD 模型,然后转到 Step 2。

F^* 的 *high* 后继为以 b_1 为根节点的复杂子 BDD 模型,将其定义为 *temp*,并视为需要基于 BDD_Element 定义的新 BDD 模型。F^* 的 *low* 后继为以 c_1 为根节点的简单 BDD 模型,c_1 也是该后继中唯一的非吸收节点,在这种情况下,将 *low* 直接定义为 c_1。F^* 的最终定义为:

temp = structure (b1, ONE, c1, 2, 1, −1)

F* = structure (a1, temp, c1, 1, 1, −1)

在该定义中,复杂 BDD 的 *component* 和 *phase* 的值与它们的根节点变量相同,便于在任务可靠性计算过程中读取相关信息。

通过嵌套定义的方式,BDD_Element 既可以

描述简单 BDD 模型又可以描述复杂 BDD 模型,因此,对 BDD 模型的描述具有统一性。

1.3　BDD 模型的 XML 存储方法

XML 是一种用嵌套的标签标记数据,并且部分描述计算机应如何处理这些数据的计算机语言[15]。XML 最大的特点是标签的层次嵌套关系十分适用于存储基于 BDD_Element 描述的 BDD 模型。由于 XML 的平台无关性,基于 XML 描述的 BDD 模型可以在不同计算机程序间无障碍传递,具有极强的通用性。

基于 XML 的 BDD_Element 数据结构标签定义为:

< BDD _ Element root ="" high ="" low ="" component ="" phase ="" value ="" / >

该数据标签包括与 BDD_Element 数据结构一一对应的 6 个基本元素。采用标签嵌套结构,基于 XML 描述图 3 中的 BDD 模型 F^* 如算法 2 所示。XML 文档的根节点描述 BDD 模型的根节点 a_1,并记录 a_1 的 2 个后继以及 a_1 的部件和阶段信息。若某节点的 2 个后继均为吸收节点,则描述该节点的标签不再嵌套子标签,例如节点 c_1;否则,该节点标签包含 2 个分别描述其 *high* 和 *low* 后继的子标签,例如节点 a_1 和 b_1。

算法 2　BDD 模型 F^* 的 XML 描述

Alg. 2　Description of F^* based on XML

```
< BDD_Element root = "a1" high = "b1" low = "c1" component = "1" phase = "1" value = " - 1" >
  < BDD_Element root = "b1" high = "1" low = "c1" component = "2" phase = "1" value = " - 1" >
    < BDD_Element root = "" high = "" low = " " component = "0" phase = "0" value = "1"/ >
    < BDD_Element root = "c1" high = "1" low = "0" component = "3" phase = "1" value = " - 1"/ >
  </BDD_Element >
  < BDD_Element root = "c1" high = "1" low = "0" component = "3" phase = "1" value = " - 1"/ >
</BDD_Element >
```

2　逻辑门转化算法

计算机自动构建多阶段任务系统可靠性模型的关键步骤是将各种逻辑门转化为 BDD 模型。分析了 3 种逻辑门:与门、或门和 *k*/*n* 表决门,多数其他类型的逻辑门都可以分解为这 3 种逻辑门。关于与门、或门和 *k*/*n* 表决门,采用如下定义:

1) Conjunction(逻辑“与”,AND)

Conjunction(a, b) = 1 当且仅当 a = 1 并且 b = 1。[5]

即,当且仅当所有由与门连接的事件发生时,顶事件发生。

2) Disjunction (逻辑“或”,OR)

Disjunction(a, b) = 1 当且仅当 a = 1 或者 b = 1。[5]

即,当且仅当不少于一个由逻辑或门连接的事件发生时,顶事件发生。

3) *k*/*n* 表决门

当且仅当不少于 k 个由 *k*/*n* 表决门连接的事件发生时,顶事件发生。

2.1　与(或)门

2.1.1　部件直接由与(或)门连接

假设部件 a 和 b 直接由与门连接,以 a 为顶变量。如果 a 处于工作状态,那么 b 决定任务能否成功完成,所以从 a 出发的 1 边指向 b;如果 a 处于失效状态,那么无论 b 处于什么状态,任务都将失败,所以从 a 出发的 0 边指向吸收节点 ZERO,如图 4 中的模型 f 所示。在这个过程中,部件 b 可以由一组与门连接的部件替代,如图 4 中的模型 g 所示。

