

# 一种基于赋时 Petri 网和 ZBDD 的装配序列规划方法

李凤英<sup>1,2</sup> 古天龙<sup>2</sup> 常亮<sup>2</sup> 徐周波<sup>2</sup>

(西安电子科技大学电子工程学院 西安 710071)<sup>1</sup>

(桂林电子科技大学计算机科学与工程学院 桂林 541004)<sup>2</sup>

**摘要** 赋时 Petri 网为装配序列规划提供了有效的建模方法,但在求解最优装配序列时受到组合复杂性的严重制约。零压缩二叉决策图(ZBDD)是处理大规模组合集合和 0-1 稀疏向量的一种有效符号技术,能够有效缓解组合爆炸问题。将赋时 Petri 网与 ZBDD 结合起来,给出了一种求解装配序列最优解的有效方法。首先通过转换算法将赋时 Petri 网转换为等价的普通 Petri 网,接下来给出普通 Petri 网可达状态及迁移引发函数的 ZBDD 表示方法,最后基于 ZBDD 给出最优装配序列求解算法。实例验证表明,该算法在求解过程中通过隐式符号操作实现了 Petri 网的可达状态搜索,有效缓解了计算过程中的组合复杂性。

**关键词** 赋时 Petri 网,装配序列规划,零压缩二叉决策图

**中图分类号** TP391.9 **文献标识码** A

## Timed Petri and ZBDD Based Approach for Assembly Sequence Planning

LI Feng-ying<sup>1,2</sup> GU Tian-long<sup>2</sup> CHANG Liang<sup>2</sup> XU Zhou-bo<sup>2</sup>

(School of Electronic Engineering, Xidian University, Xi'an 710071, China)<sup>1</sup>

(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)<sup>2</sup>

**Abstract** Timed Petri net is one of efficient modeling formalisms for assembly sequence optimization problems, but the state combination complexity renders it hard and even impossible for large scale assembly sequence optimization problems to be solved. Zero-suppressed binary decision diagram(ZBDD) is an efficient symbolic technique to represent and manipulate the combination sets and 0-1 sparse vectors. An algorithm for selecting the best assembly sequence was given out based on timed Petri net and ZBDD. Firstly, the timed Petri net model of assembly was transformed into equivalent ordinary Petri net using a transformation algorithm. And then an approach of representing the reachable states set and transition firing function of Petri net by ZBDD was given out. Henceforth, a symbolic ZBDD based algorithm for selecting the best assembly sequence was developed. The experimental tests give the proof that the ZBDD based algorithm alleviates the computing complexity by using implicitly symbolic manipulation to search reachable states of Petri net.

**Keywords** Timed Petri nets, Assembly sequence planning, Zero-suppressed binary decision diagram

装配序列规划是 CAD/CAM、数字化产品、并行工程、虚拟制造等领域的一个重要研究课题,对指导产品可装配性设计、提高产品装配质量和降低装配成本具有重要意义<sup>[1,2]</sup>。

近年来,研究者给出了适合各种装配环境的装配序列表示方法<sup>[3,4]</sup>,如优先约束、建立条件等隐式表示法,有向图、与或图、Petri 网、装配顺序表等显示枚举表示法。利用各种表示方法,提出了不同的装配序列生成方法,如交互式问答方法<sup>[5]</sup>、割集生成方法<sup>[6]</sup>、基于符号技术的生成方法<sup>[7]</sup>等。应用以上各种方法已经可以生成完备的可行装配序列。但如何有效地求解出最优的装配序列,仍然是亟需解决的问题。

针对最优装配序列的求解问题,很多研究人员利用遗传算法和蚁群算法研究了最优装配序列选择问题<sup>[8,9]</sup>。遗传算

法利用种群在适应度函数约束下的不断进化,最终得到最优或近优解,规划结果受初始序列种群影响较大。蚁群算法靠信息素积累得到最优或近优解,运行过程中信息素积累过慢,限制了搜索效率的提高。

赋时 Petri 网是一种理想的离散系统建模工具,可以用位置表示装配体的零部件,用迁移表示装配操作。将装配操作的代价蕴含在迁移的赋时中,既能清晰、直观地表示所有可行装配序列,又提供了选择最优装配序列的比较标准。但最优装配序列的选择问题受到状态组合爆炸的影响。减缓或者某种程度上避免状态组合复杂性问题的一种可行策略是:状态的符号或隐式描述。有序二叉决策图(Ordered Binary Decision Diagram, OBDD)则是迄今为止最为有效的一种符号技

