

基于符号零压缩二叉决策图的装配可行性判定方法

彭 瑞 李凤英 常 亮 孟 瑜

(桂林电子科技大学广西可信软件重点实验室 桂林 541004)

摘 要 为了扩大装配序列规划的求解规模并提高求解效率,提升装配自动化水平,给出了装配体联接矩阵和干涉矩阵的零压缩二叉决策图描述,建立了基于符号零压缩二叉决策图的装配操作可行性判定方法。基于该方法,可以高效地求解出一个装配体的可行装配操作。结合实例验证了基于符号零压缩二叉决策图的装配体模型和可行装配操作判定方法的正确性和可行性。

关键词 装配可行性,装配体联接矩阵,装配体干涉矩阵,零压缩二叉决策图

中图分类号 TP391.9 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.6.005

Symbolic ZBDD-based Judgment Method for Assembly Feasibility

PENG Rui LI Feng-ying CHANG Liang MENG Yu

(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract To enlarge the solution scale and improve the efficiency of assembly sequence planning the level of assembly automation, zero-suppressed binary decision diagram(ZBDD) was proposed to represent assembly connection matrix and interference matrix. A novel ZBDD-based method was presented to judge the feasibility of assembly operation, thereby feasible assembly operations of an assembly can be obtained efficiently. The experimental results demonstrate the validity and feasibility of ZBDD-based assembly model and judgment method for feasible assembly operation.

Keywords Assembly feasibility, Assembly connection matrix, Assembly interference matrix, Zero-suppressed binary decision diagram

1 引言

装配序列规划(Assembly Sequence Planning, ASP)是指在给定设计方案的前提下,既要保证满足所有几何约束,又要尽可能地缩短装配时间和降低装配的复杂度,从而寻找出合理的、可行的装配序列,以此来降低装配成本并优化产品性能^[1,2]。

ASP问题在本质上是NP组合优化问题^[3],穷尽搜索算法的复杂度为指数级。当产品较为复杂、零部件数较多时,极易出现状态组合爆炸问题。因此,要找到符合要求的装配序列十分困难,有些甚至无解。传统的可行装配序列生成方法有交互式问答方法^[4]、割集生成方法^[5]等,但当装配规模较大时,它们都很难保证结果的正确性和完备性,使得可以处理的装配体的零件数目受到极大的限制。由此可见,组合复杂性是制约装配序列规划效率和自动化水平的一个关键因素。

减缓或者部分程度上避免组合复杂性问题的一种可行策略是采用符号或隐式描述技术。有序二叉决策图(Ordered Binary Decision Diagram, OBDD)及其扩展形式可以实现状态空间或者变量组合的隐式表示和搜索,是迄今为止最为有效的符号技术之一^[6,7]。近年来,OBDD在装配序列规划方面也

有了一些应用。实验表明,利用OBDD作为装配序列的表示比使用AND/OR图表示占用的存储空间更小^[8,9]。

零压缩二叉决策图(Zero-Suppressed Binary Decision Diagram, ZBDD)^[10]是为了弥补OBDD处理组合集合问题的不足而提出来的,用ZBDD表示和处理组合集合问题能进一步降低空间需求,从而更好地解决组合集合问题,是求解组合集合问题的有效工具。鉴于此,本文讨论了基于符号ZBDD的装配可行性判定,给出了装配体联接矩阵和干涉矩阵模型的ZBDD描述;建立了基于符号ZBDD的装配操作可行性判定方法,利用该判定方法可以高效地求解出一个装配体的可行装配操作;并结合实例验证了基于符号ZBDD的装配体模型和可行装配操作判定方法的正确性和可行性。

2 零压缩二叉决策图

ZBDD是在OBDD的基础上,为了便于处理组合集合问题而发展起来的一种新的数据结构。相比OBDD,这种数据结构能够更加有效地表示组合集合,是求解组合问题的有效工具。

设集合 $A = \{a_1, a_2, \dots, a_n\}$,从A中不考虑次序地取出任意 r ($0 \leq r \leq n$)个元素所得到的集合称为集合中元素的一个

到稿日期:2015-05-09 返修日期:2015-10-03 本文受广西高校科研项目(YB2014125),桂林电子科技大学研究生教育创新计划资助项目(YJXCXS201541),广西高等学校高水平创新团队及卓越学者计划资助。

