doi:10.11887/j.cn.201705026 http://journal.nudt.edu.cn

多阶段任务系统可靠性分析的ROBDD算法*

闫 华[1],邢云燕[2],张 立[1],郭立卿[1]
(1. 中国人民解放军陆军勤务学院 军事物流系,重庆 401311;
2. 国防科技大学 国家安全与军事战略研究中心,湖南 长沙 410073)

摘 要:针对Markov方法分析多阶段任务系统(Phased-Mission System, PMS)可靠性时的状态空间爆炸问题,基于层次化建模思想,建立了PMS任务可靠性的顶层系统二维决策图(Binary Decision Diagram, BDD)模型和底层部件Markov模型。通过分析BDD中的同构节点和冗余节点,提出顶层模型构造过程中的同构节点合并策略和冗余节点删除策略。利用上述节点压缩策略生成简化模型,提高模型构造和存储效率。基于PMS部件排序规则,给出了层次化模型的递归求解方法,该方法的计算复杂度与顶层模型中的节点总数呈线性关系。通过算例分析,对比采用节点压缩策略前后的模型节点数,以及层次化方法与Markov方法的计算结果,验证了简化层次模型的正确性和有效性。

关键词:任务可靠性;层次化模型;多阶段任务系统
中图分类号:TP202.1 **文献标志码**:A **文章编号**:1001-2486(2017)05-164-07

ROBDD algorithm for analyzing the reliability of phased-mission system

YAN Hua[1], *XING Yunyan*[2], *ZHANG Li*[1], *GUO Liqing*[1]
(1. Department of Military Logistics, Army Logistics University of PLA, Chongqing 401311, China;
2. Center for National Security and Strategic Studies, National University of Defense Technology, Changsha 410073, China)

Abstract: The reliability analysis approach of PMS (phased-mission system) based on Markov process encounters the state explosion problem for large systems. To avoid the state-explosion problem, the PMS reliability model using hierarchical method was proposed. The hierarchical method uses a low-level continuous time Markov chain and an upper-level binary decision diagrams. The isomorphic nodes consolidation strategy and the redundant nodes removal strategy were given to construct reduced hierarchical model. The above nodes compressing strategies can improve the efficiency of model generation and storage. A recursive computing algorithm solving the hierarchical model was designed on the basis of the PMS components ordering, and its computational complexity was linear with the number of nodes in model. Finally, by a case study, the number of nodes in the original model was compared with the proposed model using nodes compressing strategy, also the mission reliability in the hierarchical model was compared with it in the Markov model. Results show the correctness and effectiveness of the reduced hierarchical model.

Key words: mission reliability; hierarchical model; phased-mission system

多阶段任务系统(Phased-Mission System, PMS)是指在任务执行过程中,系统配置、任务成败标准及部件可靠性参数等会随任务要求、环境压力等发生变化,根据系统参数是否发生变化可将任务过程划分为多个时间连续的阶段[1-4]。PMS在实际工程应用中较为常见,特别是一些关键应用中,如航天和军用装备领域[5-6],其任务可靠性对于系统分析和改进、系统可靠性分配[7]等具有重要作用。

PMS的任务可靠性分析方法主要包括三类:二元决策图(Binary Decision Diagram, BDD)方法[8-9]、仿真方法[10-11]和Markov方法[12-13]。BDD方法利用可靠性框图或故障树建立系统可靠性模型,并将模型转换为二元决策图进行计算。BDD方法高效简洁,但是无法处理部件的动态修复行为;仿真方法能够处理复杂的系统行为,但要得到可靠的计算结果,仿真次数通常要达到百万级甚至千万级,因此仿真计算量巨大是其面临的主要问题。

相比BDD方法和仿真方法,Markov方法能够处理可修系统,且能够计算得到系统可靠性的解析解。但是,随着系统中部件数量的增多,

* 收稿日期:2016-04-06
基金项目:国家自然科学基金资助项目(71401172,71401170)
作者简介:闫华(1983—),男,甘肃临潭人,讲师,博士,E-mail:yanhua_8304@163.com

Markov 方法面临状态空间爆炸问题。针对该问题,Wang 等[4]提出了一种层次化建模方法,通过组合运用 BDD 方法和 Markov 方法减小模型的复杂度,但在建立层次化模型时,如果不对 BDD 节点进行缩减处理,在最坏的情况下,其节点数与部件数目仍呈指数关系[14];闫华等[15]研究了基于 krylov 子空间投影的 Markov 可靠性模型求解算法,通过空间投影提高计算效率,但是投影方法仍然需要生成初始问题的状态空间,且子空间规模太小会造成计算误差增大;Lu 等[16]提出了一种基于 Markov 的可靠性分析方法,通过对不同阶段的部件行为向量进行组合,搜索代表系统任务成功的所有路径,算法的计算复杂度与路径数量呈线性关系,但是该方法中路径总数随系统部件数和阶段数的增长而迅速增加。

基于简化有序二元决策图(Reduced Ordered BDD, ROBDD)的 PMS 可靠性分析算法主要思想是借鉴层次化建模方法,建立顶层系统行为的 BDD 模型和底层部件行为的 Markov 模型。对于顶层系统级模型,本文根据 ROBDD 的定义,提出相应的节点压缩策略,同时结合递归计算思想给出模型的可靠性求解算法。利用上述方法,实现模型简化和计算效率提高。