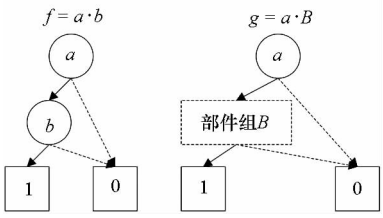

图 4　与门的 BDD 模型

Fig. 4　BDD models of AND Gates

图 5 为与门直接连接部件构成的多阶段任务系统向 BDD 模型的转化算法。假设 $n(n \geq 2)$ 个部件由与门连接,基于 BDD_Element 描述各部件并存储在部件列表 E_List 中。算法的核心思想是以递归的方式将 E_List 中的两个 BDD_Element 合并为一个,直到只剩下一个为止。部件直接由或门连接时可以采用相似方式转化为 BDD 模型。

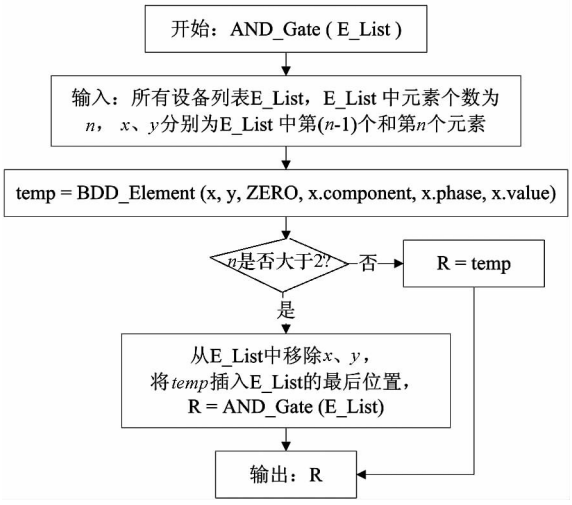

图 5 与门向 BDD 模型转化算法

Fig. 5 Algorithm for transforming AND Gate to BDD model

2.1.2 多组部件由与(或)门连接

在多数情况下,逻辑门的输入不是单个部件。假设两组部件 A 和 B 由与门连接,各组内的部件由任意逻辑门连接。可以想到将 A 组和 B 组部件视为两个 BDD 节点,以图 4 的方式建立 BDD 模型。但是,组 A 的 BDD 模型中表示 A 正常工作的路径不止一条,因此,组 B 的 BDD 模型就需要连接到每一条表示 A 正常工作的路径上,最终导致任务 BDD 模型规模较大,造成存储空间和可靠性计算时间的浪费。对于多组部件由与门和或门连接的情况,可以采用 BDD 基本布尔操作规则构建 BDD 模型。

假设 F 和 G 是两个 BDD 模型:

$F = ITE(x, F_1, F_2), G = ITE(y, G_1, G_2)$。

F 和 G 之间的布尔操作规则为:

$$F \Theta G = ITE(x, F_1, F_2) \Theta ITE(y, G_1, G_2) = \begin{cases} ITE(x, F_1 \Theta G_1, F_2 \Theta G_2) & ind(x) = ind(y) \\ ITE(x, F_1 \Theta G, F_2 \Theta G) & ind(x) < ind(y) \\ ITE(y, F \Theta G_1, F \Theta G_2) & ind(x) > ind(y) \end{cases} \quad (2)$$

其中,Θ 表示任意布尔操作,$ind(x)$ 表示变量 x 的排序。在式(2)中,基于不同的顶变量顺序比较结果,两个 BDD 模型间的布尔操作被分解为各自后继子 BDD 模型间的布尔操作,直到 BDD 模型被分解为 ONE 或 ZERO 为止。基于式(2)中的 BDD 布尔操作规则,提出了使用 BDD_Element 将由与门和或门连接的若干组部件转化为 BDD 模型的算法,如图 6 所示。

算法共有 3 个输入参数:F 和 G 为输入的 BDD 模型,op 表示输入 BDD 模型间的布尔操作。首先确定 F 和 G 的顶变量,并基于排序策略进行变量顺序比较;然后根据跟节点变量排序结果,以