到稿日期:2011-03-10 返修日期:2011-05-21 本文受国家自然科学基金项目(60563005,60243002,61063002),广西可信软件重点实验室开放基金资助。

李凤英(1974—),女,博士生,主要研究方向为 Petri 网、符号调度技术和符号模型检验等,E-mail: lfy@guet.edu.cn;古天龙(1964—),男,教授,博士生导师,主要研究方向为形式化技术、符号计算和知识工程等;常亮(1980—),男,副教授,主要研究方向为描述逻辑、语义 Web 服务等;徐周波(1976—),女,博士生,主要研究方向为符号调度技术、符号模型检验等。

术<sup>[10]</sup>。零压缩二叉决策图(ZBDD, Zero-Suppressed Binary Decision Diagrams)是OBDD的一种扩展形式<sup>[11]</sup>,克服了OBDD表示组合集合的复杂性,对Petri网可达状态空间的计算和表示效率更高。所以本文采用赋时Petri网为可行装配序列表建模,通过符号ZBDD技术求解最优装配序列。

## 1 预备知识

### 1.1 赋时Petri网

**定义1** 赋时Petri网形式化定义为一个五元组  $TPN=(P, T; F, M_0, \Gamma)=(PN, \Gamma)$ 。其中,  $PN=(P, T; F, M_0)$  是普通Petri网,  $P=\{p_1, p_2, \dots, p_n\}$  为位置集合,  $T=\{t_1, t_2, \dots, t_m\}$  为迁移集合( $P \cap T = \emptyset, P \cup T \neq \emptyset$ ),  $F \subseteq (P \times T) \cup (T \times P)$  为流关系(关系弧)。  $M_0$  为初始标识向量, 一个标识向量是一个函数  $M: P \rightarrow \mathbf{Z}^+ \cup \{0\}$ , 表示位置中托肯的分布情况;  $\Gamma: T \rightarrow \mathbf{Z}(P \rightarrow \mathbf{Z})$  是迁移(位置)赋时函数。

具有迁移(位置)赋时函数的赋时Petri网,称为赋时迁移(位置)Petri网;具有时间因子的迁移(位置)称为赋时迁移(位置)。

对于  $\forall t \in T, \forall p \in P$ , 令  $\cdot t = \{p | (p, t) \in F\}, t \cdot = \{p | (t, p) \in F\}, \cdot p = \{t | (t, p) \in F\}, p \cdot = \{t | (p, t) \in F\}$ , 称  $\cdot t(\cdot p)$  为迁移  $t$ (位置  $p$ )的前集或输入集,  $t \cdot(p \cdot)$  为迁移  $t$ (位置  $p$ )的后集或输出集。

如果  $\forall p_i \in \cdot t$  都有  $M(p_i) \geq 1$ , 且  $\forall p_j \in t \cdot$  有  $M(p_j) = 0$ , 则称迁移  $t$  是使能的或者具有引发权。在标识  $M$  下, 使能迁移  $t$  一旦引发, 就到达一个新的标识  $M'$  (记为  $M[t > M']$ ), 将标识  $M$  通过引发迁移所能到达的标识集(标识向量构成的集合)记为  $[M]$ 。

若  $\forall M \in [M_0]: M(p) \leq k (k \in \mathbf{N})$ , 称  $p$  是  $k$ -有界的。如果  $\forall p \in P, p$  都是有界的, 则称该Petri网是有界Petri网。

如果一个普通Petri网是1-有界的, 则称该Petri网是安全Petri网。

### 1.2 零压缩二叉决策图

二叉决策图(BDD)是表示从  $\{0, 1\}^n$  到  $\{0, 1\}$  的布尔函数  $f(x_1, x_2, \dots, x_n)$  的有向无环图, 由二叉树化简得到。给定变量序  $\pi$  (变量  $x_1, x_2, \dots, x_n$  是有顺序的), 通过如下化简规则可以得到简化的有序二叉决策图, 简称有序二叉决策图(OBDD), 它为布尔函数  $f(x_1, x_2, \dots, x_n)$  的表示提供规范型。