1　PMS 任务可靠性分析的层次化模型

1.1　层次化建模基本方法

层次化建模中的一个重要假设是:对于可修部件 c,一旦其在阶段 i 内某个时刻失效,则其在整个阶段内都处于失效状态;即使 c 在其后某个时刻被修复,也只能在下一阶段参与系统任务。这是因为在顶层系统模型中,部件在某个阶段 i 的状态是由 BDD 方法进行描述的,而 BDD 是静态方法,无法处理部件的动态行为,某一阶段的修复只能在下个阶段发生作用。

Wang 等[4]提出了 PMS 可靠性分析中建立层次化模型的基本方法。首先,设部件 c 的状态空间为$\{1,2,\cdots,o^{(c)},\cdots,f^{(c)}\}$,其中 $1,2,\cdots,o^{(c)}$ 为正常状态,$o^{(c)}+1,o^{(c)}+2,\cdots,f^{(c)}$ 为失效状态。定义矩阵 $\boldsymbol{Q}^{(c)}$ 和 $\boldsymbol{P}^{(c)}$ 如下:

$$\boldsymbol{Q}^{(c)}=\begin{pmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12}\\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22}\end{pmatrix}$$

$$\boldsymbol{P}^{(c)}=\begin{pmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12}\\ 0 & 0\end{pmatrix}$$

其中,$\boldsymbol{Q}^{(c)}$ 为部件 c 的转移速率矩阵,$\boldsymbol{P}^{(c)}$ 为考虑吸收态情况下的部件转移速率矩阵,子矩阵 $\boldsymbol{Q}_{11}$ 为 $o^{(c)}$ 阶矩阵。

根据 $\boldsymbol{Q}^{(c)}$ 和 $\boldsymbol{P}^{(c)}$,定义矩阵 $\boldsymbol{E}^{(c)}$、$\boldsymbol{U}^{(c)}$ 和 $\boldsymbol{D}^{(c)}$:

$$\begin{cases}\boldsymbol{E}^{(c)}=\mathrm{e}^{\boldsymbol{Q}^{(c)}T}\\ \boldsymbol{U}^{(c)}=\mathrm{e}^{\boldsymbol{P}^{(c)}T}\cdot\begin{pmatrix}\boldsymbol{I}_{o^{(c)}\times o^{(c)}} & 0\\ 0 & 0\end{pmatrix}\\ \boldsymbol{D}^{(c)}=\boldsymbol{E}^{(c)}-\boldsymbol{U}^{(c)}\end{cases}\tag{1}$$

其中,矩阵 $\boldsymbol{E}^{(c)}$ 为部件 c 在时间 T 内的正常转移概率矩阵,矩阵 $\boldsymbol{U}^{(c)}$ 为部件 c 在时间 T 内一直处于正常状态的转移概率矩阵,矩阵 $\boldsymbol{D}^{(c)}$ 为部件 c 在时间 T 内发生故障的转移概率矩阵。

以上为部件在某一阶段内的状态变化过程。对于部件跨阶段的依赖性,由于假设各部件的失效率和修复率是相互独立的,因此,仅以单个部件 c 为例进行说明。为区分在不同任务阶段的部件,定义阶段部件 c_i 表示第 i 个阶段中的部件 c,c_i 的状态为 $s_i^{(c)}$。为简化分析,假设部件只有正常与失效两种状态,由此可知 $s_i^{(c)}$ 为一布尔变量。同时,令状态 x 表示阶段部件 c_i 无论正常或者失效都不会影响系统任务的成败。综上,定义 $s_i^{(c)}$ 如下:

$$s_i^{(c)}=\begin{cases}1 & \text{部件 } c \text{ 在阶段 } i \text{ 处于正常}\\ 0 & \text{部件 } c \text{ 在阶段 } i \text{ 处于失效}\\ x & \text{部件 } c \text{ 在阶段 } i \text{ 的状态不影响任务成败}\end{cases}\tag{2}$$

根据部件状态 $s_i^{(c)}$,定义矩阵 $\boldsymbol{X}^{(c_i)}$:

$$\boldsymbol{X}^{(c_i)}=\begin{cases}\boldsymbol{U}^{(c_i)} & s_i^{(c)}=1\\ \boldsymbol{D}^{(c_i)} & s_i^{(c)}=0\\ \boldsymbol{E}^{(c_i)} & s_i^{(c)}=x\end{cases}\tag{3}$$

令 $\boldsymbol{v}(c_i)$ 表示部件 c 在第 i 个阶段末的状态概率向量,可得:

$$\boldsymbol{v}(c_i)=\boldsymbol{v}(c_{i-1})\cdot\boldsymbol{X}^{(c_i)}\tag{4}$$

设 PMS 共有 p 个阶段,通过迭代计算可推得该部件在第 p 个阶段末的状态概率向量为:

$$\boldsymbol{v}(c_p)=\boldsymbol{v}(c_0)\cdot\prod_{i=1}^{p}\boldsymbol{X}^{(c_i)}\tag{5}$$