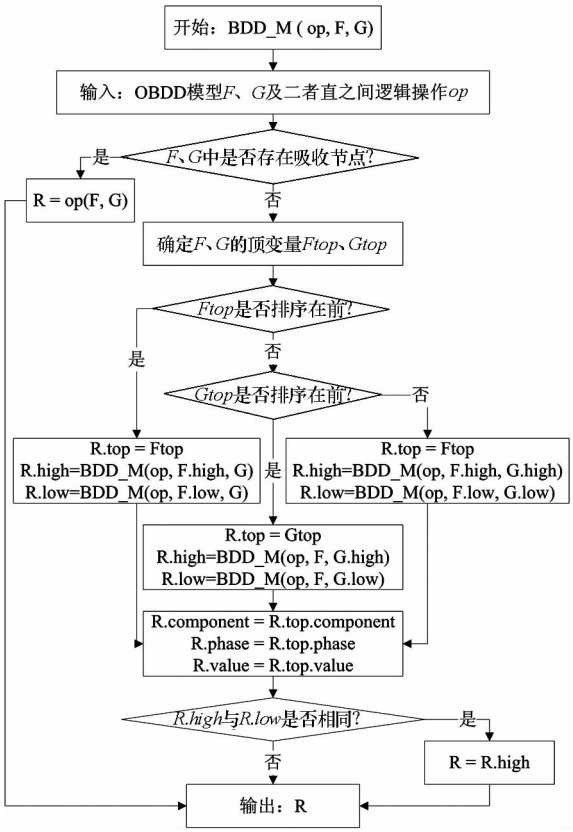

图 6 BDD 布尔操作算法

Fig. 6 Algorithm for BDD Boolean manipulation

递归调用方式执行后继子 BDD 模型间的布尔操作,直到任意子 BDD 模型分解为 ONE 或 ZERO 为止。

BDD_Element 封装了部件及阶段信息,因此可以直接比较两个部件顺序。例如,在采用前向阶段依赖操作(Forward Phased Dependent Operation,FPDO)[1]的情况下,可以通过以下两个条件之一确定 $ind(F) < ind(G)$。

1) $F.component < G.component$;

2) $F.component = G.component$,并且 $F.phase < G.phase$.

2.2 k/n 表决门

提出了一种自顶向下的将 k/n 表决门直接转化为 BDD 模型的转化算法。

假设某系统由 n 个部件组成,部件间由 k/n 表决门连接,要求在至少 $k(1 \leqslant k \leqslant n)$ 个部件正常工作的条件下任务可以成功完成,这样的 k/n 表决门记为 $V(k/n)$。

假设 $V(k/n)$ 中的顶变量为 x,下面考虑两种情况:如果 x 处于正常工作状态,则 $V(k/n)$ 转化为由除 x 以外的由 $n-1$ 个部件组成的 $V((k-1)/(n-1))$ 表决门,且至少需要 $k-1$ 个部件正常工作才能保证任务成功;相反地,如果 x

失效,$V(k/n)$转化为由 $n-1$ 个部件组成的 $V(k/(n-1))$表决门,至少需要 k 个部件工作才能保证任务成功完成。按照这种方式,k/n 表决门可以分解为 2 个子表决门系统,如图 7 所示。

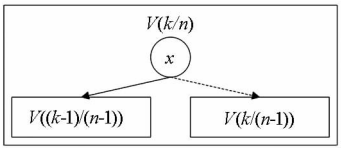

图 7　k/n 表决门分解过程

Fig. 7　Decomposing process of k/n Gate

基于这个过程,提出了将 k/n 表决门转化为 BDD 模型的算法,如图 8 所示。算法有 2 个输入参数:k 表示任务成功完成所需最小工作部件数量,*E_List* 是表决系统包含所有部件的列表。在 $k=1$的情况下,如果 x 处于正常工作状态,无论其他 $n-1$ 个部件处于何种状态,任务都能够成功完成。因此,x 的 *high* 后继直接指向吸收节点 ONE。通过递归调用该算法,图 7 中从 x 分解出来的 2 个子表决系统可以转化为 BDD 模型。