**规则1(S-删除规则)** 删除所有两条边指向同一节点的冗余节点。

**规则2(合并规则)** 共享所有等价的子图。

零压缩二叉决策图(ZBDD)是基于以下化简规则的BDD的特殊形式。

**规则1'(pD-删除规则)** 删除所有1-边指向0-终节点的节点。

**规则2'(合并规则)** 与OBDD一样, 共享所有等价的子图。

以图1(a)所示的一个Petri网为例, 其可达状态如图1(b)所示。图2(a)为其可达状态空间的OBDD表示, 需要13个节点。同样的可达状态空间用ZBDD表示, 需要8个节点, 如图2(b)所示。ZBDD的紧凑表示得益于ZBDD使用了与OBDD不同的化简规则, 即pD-删除规则, 使ZBDD能更简洁地表示组合集合, 尤其是稀疏组合集合<sup>[12]</sup>。

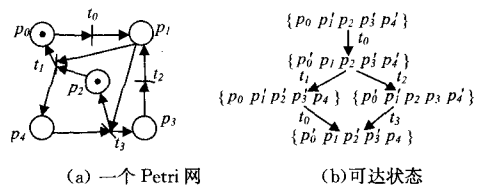


图1 一个简单Petri网及其可达状态

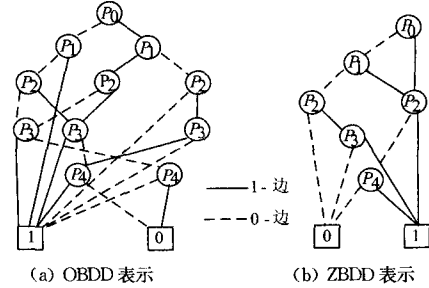


图2 Petri网可达状态空间的符号表示

## 2 装配序列的Petri网模型

装配序列的赋时Petri网模型可从装配序列的与/或图模型转换得到。从与/或图到赋时Petri网的转换算法描述如下。

**算法1** 给定一个具有  $R$  个零件的装配体与/或图模型, 通过以下步骤构造装配体的赋时Petri网模型。

- (1) 将与/或图的每个节点转换为Petri网的一个位置, 其中每个与/或节点  $i$  对应一个位置  $p_i$ , 表示子装配体, 叶节点  $k$  对应位置  $p_k$ , 表示零件,  $k=1, \dots, R$ 。
- (2) 将从节点  $i$  到节点  $l$  和  $m$  的与分支转换为一个迁移  $t_j$ , 令  $\cdot t_j = \{p_i, p_m\}$  且  $t_j \cdot = \{p_l\}$ 。
- (3) 创建弧  $(p_l, t_j)$ ,  $(p_m, t_j)$  和  $(t_j, p_i)$ 。
- (4) 将装配代价分配给相应的迁移。
- (5) 分配一个初始标识  $M_0 = \prod p_k, k=1, \dots, R$ , 表示每个零件都准备就绪。

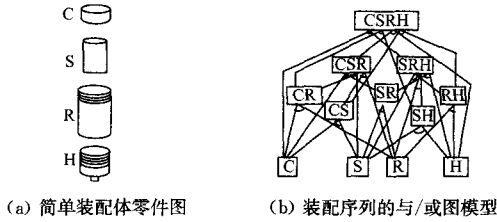
为迁移分配装配代价时可综合多种因素, 包括操作时间、所需的资源成本、装配操作类型、稳定性、装配优先关系和约束等。本文为迁移分配装配代价时参照了文献[13], 如表1所列。主要考虑了两个方面的因素: 一个因素是装配操作的类型, 拧紧操作代价为4, 插入操作代价为2, 安放操作代价为1, 主要与装配时间、装配工具和装配条件相关; 另一个因素是处理子装配体的难度, 这与装配体的自由度成正比, 一个子装配体的自由度越大, 就越不稳定, 越难处理。

表1 简单装配体的装配代价分配

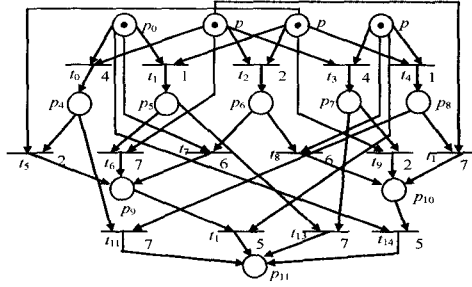
迁移	操作复杂度	子装体自由度	总代价	迁移	操作复杂度	子装体自由度	总代价
0	4	0	4	8	4	2	6
1	1	0	1	9	2	0	2
2	2	0	2	10	4	3	7
3	4	0	4	11	4	3	7
4	1	0	1	12	4	1	5
5	2	0	2	13	4	3	7
6	4	3	7	14	4	1	5
7	4	2	6				