式中,$\boldsymbol{v}(c_0)$ 为部件 c 初始时刻的状态概率向量,$\boldsymbol{X}^{(c_i)}$ 的取值由 $s_i^{(c)}$ 决定。部件在系统各个阶段的状态构成了一组状态序列,记为 Ω,则部件 c 对应的 Ω 可表示为 $\Omega_c=\{s_1^{(c)},s_2^{(c)},\cdots,s_i^{(c)},\cdots,s_p^{(c)}\}$。令 $\mathbf{1}^{\mathrm{T}}$ 表示单位列向量,进一步可推得部件 c 在状态序列 Ω_c 下的概率值为:

$$Pr\{c_i=s_i^{(c)}(1\leqslant i\leqslant p)\}\mid_{\Omega_{ab}}=\boldsymbol{v}(c_p)\cdot\mathbf{1}^{\mathrm{T}}\tag{6}$$

对于多个部件的情形,假设系统由两个部件 a 和 b 组成,各阶段部件所对应的一组状态序列为 $\Omega_{ab}=\{s_1^{(a)},s_2^{(a)},\cdots,s_p^{(a)};s_1^{(b)},s_2^{(b)},\cdots,s_p^{(b)}\}$。可

得该状态序列下的部件状态概率向量为：

$$\boldsymbol{v}(b_p)\big|_{\Omega_{ab}} = \boldsymbol{v}(a_p)\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\prod_{i=1}^{p}\boldsymbol{X}^{(b_i)} = \boldsymbol{v}(a_0)\cdot\prod_{i=1}^{p}\boldsymbol{X}^{(a_i)}\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\prod_{i=1}^{p}\boldsymbol{X}^{(b_i)} \tag{7}$$

以及该状态序列下的相应概率值为：

$$\begin{aligned}&Pr\{a_i = s_i^{(a)}(1\leqslant i\leqslant p), b_i = s_i^{(b)}(1\leqslant i\leqslant p)\}\big|_{\Omega_{ab}}\\ &= Pr\{a_i = s_i^{(a)}(1\leqslant i\leqslant p)\}\cdot Pr\{b_i = s_i^{(b)}(1\leqslant i\leqslant p)\}\\ &= (\boldsymbol{v}(a_p)\cdot\mathbf{1}^{\mathrm{T}})\cdot(\boldsymbol{v}(b_p)\cdot\mathbf{1}^{\mathrm{T}})\end{aligned} \tag{8}$$

式(7)和式(8)可以推广到 n 个部件的情形。

利用式(7)和式(8)，采用 BDD 方法建立系统顶层模型，搜索模型中对应任务成功的所有状态序列，然后计算每个状态序列下的概率值并依次相加，结果即为 PMS 的任务可靠性。

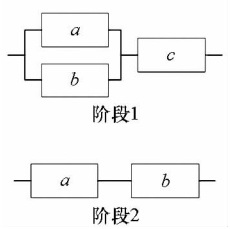
1.2 基于节点压缩策略的模型简化

BDD 是一个具有有限个节点的有向无环图，包含非终结节点和终结节点。终结节点没有输出边，包括 0 节点和 1 节点，分别表示布尔假和真。若用 BDD 表示 PMS 系统的逻辑结构函数，则 0 节点代表系统任务失败，1 节点代表任务成功。用 BDD 分析 PMS 时，非终结节点由 PMS 中的阶段部件状态表示，其 0 边和 1 边分别表示该阶段部件的状态取 0 和 1 时的分支。

在构造 BDD 之前，如果给定一个变量顺序，使得 BDD 中任何一条从根节点到终结节点的路径中，各节点的先后顺序与给定的变量顺序一致，由此规则生成的 BDD 称为有序 BDD(ordered BDD)[17]。模型中采用如下的排序策略：将 PMS 中属于同一部件的状态变量分为一组，每组内部的阶段部件状态按阶段顺序由后至前排列，不同部件之间的顺序任意。以图 1 中的 PMS 为例，各阶段可靠性框图如图 1(a)所示，上述排序策略下生成的 BDD 如图 1(b)所示。

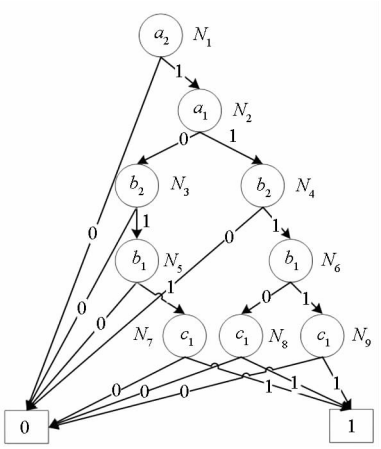
令 $a<b$ 表示部件顺序为 a 先于 b，则图中阶段部件顺序为：$a_2<a_1<b_2<b_1<c_1$。BDD 中的节点总数和计算效率与其变量顺序密切相关[18]，上述排序策略中，在最坏的情况下，总节点数为 2^M-1，M为系统中的阶段部件数量。可见，如果不对 BDD 节点采取任何压缩策略，在系统部件数量较多的情况下，生成的 BDD 依然十分庞大，相应的存储和计算必将耗费大量空间和时间资源。因此，借鉴 ROBDD 思想，提出了层次化模型生成过程中的节点压缩策略。