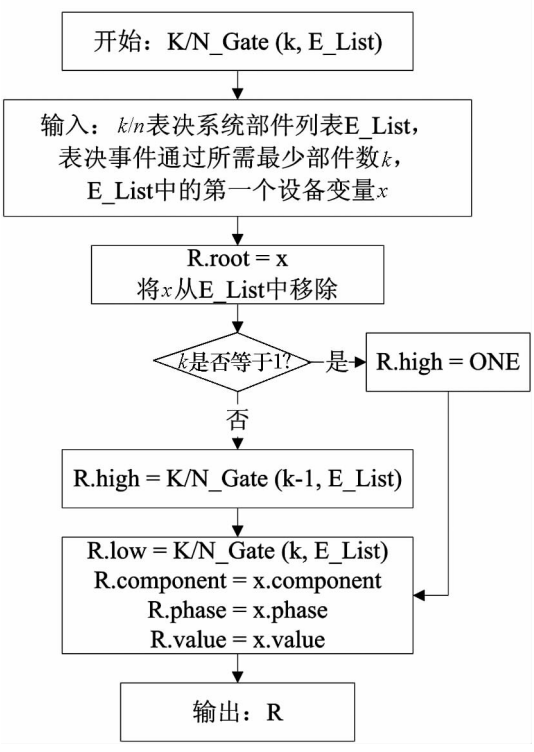

图 8　k/n 表决门向 BDD 模型的转化算法

Fig. 8　Algorithm for transforming k/n Gate to BDD model

3　BDD 模型构建流程

基于 BDD 构建多阶段任务系统可靠性模型一般流程如图 9 所示。首先,读入多阶段任务系统信息,记录部件可靠性参数,并分析各阶段中部件逻辑关系,作为建模过程的输入。然后,采用逻辑门分析算法,建立单阶段 BDD 模型。最后,基于 BDD 布尔操作规则,将所有单阶段 BDD 模型整合为任务 BDD 模型。

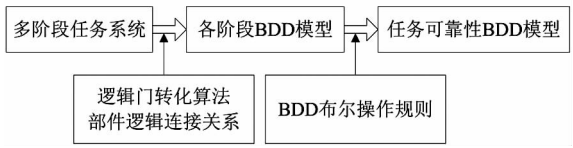

图 9　基于 BDD 的 PMS 可靠性模型建立流程

Fig. 9　BDD construction process for a phased mission

假设某多阶段任务系统 X 不同阶段的部件逻辑结构函数分别为:

阶段 1:$f_1 = a_1 \cdot (b_1 + c_1)$

阶段 2:$f_2 = a_2 \cdot V(2/4)\{b_2, c_2, d_2, e_2\}$

阶段 3:$f_3 = b_3 \cdot (c_3 \cdot d_3 + e_3)$

通过以下 2 个步骤建立多阶段任务系统的 BDD 可靠性模型。

Step 1:根据逻辑门分析算法构建各阶段 BDD 模型。

假设在各阶段内部,部件排序结果为 $a<b<c<d<e<f$。基于给出的逻辑门分析算法,表 2 中各阶段 BDD 可靠性模型建立结果如图 10 所示。

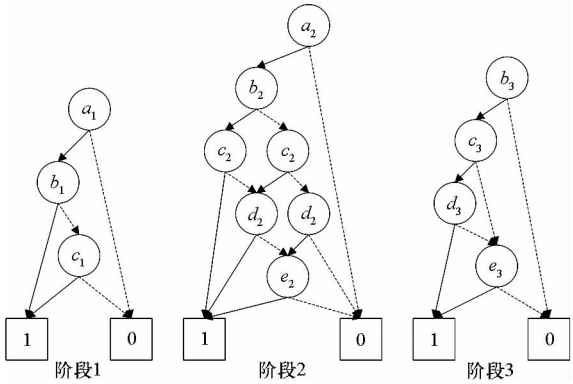

图 10　多阶段任务系统 X 各阶段 BDD 模型

Fig. 10　Phase BDD models of PMS X

Step 2:根据 BDD 布尔操作规则,按阶段间逻辑“与”的关系整合各阶段 BDD 模型。

这里采用 FPDO 排序策略,即按照阶段顺序对部件进行排序。阶段 BDD 模型整合结果如图 11所示。

4　模型验证

PMS-BDD 算法[8]采用阶段代数处理部件跨阶段依赖关系,在工程中得到了广泛的应用,取得了较好的效果。该算法将 BDD 视为一个三元组,通过计算后继节点的不可靠性得到根节点的不可