以图3(a)给出的简单装配体为例, 可以由图3(b)所示的装配序列与/或图模型, 利用算法1得到等价的赋时Petri网模型, 如图3(c)所示。图3(c)中的位置  $p_0, p_1, p_2$  和  $p_3$  分别

表示零件  $H, R, S$  和  $C$ ,  $p_0 - p_3$  中的拖肯表示零件  $H, R, S$  和  $C$  处于就绪状态; 位置  $p_4 - p_{10}$  分别表示子装配体  $HR, HS, RS, RC, SC, HRS$  和  $RSC$ ,  $p_{11}$  表示装配产品  $HRSC$ , 其中位置中没有拖肯, 表示还没有得到该(子)装配体; 迁移  $t_0 - t_{14}$  表示装配操作, 迁移旁边的数字表示该装配操作所需的代价。



(a) 简单装配体零件图 (b) 装配序列的与/或图模型



(c) 装配序列的赋时 Petri 网模型

图 3 简单装配体及其装配序列模型

装配序列的 Petri 网模型也可以从低层过程约束建立, 文献[14]给出了根据装配的静态信息建立可行装配序列的 Petri 网模型的方法, 根据该方法可以得到所有可行装配序列的基本 Petri 网表示。根据表 1 给出的装配代价分配方案, 为每个代表装配操作的迁移赋时, 就得到了装配序列的赋时 Petri 网模型, 如图 3(c) 所示。

在给定变量序下, ZBDD 能为  $(0, 1)^n$  到  $(0, 1)$  的布尔函数提供唯一的符号表示。为了能用 ZBDD 技术求解基于赋时 Petri 网模型下的装配序列规划问题, 给出了赋时 Petri 网到普通 Petri 网的转换算法(具体描述见算法 2), 将装配序列的赋时 Petri 网模型转换为等价的普通 Petri 网模型。这样, 装配代价被转化在迁移的个数上而无须再单独考虑, 为 ZBDD 的编码提供了便利。同时, 装配序列可用迁移序列表示, 迁移序列中迁移的个数(并发的迁移看成是一个)即为装配代价, 从而简化了装配序列规划算法的实现。

**算法 2** 给定一个具有  $n$  个位置和  $m$  个迁移的赋时 Petri 网  $TPN=(P, T; F, K, \Gamma)$ , 通过以下步骤转换为等价的具有  $N$  个位置和  $M$  个迁移的普通 Petri 网  $PN=(P', T'; F', K')$ 。

输入: 赋时 Petri 网  $TPN=(P, T; F, K, \Gamma)$   
 输出: 普通 Petri 网  $PN=(P', T'; F', K')$   
 Trans\_TPN( $TPN=(P, T; F, K, M_0, \Gamma)$ ) {  
      $P'=P; T'=T; F'=F; K'=K;$   
     for( $i=0; i++ ; i \leq |T|$ ) {  
          $\tau = \Gamma(t_i);$   
         创建  $\tau$  个迁移  $\{t_{i1}, t_{i2}, \dots, t_{i\tau}\};$   
          $T' = T' \cup \{t_{i1}, t_{i2}, \dots, t_{i\tau}\};$   
         创建  $\tau-1$  个位置  $\{p_{i1}, p_{i2}, \dots, p_{i(\tau-1)}\};$   
          $P' = P' \cup \{p_{i1}, p_{i2}, \dots, p_{i(\tau-1)}\};$   
         for( $j=1; j++ ; j \leq \tau-1$ ) {  
              $K'(p_{ij})=1;$   
         }  
     }

创建弧  $f=(t_i, t_{i1})$  即  $t_{i1} = t_i;$   
 $F' = F' \cup \{f\};$   
 创建弧  $f=(t_{i\tau}, t_i')$ , 即  $t_{i\tau} = t_i';$   
 $F' = F' \cup \{f\};$   
 for( $j=1; j++ ; j \leq \tau-1$ ) {  
     创建弧  $f=(t_{ij}, p_{ij})$ , 即  $t_{ij} = p_{ij};$   
      $F' = F' \cup \{f\};$   
     创建弧  $f=(p_{ij}, t_{i(j+1)})$ , 即  $t_{i(j+1)} = p_{ij};$   
      $F' = F' \cup \{f\};$   
 }  
 $T' = T' - \{t_i\};$   
}