ROBDD 是指，一个 BDD 在其所有的路径上，



(a) 各阶段可靠性框图

(a) Reliability block diagram of each mission phase



(b) 给定排序策略下的 BDD 模型

(b) BDD model under specifying ordering strategy

图 1 示例 PMS 各阶段可靠性框图和 BDD 模型

Fig. 1 Reliability block diagrams of each phase and BDD model of the example PMS

变量都遵循给定的变量次序 $x_1<x_2<\cdots<x_n$，并且满足：

1)不存在同构节点。同构节点是指两个节点具有相同的变量名和左右子节点。若以 N 表示节点，$var(N)$表示节点变量，$low(N)$和$high(N)$分别表示其左右子节点，当满足式(9)所示条件时，称 N_i 和 N_j 为同构节点，即 $N_i=N_j$。

$$\begin{cases}var(N_i)=var(N_j)\\ low(N_i)=low(N_j)\\ high(N_i)=high(N_j)\end{cases} \tag{9}$$

2)不存在冗余节点。冗余节点是指节点的左右子节点为同构节点，即 $low(N)=high(N)$。当消除同构节点后，表现为节点的左右子节点指向同一节点。

PMS 任务可靠性分析时，对于顶层 BDD 模型中的同构节点，只保留一个；冗余节点则直接从 BDD 中删除。但上述处理方式需要先生成初始 BDD，再根据条件 1 和 2 对模型进行简化，处理过程十分低效。因此，在 BDD 生成过程中，提出两种节点压缩策略。

第一种：同构节点合并策略。令 $\Gamma=\{N_1,$

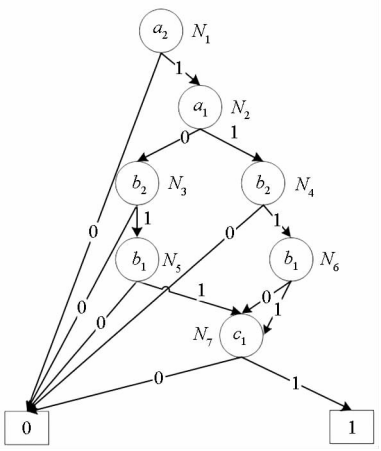
$N_2,\cdots,N_i,\cdots,N_A\}$ 表示已生成节点集合,Γ 中不存在同构节点。令 N_{temp} 表示新节点,若 N_{temp} 满足条件——$\forall N_j \in \Gamma$,均有 $N_j \neq N_{\text{temp}}$,则将其加入集合 Γ 中;否则,如果集合 Γ 中存在 $N_j = N_{\text{temp}}$,即表示 N_j 和 N_{temp} 满足式(9) 中所列条件,为同构节点。此时,用同构节点 N_j 代替 N_{temp} 即可。上述策略可以总结为:每次生成新节点时,将新节点存储在临时空间 N_{temp} 中,判断 N_{temp} 是否与已生成节点集合 Γ 中的元素相同,若全不相同,则将 N_{temp} 加入集合 Γ 中;否则,以集合 Γ 中的同构节点替代 N_{temp}。可知,由于以上过程是在 BDD 生成中动态进行,因此,能够有效提高 BDD 构造的执行效率。在具体实现过程中,以 Hash 表存储所有已生成的节点,能够进一步提高节点的搜索效率。

第二种:冗余节点删除策略。由定义可知,冗余节点的左右子节点指向相同的节点,这也意味着该节点状态为 0 或 1,都不会对系统任务产生影响。因此,冗余节点即对应阶段部件状态 x。PMS 中,阶段部件状态是否为 x,取决于系统的结构函数及其他部件的状态值。根据冗余节点,定义依赖性冗余部件如下:

设有两部件 a 和 b,其结构函数为 $\Phi = a \diamond b$,操作 $\diamond$ 为逻辑与或者逻辑或,对应部件的并联或串联。若当 $s^{(a)} = 1$ 或 $s^{(a)} = 0$ 时,无须 $s^{(b)}$ 即可确定 Φ 值,即 $s^{(b)} = x$,则称部件 b 为 $s^{(a)} = 1$ 或 $s^{(a)} = 0$时的依赖性冗余部件,因为部件 b 是否冗余依赖于部件 a 的状态。根据该定义,存在两种情况:①若 $\Phi = a + b$,则当 $s^{(a)} = 1$ 时,$\Phi = 1$,部件 b 为 $s^{(a)} = 1$ 时的依赖性冗余部件;反之,a 为 $s^{(b)} = 1$时的依赖性冗余部件;②若 $\Phi = a \cdot b$,则当 $s^{(a)} = 0$ 时,$\Phi = 0$,部件 b 为 $s^{(a)} = 0$ 的依赖性冗余部件;反之,a 为 $s^{(b)} = 0$ 时的依赖性冗余部件。该定义可以扩展至多个部件或多个模块组合,即当 $\Phi = a \diamond b \diamond \cdots \diamond n$ 或者 $\Phi = \Phi_1 \diamond \Phi_2 \diamond \cdots \diamond \Phi_n$ 时,由逻辑与和逻辑或的运算性质可知,上述定义仍然成立。