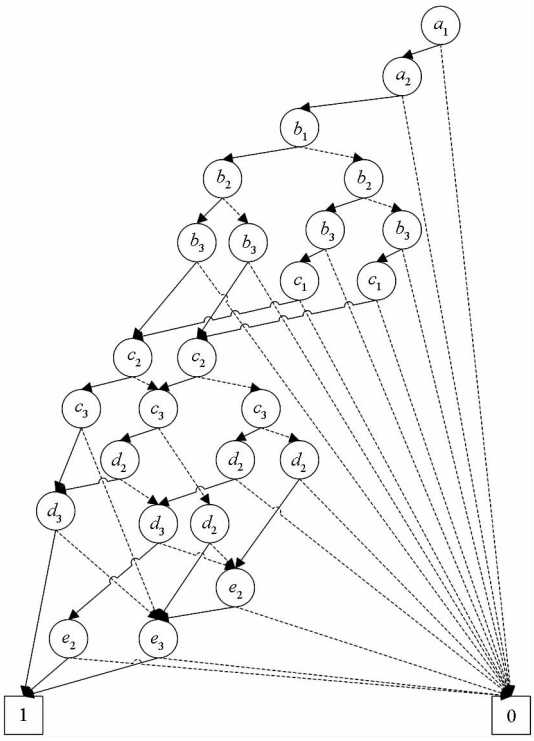

图 11 任务 BDD 模型

Fig. 11 Mission BDD model

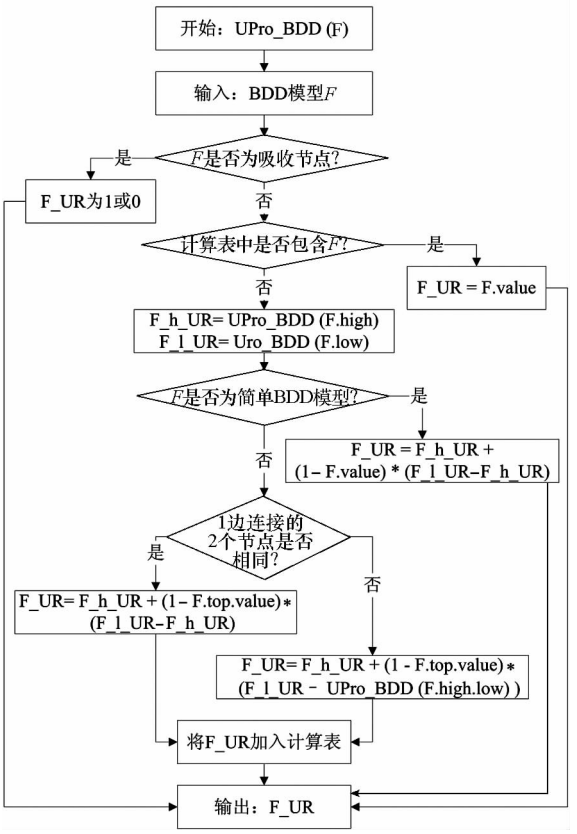

图 12 PMS-BDD 算法实现

Fig. 12 Implementation of PMS-BDD

靠性。采用 BDD_Element 实现该算法如图 12 所示,其中 *F_h_UR*、*F_l_UR* 分别为 *F* 两个后继的不可靠性。

算法的输入是任务 BDD 模型,通过递归调用算法,对模型进行反复分解,直至分解为吸收节点。与文献[8]提供的 PMS-BDD 算法伪代码相比,采用 BDD_Element 具有 3 个优势:①节点的后继可以表示为 BDD_Element 结构,进而作为参数直接传递;②BDD 模型的不可靠性计算数据存储在元素 *value* 中,所有已经计算过的 BDD 模型信息可以在计算表中进行查询,提高了运算效率;③存储在 BDD_Element 中的部件和阶段信息可以直接用于部件顺序比较。通过测试文献[8]中的算例,不可靠性计算结果一致,表明该算法可以正确执行。