利用算法 2, 图 3(c) 所示的赋时 Petri 网可转换为图 4 所示的普通 Petri 网。

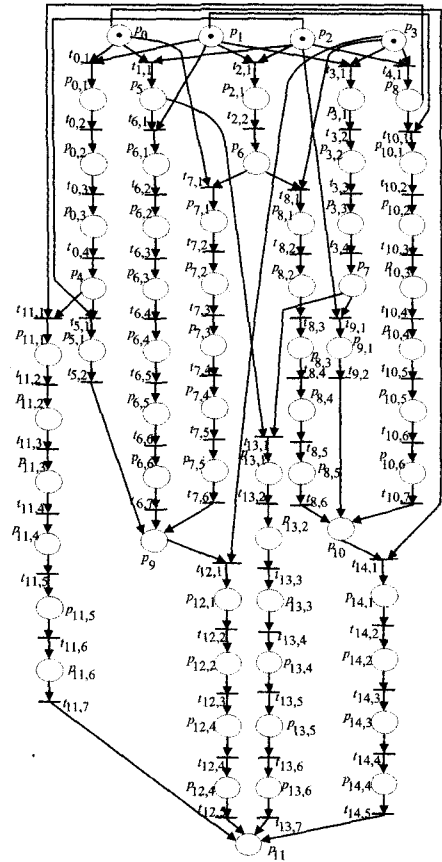


图 4 装配序列的普通 Petri 网模型

### 3 装配序列规划问题的 ZBDD 求解

#### 3.1 Petri 网的符号 ZBDD 表示

在装配序列的普通 Petri 网模型中, 每个装配操作只处理两个零部件或子装配体, 生成一个子装配体, 所以每个位置最多只有 1 个托肯, 即每个位置都是 1-有界的, 该 Petri 网模型是安全的。

在安全 Petri 网中, 一个位置  $p_i \in P$  可以由一个布尔变量  $p_i$  来编码。一个具有  $m$  个位置的安全 Petri 网, 需要  $m$  个变量进行编码。如果位置  $p_i$  中有托肯, 就用  $p_i$  表示, 否则就缺省。Petri 网可达标识的特征函数可以用位置集合的子集来表示。若某个位置包含了托肯, 该位置就在子集中出现。例如, 图 4 给出的普通 Petri 网的初始标识的特征函数可表示为  $\{p_0 p_1 p_2 p_3\}$ , 表示位置  $p_0 - p_3$  有托肯, 其它位置没有托肯。空标识集用逻辑“0”表示, 对应 ZBDD 的 0-终节点, 本文用

“LogicZero”表示。标识全集(不一定与可达标识集相等)用逻辑“1”表示,对应 ZBDD 的 1-终节点,本文用“LogicOne”表示。标识集之间的操作可以通过 ZBDD 提供的函数隐式实现。

下面讨论安全 Petri 网的迁移函数和迁移关系的二进制编码问题。为了方便起见,将迁移  $t$  的前集和后集分别记为  $I_t$  和  $O_t$ ,即:

$$I_t = \prod_{p_i \in I_t} p_i$$

$$O_t = \prod_{p_i \in O_t} p_i$$

则迁移  $t$  的最小使能标识的特征函数( $E_t$ )写成如下形式:

$$E_t = \prod_{p_i \in I_t} p_i \quad (1)$$

对于  $\forall M \in [M_0]$ ,若  $M \supseteq E_t$ ,则迁移在标识  $M$  下是使能的。

迁移  $t_k$  和迁移  $t_l$  并发的最小使能标识集的特征函数为:

$$E_{t_k \wedge t_l} = E_{t_k} \cdot E_{t_l} \quad (2)$$

对于  $M \in [M_0]$ ,若  $M \supseteq E_{t_k \wedge t_l}$ ,则迁移  $t_k$  和迁移  $t_l$  在标识  $M$  下是并发使能的。