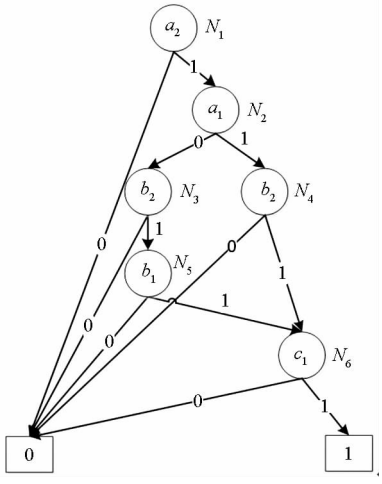
由于 BDD 生成过程是一个自顶向下的赋值和自底向上的构造过程,根据上述依赖性冗余部件的定义,可以在 BDD 自顶向下的赋值过程中,根据系统结构函数和已赋值部件状态,判断后续部件是否属于依赖性冗余部件。如果是,可知当前路径中,该部件对应节点为冗余节点,可直接跳过该部件节点的构造,进入后续处理过程,从而有效节省 BDD 的构造时间和存储空间。

以图 1 中的 PMS 为例,利用同构节点合并策略和冗余节点删除策略后得到的 BDD 分别如图 2(a)和图 2(b)所示。由图 1(b)可以看出,该 BDD 中节点 N_7、N_8、N_9 为同构节点,根据同构节点合并策略,只保留节点 N_7,得到图 2(a);同时,图 2(a)中节点 N_6 满足 $low(N_6) = high(N_6) = N_7$,即 N_6 为冗余节点,根据冗余节点删除策略,删除节点 N_6 得到图 2(b)。其中,图 2(b)即为最终生成的简化层次模型。



(a) 合并同构节点后的 BDD 模型

(a) BDD model after combining isomorphic nodes



(b) 删除冗余节点后的 BDD 模型

(b) BDD model after deleting redundancy nodes

图 2　采取节点压缩策略后得到的简化 BDD 模型

Fig. 2　Simplified BDD model using nodes compressing strategies

2　模型递归求解方法

令 N_z 表示系统 BDD 模型中的节点,$low(N_z) = N_x$,$high(N_z) = N_y$,$\Phi(N_z)$ 为该节点的逻辑结构函数。节点 N_z 对应的阶段部件及其状态变量分别为 c_i 和 $s_i^{(c)}$,$\boldsymbol{w}(N_z)$ 表示当 $\Phi(N_z) = 1$ 时节点对应部件的状态概率向量。定义任一满足 $\Phi(N_z) = 1$ 的阶段部件状态序列为结构函数 $\Phi(N_z)$ 的任务

成功序列,并令集合 $\Omega(N_z)$ 表示 $\Phi(N_z)$ 下的所有任务成功序列集合。$|\Omega(N_z)|$ 为集合中元素数量,则有:

$$\boldsymbol{w}(N_z) = \sum_{l}^{|\Omega(N_z)|} \boldsymbol{v}(c_i)\big|_{\Omega(N_z)_l} \tag{10}$$

由于 BDD 中的每个非终节点有且仅有左右两个子节点,可知:

$$\Omega(N_z) = \{\Omega(N_x), s_i^{(c)} = 0\} \cup \{\Omega(N_y), s_i^{(c)} = 1\}$$

若令 $\Omega' = \{\Omega(N_x), s_i^{(c)} = 0\}$, $\Omega'' = \{\Omega(N_y), s_i^{(c)} = 1\}$, 则根据式(4)可得:

$$\begin{aligned}\boldsymbol{w}(N_z) &= \sum_{j}^{|\Omega'|} \boldsymbol{v}(c_i)\Big|_{\Omega'_j} + \sum_{k}^{|\Omega''|} \boldsymbol{v}(c_i)\Big|_{\Omega''_k} \\ &= \Big(\sum_{l}^{|\Omega(N_x)|} \boldsymbol{v}(c_{i-1})\Big|_{\Omega(N_x)_l}\Big)\cdot \boldsymbol{D}^{(c_i)} + \Big(\sum_{m}^{|\Omega(N_y)|} \boldsymbol{v}(c_{i-1})t\big|_{\Omega(N_y)_m}\Big)\cdot \boldsymbol{U}^{(c_i)} \\ &= \boldsymbol{w}(N_x)\cdot \boldsymbol{D}^{(c_i)} + \boldsymbol{w}(N_y)\cdot \boldsymbol{U}^{(c_i)}\end{aligned} \tag{11}$$

式(11)中假定节点 N_z 的左右子节点 N_x、N_y 对应的阶段部件为 c_{i-1}。实际上,由于任务成功序列 $\Omega(N_z)_l$ 中包含不同部件在不同阶段的状态变量,父节点与子节点不一定对应同一部件;同时,由于模型中删除了冗余节点,父节点与子节点对应的阶段部件并不一定是按照给定顺序依次排列。因此,根据节点排序策略,对式(11)进行修正后得到:

1)若节点 N_z 与子节点为同一部件,记子节点对应阶段部件为 c_j,则有:

$$\begin{aligned}\boldsymbol{w}(N_z) = {} & \boldsymbol{w}(N_x)\cdot\Big(\prod_{k=j+1}^{i-1}\boldsymbol{E}^{(c_k)}\Big)\cdot \boldsymbol{D}^{(c_i)} + \\ & \boldsymbol{w}(N_y)\cdot\Big(\prod_{k=j+1}^{i-1}\boldsymbol{E}^{(c_k)}\Big)\cdot \boldsymbol{U}^{(c_i)}\end{aligned} \tag{12}$$

2)若节点 N_z 与子节点为不同部件,则根据式(7)可得:

$$\begin{aligned}\boldsymbol{w}(N_z) = {} & \boldsymbol{w}(N_x)\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(c_0)\cdot\Big(\prod_{k=1}^{i-1}\boldsymbol{E}^{(c_k)}\Big)\cdot \boldsymbol{D}^{(c_i)} + \\ & \boldsymbol{w}(N_y)\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(c_0)\cdot\Big(\prod_{k=1}^{i-1}\boldsymbol{E}^{(c_k)}\Big)\cdot \boldsymbol{U}^{(c_i)}\end{aligned} \tag{13}$$

利用式(12)和式(13),可自底向上依次对各节点进行计算,终结节点 0 和 1 对应的状态概率向量设为 $\mathbf{0}$ 和$(1,0,\cdots,0)$。设根节点为 N_{root},令 R 表示系统的任务可靠性,则可得:

$$R(\mathrm{PMS}) = \boldsymbol{w}(N_{\mathrm{root}})\cdot\mathbf{1}^{\mathrm{T}} \tag{14}$$

以图 2(b)中生成的 BDD 为例,各节点对应的 $\boldsymbol{w}(N_i)$ 如表 1 所示。其中

$$q_1 = \boldsymbol{v}(c_0)\cdot\boldsymbol{U}^{(c_1)}\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\boldsymbol{U}^{(b_1)}\cdot\boldsymbol{U}^{(b_2)}\cdot\mathbf{1}^{\mathrm{T}}$$

$$q_2 = \boldsymbol{v}(c_0)\cdot\boldsymbol{U}^{(c_1)}\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\boldsymbol{E}^{(b_1)}\cdot\boldsymbol{U}^{(b_2)}\cdot\mathbf{1}^{\mathrm{T}}$$

利用上述方法计算任务可靠性时,各节点的状态概率向量 $\boldsymbol{w}(N_i)$ 只需计算一次,以后每次遍历到该节点,取得已有结果进行计算即可。设最终生成的 ROBDD 中节点总数为 θ,则该策略下任务可靠性的计算复杂度为 $O(|\theta|)$。

表 1 示例 BDD 中各节点的对应 $\boldsymbol{w}(N_i)$ 值

Tab. 1 The $\boldsymbol{w}(N_i)$ of each node in the example BDD

节点	$\boldsymbol{w}(N_i)$
N_1	$\boldsymbol{w}(N_1)=\boldsymbol{v}(a_0)\cdot(q_1\cdot\boldsymbol{D}^{(a_1)}+q_2\cdot\boldsymbol{U}^{(a_1)})\cdot\boldsymbol{U}^{(a_2)}$
N_2	$\boldsymbol{w}(N_2)=q_1\cdot\boldsymbol{v}(a_0)\cdot\boldsymbol{D}^{(a_1)}+q_2\cdot\boldsymbol{v}(a_0)\cdot\boldsymbol{U}^{(a_1)}$
N_3	$\boldsymbol{w}(N_3)=\boldsymbol{v}(c_0)\cdot\boldsymbol{U}^{(c_1)}\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\boldsymbol{U}^{(b_1)}\cdot\boldsymbol{U}^{(b_2)}$
N_4	$\boldsymbol{w}(N_4)=\boldsymbol{v}(c_0)\cdot\boldsymbol{U}^{(c_1)}\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\boldsymbol{E}^{(b_1)}\cdot\boldsymbol{U}^{(b_2)}$
N_5	$\boldsymbol{w}(N_5)=\boldsymbol{v}(c_0)\cdot\boldsymbol{U}^{(c_1)}\cdot\mathbf{1}^{\mathrm{T}}\cdot\boldsymbol{v}(b_0)\cdot\boldsymbol{U}^{(b_1)}$
N_6	$\boldsymbol{w}(N_6)=\boldsymbol{v}(c_0)\cdot\boldsymbol{U}^{(c_1)}$