假设第 3 节多阶段任务系统 *X* 各阶段分别为[0 s, 100 s]、[100 s, 200 s]和[200 s, 300 s],各部件故障率 λ 见表 1。

根据图 12 中的算法流程,计算多阶段任务系统 X 可靠性结果,见表 2。在阶段转换时刻,任务可靠性存在断点,即阶段 $i(1\leqslant i\leqslant 2)$ 结束时刻任务可靠性比 $i+1$ 阶段起始时刻高。原因是在阶段 i 处于正常状态的部件有可能在阶段 $i+1$ 失效,该情况定义为潜在失效(latent fault)[16]。采用文献[17]的 Petri 网方法,其计算结果与本文一致。

表 1 部件故障率

Tab. 1 Failure rate of each component

	部件				
	a	*b*	*c*	*d*	*e*
$\lambda(\times 10^{-3})$	0.5	0.7	2	1.5	2

表 2 多阶段任务系统 X 的任务可靠性

Tab. 2 Mission reliability of PMS X

阶段	时刻/s	可靠性
阶段 1	0	1
	100 – ①	0.939 572 18
阶段 2	100 + ②	0.939 550 96
	200 –	0.882 038 02
阶段 3	200 +	0.876 870 02
	300	0.761 719 45

注:①“ – ”表示当前阶段的结束时刻;

②“ + ”表示当前阶段的开始时刻。

5 建模流程示例

航天测控系统是航天工程的重要组成部分,在航天器飞行的各个阶段,建立地面与航天器之

间的通信链路,完成对航天器的跟踪、遥测、遥控和数据传输等任务[18-19]。航天测控系统是典型的多阶段任务系统[20]。

假设某卫星的飞行姿态需要调整,该测控任务由 3 个地面站和 1 个指控中心共同完成,如图 13所示。在这个任务中,航天测控系统包括:①指控中心计算机系统 CS;②指控中心与各地面站间链路 L_1,L_2 和 L_3;③各地面站的雷达系统 R_1,R_2 和 R_3;④卫星传感器 S,卫星发动机电力系统 PS。假设各测控设备的状态是相互独立的。

各地面站对该卫星的可视时间窗口分别为 $[t_0, t_4]$、$[t_3, t_6]$和$[t_5, t_9]$,且该任务计划于$[t_1, t_8]$时间段内完成。从图 13 中可以看出,在时段 $[t_3, t_4]$内地面站 1 和地面站 2 同时对卫星可视;在时段$[t_5, t_6]$内地面站 2 和地面站 3 同时对卫星可视。在这两个时段内,两个可视地面站中的任意一个都可以执行该测控任务,所以地面站间为逻辑"或"的关系。根据不同时间涉及的测控部件,卫星姿态调整测控任务被划分为 7 个阶段,见表 3。