更多迁移并发的最小使能标识集特征函数可以类似求出。

如果是在一个标识集  $\mathcal{M}$ ,而不一定是单个标识  $M$  下,若  $\mathcal{M}$  的任一子集包含  $E_t$ ,则迁移  $t$  是使能的,可以通过 ZBDD 操作  $\text{SupSet}(\mathcal{M}, E_t)$  进行判断。若返回值不为 0,则迁移  $t$  是使能的;否则不是。其中  $\text{SupSet}(X, Y) = \{x \in X \mid \exists y \in Y, x \supseteq y\}$ 。

迁移函数  $\delta$  把包含标识  $E_t$  的每个标识  $M$  通过引发迁移  $t$  而变换成新标识  $M'$ 。迁移  $t$  的迁移函数定义如下:

$$\delta(M) = (M/E_t) * \prod_{p_j \in O_t} p_j \quad (3)$$

式中,“/”和“\*”分别是 ZBDD 的除和乘操作。

如果在一个标识集  $\mathcal{M}_1$  下引发迁移  $t$ ,标识集  $\mathcal{M}_1$  由于迁移  $t$  引发而变换成标识集  $\mathcal{M}_2$ 。通常把迁移函数  $\delta$  导出的这种求一个标识集下由于迁移  $t$  引发而得到的可达标识集的计算称作 image 运算。image 运算可表达为:

$$\begin{aligned} \mathcal{M}_2 &= \text{img}(\mathcal{M}_1, t) \\ &= \{M_2 \mid \exists M_1 \in \mathcal{M}_1, M_1 \supseteq E_t, \delta(M_1) = M_2\} \end{aligned} \quad (4)$$

标识集  $\mathcal{M}$  一步可达的标识集的计算可表达为:

$$\Delta(\mathcal{M}) = \sum_{v_i \in T} (\text{img}(\mathcal{M}, t)) \quad (5)$$

### 3.2 装配序列规划的符号 ZBDD 算法

得到装配序列的 Petri 网模型后,装配序列规划问题就变成了 Petri 网的状态空间搜索问题。即找出从表示初始状态的初始标识到表示目标状态的目标标识的一个迁移引发序列,使得装配代价最小。具体算法描述如下。

**算法 3** 给定装配序列的 Petri 网模型,可以通过以下算法步骤求出最优装配序列。

**步骤 1** 将当前标识集  $init$  和已生成的可达标识集  $visited$  初始化为初始标识  $M_0$  的 ZBDD 表示,目标标识  $end$  初始化为装配产品的 ZBDD 表示,一步可达标识集  $derived$  和新标识集  $new$  初始化为 LogicZero,装配代价  $cost$  初始化为 0。

**步骤 2** 找出当前标识集  $init$  下能引发的迁移,并判定

能并发的迁移,将它们放入  $EnabledTr$ ,装配代价  $cost = cost + 1$ 。这里分 4 步:

①利用式(1)创建每个迁移的引发条件的 ZBDD,记为  $E_t$ 。

②利用 ZBDD 的外部函数  $Extra\_zddSupSet(zdd, init, E_t)$ ,找出在标识  $init$  下能引发的迁移,将能引发的迁移的个数记为  $enabl\_max$ 。

③在能引发的迁移中,利用式(2)和  $Extra\_zddSupSet$  函数找出在标识  $init$  下能并发的迁移(这里包括两个迁移并发和多个迁移并发)。

④将②和③找到的结果放入  $EnabledTr$ ,装配代价  $cost = cost + 1$ 。

**步骤 3** 判断  $EnabledTr$  是否为空。若不空,从中取出一个迁移集  $M_t$  进行引发,利用式(5)得到引发迁移后的中间结果标识集  $mediderive$ ,将其放入一步可达标识集  $derived$ ,令  $EnabledTr = EnabledTr - M_t$ ,返回步骤 3;若  $EnabledTr$  为空,则继续。其中  $mediderive = (init/E_{M_t}) * \prod_{p_j \in M_t} p_j$ 。