3 算例分析

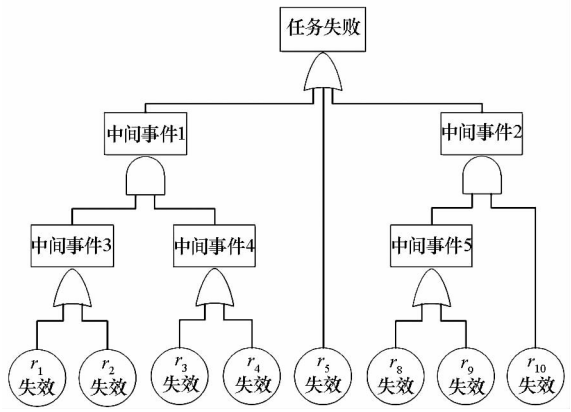
以图 3 所示的 PMS 系统为例,图中列出了系统各阶段的故障树。记系统任务为 T,共包含 3 个任务阶段,分别为 p_1, p_2 和 p_3,并假设一旦任务在某个阶段失败,则整个任务失败;各阶段持续时间分别为 $t_1=90$ s, $t_2=150$ s 和 $t_3=90$ s。假设系统中各部件的平均修复时间(Mean Time To Repair, MTTR)和平均故障间隔时间(Mean Time Between Failures, MTBF)均为 30 min 和 300 min。

将任务 T 中阶段 p_1 构成的单阶段系统任务记为 T_1,由阶段 p_1 和 p_2 组成的 PMS 记为任务 T_2。分别建立任务 T_1, T_2 和 T 的层次化模型,对比在不进行压缩和采用节点压缩策略下,上述各模型中的 BDD 节点数量,其结果如表 2 所示。

由表 2 可以看出,采用节点压缩策略后,顶层 BDD 模型中的节点数量明显减少。若记压缩率为:(压缩前节点数 - 压缩后节点数) / 压缩前节点数,则对于任务 T_1,T_2 和 T,采用节点压缩策略后,顶层 BDD 模型的压缩率分别达到了 72.22%,99.09% 和 99.98%。

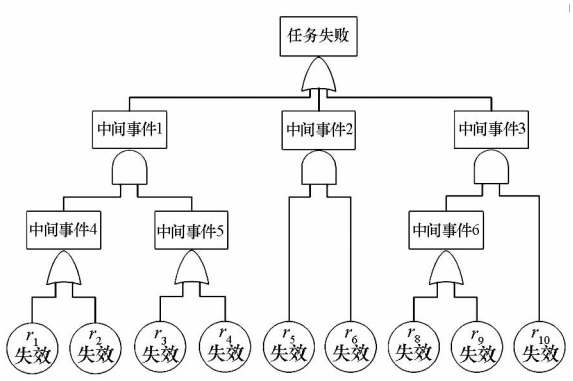
根据任务 T_1 和 T_2 的阶段组成可知,$R(T_1)$ 和 $R(T_2)$ 即为任务 T 在第一阶段末和第二阶段末的任务可靠性。利用层次化模型的递归求解算法,可计算得到 $R(T_1)$、$R(T_2)$ 和 $R(T)$,并与 Markov 方法的计算结果进行对比,两种方法的可靠性结果如表 3 所示。

由表 3 可以看出,层次化方法的可靠性计算结果略低于 Markov 方法。其中,任务 T 的计算误差最大,约为 4.46E - 05。可见,随着系统运行时间的增加,层次化方法的计算误差逐步增大。出现计算误差是由于假设部件在阶段某个时刻失效,则即使部件被修复,在该阶段内部件也将一直



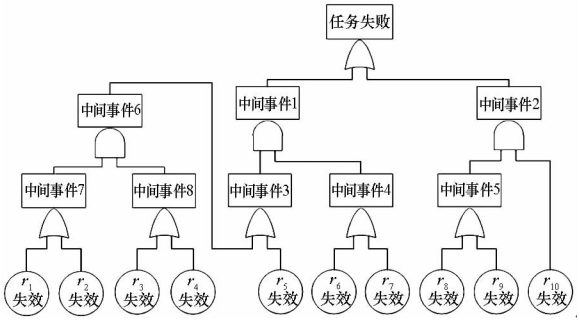
(a) 阶段 1 故障树

(a) Fault failure tree of phase 1



(b) 阶段 2 故障树

(b) Fault failure tree of phase 2



(c) 阶段 3 故障树

(c) Fault failure tree of phase 3

图 3　算例 PMS 中各阶段故障树

Fig. 3　Fault failure trees of each phase in example PMS

表 2　各任务可靠性模型中的 BDD 节点数量对比

Tab. 2　Comparison of the number of BDD nodes of each mission under different strategies

任务	不进行压缩	采用节点压缩策略	压缩率/%
T_1	36	10	72.22
T_2	3298	30	99.09
T	852 637	144	99.98

表 3　层次化方法与 Markov 方法计算结果比较

Tab. 3　Reliability under hierarchical method compared with the Markov method

任务	层次化方法	Markov 方法	结果误差
T_1	0.994 864	0.994 867	2.42E－06
T_2	0.993 920	0.993 954	3.39E－05
T	0.993 457	0.993 501	4.46E－05

处于失效状态,直到下一阶段初始时刻。因此,误差的产生与系统在各个阶段的任务持续时间有关,当系统各阶段的运行时间越小于部件的 MTTR 时,由上述假设导致的计算误差就越小。当误差较大时,一种可行的解决方法就是将原 PMS 划分为更多阶段,使部件修复后能够尽早参与任务执行,从而减小计算误差。但是,阶段数增多的同时也会导致层次化模型中的 BDD 节点数迅速增加,模型求解的计算复杂度增大。因此,在实际计算中,应根据具体的精度要求,对系统任务阶段进行合理划分。