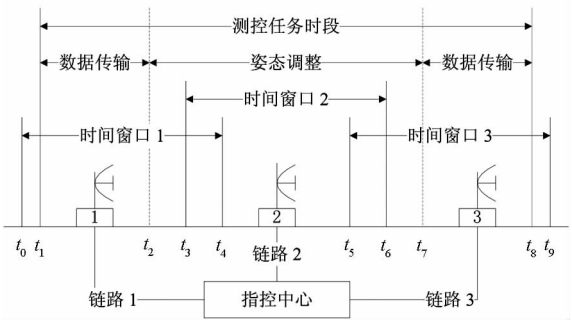

图 13　卫星姿态调整任务

Fig. 13　Satellite attitude adjustment mission

表 3　卫星姿态调整测控任务阶段配置

Tab. 3　Phase configuration of satellite attitude adjustment mission

阶段	可靠性结构函数	时段	测控事件
1	$f_1 = CS \cdot L_1 \cdot R_1 \cdot S$	$[t_1, t_2]$	指控中心通过地面站雷达向卫星上传姿态调整数据,卫星传感器接收数据
2	$f_2 = CS \cdot L_1 \cdot R_1 \cdot S \cdot PS$	$[t_2, t_3]$	卫星发动机电力系统启动,卫星处于姿态调整状态;同时,地面站雷达对卫星保持跟踪,测量卫星飞行状态数据,接收卫星下传数据,并通过通信链路将数据发送给指控中心
3	$f_3 = CS \cdot (L_1R_1 + L_2R_2) \cdot S \cdot PS$	$[t_3, t_4]$	
4	$f_4 = CS \cdot L_2 \cdot R_2 \cdot S \cdot PS$	$[t_4, t_5]$	
5	$f_5 = CS \cdot (L_2R_2 + L_3R_3) \cdot S \cdot PS$	$[t_5, t_6]$	
6	$f_6 = CS \cdot L_3 \cdot R_3 \cdot S \cdot PS$	$[t_6, t_7]$	
7	$f_7 = CS \cdot L_3 \cdot R_3 \cdot S$	$[t_7, t_8]$	卫星传感器向地面站下传飞行数据,地面站将接收的数据发送给指控中心

卫星姿态调整测控任务的可靠性计算算法流程如图 14 所示。

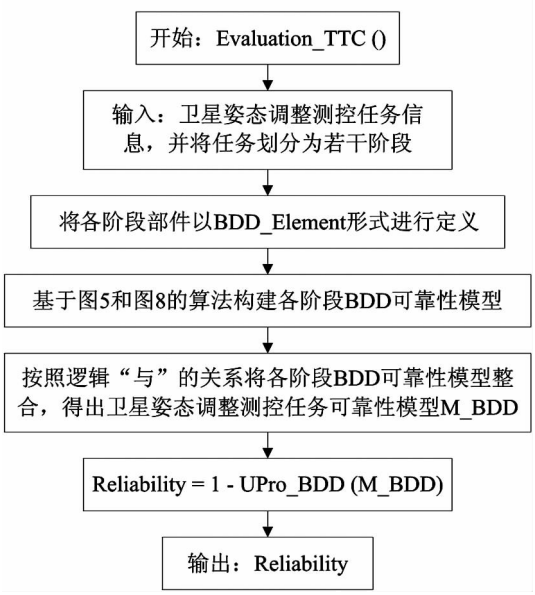

图 14　任务可靠性计算算法

Fig. 14　Algorithm for evaluating mission reliability

假设卫星姿态调整测控任务各阶段时刻和部件可靠性参数分别见表 4 和表 5,卫星飞行姿态调整测控任务可靠性计算结果为0. 995 864 13。

表 4　阶段直起止时刻

Tab. 4　Duration of each phase

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
时刻/s	0	10	50	120	280	350	410	420

表 5　部件可靠性参数

表 5　Reliability parameters of each component

	CS	L_1	L_2	L_3	R_1
MTBF①/h	2000	200	200	200	100
	R_2	R_3	S	PS	
MTBF/h	100	100	1000	50	

注:①MTBF 为 Mean time between failure,平均故障间隔时间。

6 结论

针对多阶段任务系统可靠性分析的模型输入问题,提出了一种基于 BDD 的可靠性模型计算机自动构建方法。共分为 3 个步骤:①将各阶段的所有部件定义为 BDD 节点形式,根据阶段可靠性逻辑关系分析结果和逻辑门转化算法,建立各阶段 BDD 可靠性模型;②基于 BDD 布尔操作规则,按照逻辑"与"的关系将阶段 BDD 模型整合为最终的任务 BDD 模型;③基于 BDD 理论,计算多阶段任务系统的可靠性。该自动建模方法的基础是新定义的 BDD 数据结构 BDD_Element,在该数据结构中,嵌套定义了 3 个基本元素用于统一描述 BDD 模型的根节点及其 2 个后继,同时封装了多阶段任务系统的部件和阶段信息,可以直接用于变量顺序比较。给出了基于 BDD_Element 的 BDD 模型描述和存储方法。基于 BDD_Element,提出了将与门、或门和 k/n 表决门转化为 BDD 模型的算法,给出了两个 BDD 间进行布尔操作的算法,基于这些算法能够完成 BDD 可靠性模型的自动构建步骤。为了说明方法的正确性,实现并验证了工程中广泛应用的 PMS-BDD 算法,并以航天测控系统为例,计算了某卫星姿态调整测控任务的可靠性。

参考文献(References)

[1] Mo Y C. New insights into the BDD-based reliability analysis of phased-mission systems [J]. IEEE Transaction on Reliability, 2009, 58(4): 667 – 678.