它通过以下函数实现:

$$tmp_1 = \text{Cudd\_zddDivide}(zdd, init, E_{M_t})$$

$$tmp_2 = \text{Cudd\_zddSubset1}(zdd, tmp_1, I_{M_t})$$

$$mediderive = \text{Cudd\_zddSubset1}(zdd, tmp_2, O_{M_t})$$

$derived$  和  $EnabledTr$  通过以下函数实现更新:

$$derived = \text{Cudd\_zddUnion}(zdd, derived, mediderive)$$

$$EnabledTr = \text{Cudd\_zddDiff}(zdd, EnabledTr, M_t)$$

**步骤 4** 求出从步骤 3 得到的新标识,放入  $new$ ,即  $new = derived - reached$ ,通过  $\text{Cudd\_zddDiff}(zdd, derived, init)$  实现。

**步骤 5** 判断新标识集  $new$  是否为空。若  $new$  为空,说明没有新标识产生,但又没有到达目标标识,算法结束,提示出错;否则继续。

**步骤 6** 利用  $Extra\_zddSupSet(zdd, new, end)$  函数判断新标识集  $new$  中是否包含目标标识(装配产品的 ZBDD 表示)。若包含,说明经过最少的迁移引发到达了目标标识,即已经找到了一条代价最小的装配序列,此时从目标标识开始回溯,以迁移序列的形式得到最优装配序列,装配代价为  $cost$ ,算法结束;若不包含,则令  $init = new$ ,  $derived = LogicZero$ ,继续。其中回溯过程是拓展新标识的逆过程。

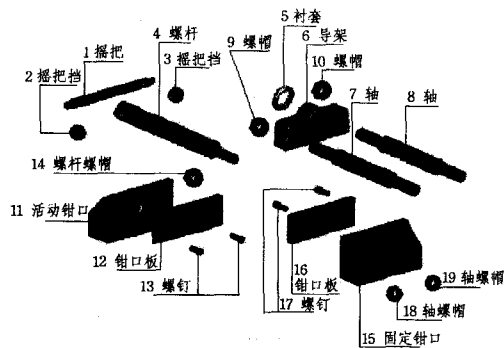
**步骤 7** 判断  $init$  是否为空。若不空,返回步骤 2;若为空,则提示出错,算法结束。

装配序列规划算法的主要思想是先将装配代价转化到迁移的个数上,装配代价的计算简化为计算迁移的总个数。然后采用实质上为宽度优先的搜索策略,保证所求的解为最优装配序列。搜索过程中对状态空间及其搜索过程中的相关数据采用 ZBDD 表示,并进行隐式操作。同时为了提高算法效率,在处理使能迁移时,引入了贪婪链技术(见步骤 3),每次处理一个迁移集而不是一个迁移,以减少循环次数。

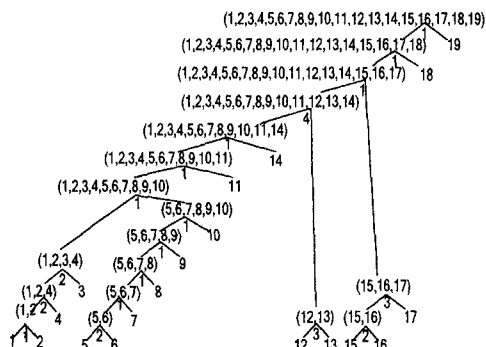
## 4 应用实例

为了检验本文给出的符号 ZBDD 规划算法的有效性,利用 Colorado 大学的 CUDD<sup>[15]</sup> 软件包及其外部函数包,在运行

平台 Windows 2000, P4 3.00GHz CPU, 256MB RAM 上, 以一个具有 19 个零件的机用虎钳为对象进行实验, 如图 5(a) 所示。得到虎钳的最优装配序列如图 5(b) 所示, 最优装配代价为 16, 运行时间是 116.66s。如果算法基于 OBDD 实现, 算法运行时间是 234.51s。基于 ZBDD 的装配序列优化算法比基于 OBDD 的装配序列优化算法节省一半的时间。



(a) 虎钳的零件图



(b) 虎钳的最优装配序列

图 5 虎钳装配实例

**结束语** 采用赋时 Petri 网为装配序列建模, 将装配类型、装配时间以及装配难度等影响装配操作的因素统一为装配代价, 体现在迁移的赋时上; 然后利用赋时 Petri 网到普通 Petri 网的转换算法, 将装配序列的赋时 Petri 网模型转换为等价的普通 Petri 网模型; 最后给出了装配序列规划的符号 ZBDD 求解算法。算法中对状态空间的隐式搜索避免了状态和搜索的显式枚举, 实现了隐式高效操作, 提高了算法的执行效率, 通过实例检验了算法的有效性。