4　结论

利用 BDD 和 Markov 相结合的层次化建模方法,建立了 PMS 任务可靠性的层次化模型;通过提出的同构节点合并策略和冗余节点删除策略,实现了模型规模的简化;给出了基于层次化模型计算任务可靠性的递归求解方法,实现了模型运算的简化。最后,通过算例验证了节点压缩策略和模型求解算法的正确性和有效性。算例中假设各任务阶段之间是串联关系,实际上简化层次方法能够处理任务阶段为复杂逻辑组合时的情况,只要根据阶段间的逻辑组合需求,生成相应的顶层 BDD 模型;同时,由提出的层次化模型求解过程可知,该算法复杂度只与模型中的节点数有关,各任务阶段为串并联或混联关系并不能直接决定模型中的节点数量。因此,提出的递归求解算法同样适用于并联和混联任务系统。层次化方法中,BDD 无法处理系统的动态行为,导致其可靠性计算结果略低于 Markov 方法,解决方法是在初始 PMS 基础上,划分更多的任务阶段,使部件修复后尽早生效。

参考文献(References)

[1] Esary J D, Ziehms H. Reliability analysis of phased missions[J]. International Journal of Reliability Quality & Safety Engineering, 1975, 2(4): 213－236.

[2] Burdick G R, Fussell J B, Rasmuson D M, et al. Phased mission analysis: a review of new developments and an

application[J]. IEEE Transactions on Reliability, 1977, R-26(1): 43-49.

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

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

[5] Wu X Y, Yan H, Li L R. Numerical method for reliability analysis of phased-mission system using Markov chains[J]. Communications in Statistics-Theory and Methods, 2012, 41(21): 3960-3973.

[6] 郭继周, 赵朝先, 郭波. 面向多阶段任务的武器系统备件优化配置建模[J]. 数学的实践与认识, 2009, 39(12): 64-70.
GUO Jizhou, ZHAO Chaoxian, GUO Bo. Spare optimization modeling of phased-mission system for air-defense combat unit[J]. Mathematics in Practice and Theory, 2009, 39(12): 64-70. (in Chinese)

[7] 张根保, 柳剑, 王国强. 基于任务的数控机床模糊可靠性分配方法[J]. 计算机集成制造系统, 2012, 18(4): 768-774.
ZHANG Genbao, LIU Jian, WANG Guoqiang. Fuzzy reliability allocation of CNC machine tools based on task[J]. Computer Integrated Manufacturing Systems, 2012, 18(4): 768-774. (in Chinese)

[8] Xing L D, Levitin G. BDD-based reliability evaluation of phased-mission systems with internal/external common-cause failures[J]. Reliability Engineering & System Safety, 2013, 112: 145-153.

[9] 陈光宇, 张小民, 唐小我. 基于分离BDD的通用多阶段任务系统可靠性分析[J]. 系统工程理论与实践, 2013, 33(5): 1240-1246.
CHEN Guangyu, ZHANG Xiaomin, TANG Xiaowo. Analysis of generalized phased-mission systems reliability based on separated binary decision diagram[J]. System Engineering—Theory & Practice, 2013, 33(5): 1240-1246. (in Chinese)

[10] Xu S W, Wu X Y. Simulation method for reliability of TT&C mission with high redundancy and small time horizon[J]. Journal of Systems Engineering and Electronics, 2012, 23(6): 943-948.

[11] 陈志诚, 姚路, 杨建军. 可修多阶段任务系统可靠性仿真研究[J]. 现代制造工程, 2013(11): 126-128,42.
CHEN Zhicheng, YAO Lu, YANG Jianjun. Reliability simulation research on repairable phased-mission system[J]. Modern Manufacturing Engineering, 2013(11): 126-128, 42. (in Chinese)

[12] Yang C H, Yang J J, Bian J L. The PMS modeling method based on interactive Markov chain[C]//Proceedings of the 9th Conference on Reliability, Maintainability and Safety, 2011: 349-352.

[13] Wu X Y, Hillston J. Mission reliability of semi-Markov systems under generalized operational time requirements[J]. Reliability Engineering & System Safety, 2015, 140: 122-129.

[14] 张旭. BDD实现方法及改进方法的研究[D]. 兰州: 兰州大学, 2005.
ZHANG Xu. Research on construction method and improving method of BDD[D]. Lanzhou: Lanzhou University, 2005. (in Chinese)

[15] 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 for reliability analysis of TT&C and communication system[J]. Journal of National University of Defense Technology, 2012, 34(4): 63-67. (in Chinese)

[16] Lu J M, Wu X Y. Reliability evaluation of generalized phased-mission systems with repairable components[J]. Reliability Engineering & System Safety, 2014, 121: 136-145.

[17] Randal E B. Graph-based algorithms for Boolean function manipulation[J]. IEEE Transactions on Computers, 1986, 35(8): 677-691.

[18] 王楠, 杜素果. 一种多阶段任务系统的BDD排序新方法[J]. 科学技术与工程, 2010, 10(17): 4217-4224.
WANG Nan, DU Suguo. A novel binary decision diagram variable ordering approach on phased mission system[J]. Science Technology and Engineering, 2010, 10(17): 4217-4224. (in Chinese)