[2] Esary J D, Ziehms H. Reliability analysis of phased missions[J]. International Journal of Reliability Quality & Safety Engineering, 1974, 02(4): 213 – 236.

[3] Alam M, Al-Saggaf U M. Quantitative reliability evaluation of repairable phased-mission systems using Markov approach [J]. IEEE Transaction on Reliability, 1986, 35(5): 498 – 503.

[4] Bryant E R. Graph based algorithms for Boolean function manipulation[J]. IEEE Transaction on Computers, 1986, 35(8): 677 – 691.

[5] Meinel C, Theobald T. Algorithms and data structures in VLSI design: BDD foundations and applications [M]. Germany: Springer Berlin Heidelberg, 1998.

[6] Lee C Y. Representation of switching circuits by binary decision programs [J]. Bell System Technology Journal, 1959, 38(4): 985 – 999.

[7] Akers B. Binary decision diagrams[J]. IEEE Transaction on Compute, 1978, C – 27(6): 509 – 516.

[8] Zang X Y, Sun H R, Trivedi K S. A BDD-based algorithm for reliability analysis of phased-mission systems[J]. IEEE Transaction on Reliability, 1999, 48(1): 50 – 60.

[9] Wang D Z, Trivedi K S. Reliability analysis of phased-mission system with independent component repairs [J]. IEEE Transaction on Reliability, 2007, 56(3): 540 – 551.

[10] Somenzi F. CUDD: CU decision diagram package release [CP/OL]. (2012 – 2 – 2) [2015 – 8 – 20]. http://vlsi.colorado.edu/ ~ fabio/.

[11] Meng L, Wang G, Wu X Y. Mission reliability analysis of TT&C system using BDD algorithm [C]//Proceedings of the 7th International Conference on Mathematical Methods in Reliability: Theory, Methods, Application, 2011: 682 – 686.

[12] Sinnamon R M, Andrews J D. New approaches to evaluating fault trees [J]. Reliability Engineering and System Safety, 1997, 58(3): 89 – 96.

[13] 段珊. 二元决策图的排序优化及故障树转化方法的研究[D]. 长沙: 中南大学, 2008.
DUAN Shan. Research on ordering optimization of BDD and fault tree transform algorithm[D]. Changsha: Central South University, 2008. (in Chinese)

[14] Xing L D, Amari S V, Wang C N. Reliability of k-out-of-n systems with phased-mission requirements and imperfect fault coverage [J]. Reliability Engineering and System Safety, 2012, 103: 45 – 55.

[15] 左伟明. 完全掌握 XML 基础概念、核心技术与典型案例[M]. 北京: 人民邮电出版社, 2010.
ZUO Weiming. Perfect yourself in the basic concepts, core technologies and typical cases of XML[M]. Beijing: Posts & Telecom Press, 2010. (in Chinese)

[16] Somani A K. Simplified phased-mission system analysis for systems with independent component repairs[J]. International Journal of Reliability Quality and Safety Engineering, 1997, 4: 167 – 189.

[17] 杨晓松, 武小悦. 航天测控系统任务可靠性分析的 EOOPN 模型[J]. 国防科技大学学报, 2013, 35(5): 37 – 43.
YANG Xiaosong, WU Xiaoyue. Mission reliability analysis of space TT&C system using extended objected Petri net model[J]. Journal of National University of Defense Technology, 2013, 35(5): 37 – 43. (in Chinese)

[18] 夏南银. 航天测控系统[M]. 北京: 国防工业出版社, 2002.
XIA Nanyin. Spacetracking, telemetry and command system[M]. Beijing: National Defense Industry Press, 2002. (in Chinese)

[19] Demircioglu E, Nefes M M. Reliability-based TT&C subsystem design methodology for complex spacecraft missions [C]// Proceedings of IEEE Information Sciences and Systems Conference, Princeton, 2008: 1268 – 1272.

[20] 闫华, 武小悦. 航天测控通信系统可靠性分析的 Krylov 子空间投影算法 [J]. 国防科技大学学报, 2012, 34(4): 63 – 67.
YAN Hua, WU Xiaoyue. Krylov subspace projection algorithm of reliability analysis of TT&C and communication system [J]. Journal of National University of Defense Technology, 2012, 34(4): 63 – 67. (in Chinese)