对于实际应用中的大型复杂装配体, 可以采用分层和子装配体局部规划的思想, 将复杂装配体分解为简单子装配体, 再将子装配体看作零件进行规划, 这也是今后将要开展的重要研究工作。

## 参考文献

- [1] Molloy E, Yang H, Browne J. Feature-based Modeling in Design for Assembly [J]. International Journal of Computer Integrated Manufacturing, 1993, 6(12): 119-125
- [2] 王俊峰, 李世其, 刘继红, 等. 计算机辅助装配规划研究综述[J]. 工程图学学报, 2005, 26(2): 1-6
- [3] de Meol L S H. Representation of Mechanical Assembly Sequences [J]. IEEE Transaction on Robotics and Automation, 1991, 7(2): 211-227
- [4] Gottipolu B, GHhosh K. A Simplified and Efficient Representation for Evaluation and Selection of Assembly Sequences [J]. Computers in Industry, 2003, 50(2): 251-264
- [5] Fazio D T, Whitney D E. Simplified Generation of All Mechanical Assembly Sequences [J]. IEEE Journal Robotics and Automation, 1987, 3(6): 640-658
- [6] de Mello L S H, Sanderson A C. A Correct and Complete Algorithm for Mechanical Assembly Sequences [J]. IEEE Transaction on Robotics and Automation, 1991, 7(2): 228-240
- [7] Tianlong G, Zhoubo X, Zhifei Y. Symbolic OBDD Representations for Mechanical Assembly Sequences [J]. Computer-Aided Design, 2008, 40(4): 411-421
- [8] Lazzarini B, Marcelloni F. A Genetic Algorithm for Generating Optimal Assembly plans [J]. Artificial Intelligence in Engineering, 2000, 14(4): 319-329
- [9] 彭涛, 李世真, 王俊峰, 等. 基于集成干涉矩阵的蚁群装配序列规划 [J]. 计算机科学, 2010, 37(4): 179-182
- [10] 古天龙, 徐周波. 有序二叉决策图及应用[M]. 北京: 科学出版社, 2009
- [11] Minato S. Zero-suppressed BDDs and Their Applications [J]. International Journal on Software Tools for Technology Transfer, 2001, 3(2): 156-170
- [12] 李凤英, 古天龙, 徐周波. Petri 网的符号 ZBDD 可达树分析技术 [J]. 计算机学报, 2009, 32(12): 2420-2428
- [13] Luiz S, Homen D M, Sanderson A C. AND/OR Graph Representation of Assembly Plans [J]. IEEE Transactions on Robotics and Automation, 1990, 6(2): 188-198
- [14] Xiaoming Z, Pingan D, Yuege Z. A Model-based Approach to Assembly Sequence Planning [C] // Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation. 2007: 599-606
- [15] Somenzi F. CUDD; CU Decision Diagram Package Release 2. 3. 1 [EB/OL]. <http://vlsi.Colorado.edu/~fabio/CUDD/cuddIntro.html>, 2001-02-16

(上接第 142 页)

- [16] 闫伟, 卢炎生. 命令式语言的化学语义[J]. 小型微型计算机系统
- [17] 黄林鹏, 童维勤. 并行计算 GAMMA 概述[J]. 计算机科学, 1994, 21(5): 20-24
- [18] 赵恒, 王振宇, 曹万华, 等. 化学抽象机的分析与应用研究[J]. 计算机科学, 2003, 30(1): 42-45
- [19] 廖文昭, 童维勤, 蔡立志. 生物序列拼装欧拉路径算法的 Gamma 描述及其并行化研究[J]. 小型微型计算机系统, 2004, 25(4): 707-711
- [20] 叶俊民, 罗景, 朱凯, 等. 基于扩展 CHAM 模型的软件集成测试

方法[J]. 计算机科学, 2005, 32(6): 199-201

- [21] Kiczales G, Lamping J, Mendhekar A, et al. Aspect-Oriented Programming[A] // Aksit M, Matsuo S, eds. Proceedings of the 11th European Conference on Object-Oriented Programming [C]. LNCS 1241, Heidelberg: Springer-Verlag, 1997: 220-242
- [22] Kiczales G, Hilsdale E, Hugunin J, et al. An Overview of AspectJ[A] // Knudsen J L, ed. Proceedings of the 15th European Conference on Object-Oriented Programming[C]. LNCS 2072, Heidelberg: Springer-Verlag, 2001: 327-353