彭 瑞(1990—),女,硕士生,主要研究方向为符号技术和装配规划,E-mail: pengrui_email@163.com;李凤英(1974—),女,副教授,主要研究方向为Petri网、符号计算和装配规划等,E-mail: lfy@guet.edu.cn;常 亮(1980—),男,教授,主要研究方向为描述逻辑、语义Web服务等;孟 瑜(1976—),女,博士生,主要研究方向为符号计算和装配规划等。

组合。若集合 A 中的每个元素 a_i 均对应一个布尔变量 x_i ，则组合集合的元素可以用 n 维布尔向量 $x = (x_1, x_2, \dots, x_n)$ 来表示，其中 $x_i \in \{0, 1\} (i=1, 2, 3, \dots, n)$ 为向量 x 的元素，称为向量元素。当 $x_i = 0$ 时，表示对应的元素 a_i 不包含在组合中；否则，表示对应的元素 a_i 包含在组合中。由此可见，组合集合可以用一个 n 元布尔函数 $f(x_1, x_2, \dots, x_n)$ 来表示。函数 f 的值表示由该变量对应的 A 中元素的组合是否属于该组合集合，称布尔函数 f 为该组合集合的特征函数。由于 OBDD 是布尔函数的一种紧凑规范表示，基于 OBDD 可以有效完成布尔表达式的各种逻辑操作，因此可以用 OBDD 来表示组合集合的特征函数。在很多情况下，表示组合集合的 OBDD 中的节点数远少于组合集合中包含的元素个数，基于 OBDD 的组合集合上的操作时间则表示该组合集合的 OBDD 中的节点数目成正比。

但是，用 OBDD 表示组合集合存在一些问题，即组合集合的 OBDD 表示依赖于变量的个数。当变量个数不同时，相同的组合集合会得到全然不同的 OBDD。例如，如图 1 所示，在不考虑不相关元素 c 和 d 的情况下，函数 $S(a, b, c)$ 和 $S(a, b, c, d)$ 表示相同的组合集合 $\{a, b\}$ 。但由于变量个数不同，表示 $S(a, b, c)$ 和 $S(a, b, c, d)$ 的 OBDD 为两个不同的 OBDD。

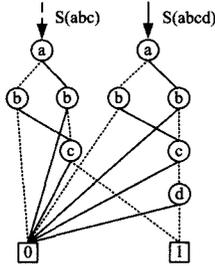


图 1 组合集合的 OBDD 表示

之所以出现上述情况，是因为不相关的变量在集合和布尔函数中的含义是不同的。在集合中，如果某变量不出现，则表明该集合中不包括这个元素，取值为 0；而在布尔函数中，如果某变量不出现，则表明该变量取值对函数值没有影响，即函数值不依赖于该变量，该变量可以取值 0 或 1。由此可见，OBDD 中的删除规则对不相关变量的影响不是很大。为了更加高效地表示组合集合，ZBDD 引入了不同于 OBDD 的化简规则，即 pD-删除规则^[10]，使 ZBDD 能够更简捷地表示组合集合，尤其是稀疏组合集合。

pD-删除规则：对于 OBDD 中的节点 u ，如果 u 的 $high$ 为终节点 0，则删除节点 u ，并将节点 u 的父节点直接连接至 u 的 low 所对应的节点，如图 2 所示。与 OBDD 相比，ZBDD 并不删除那些 0-分支节点和 1-分支节点相同的节点，而删除 1-分支节点为终节点 0 的节点。可见，在 ZBDD 中未出现变量的缺省含义与集合中未出现元素的含义一致，因此，pD-删除规则更有利于组合集合的表示。例如，在图 3 中，表示图 1 中所示的相同组合集合 $\{a, b\}$ 的 ZBDD 具有相同的结构，即 ZBDD 的结构表示不依赖于变量个数。

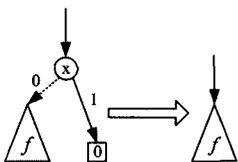


图 2 pD-删除规则

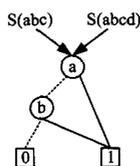


图 3 组合集合的 ZBDD 表示

另外，在 ZBDD 中，从根节点到终节点 1 的每一条路径都唯一地对应应该组合集合中的一个元素。ZBDD 以上的这些优点有效地改善了组合集合应用问题，特别是稀疏组合集合应用问题描述和处理的时空效率，可使稀疏组合集合应用问题描述的规模变为 OBDD 的 $1/n$ 。因此，ZBDD 更适合于表示组合集合，是求解组合集合问题的有效工具^[11]。

采用 OBDD 表示装配序列和装配操作时，实质上是将装配操作和装配序列表示成布尔特征函数，然后根据布尔函数和 OBDD 之间的对应关系建立装配序列的 OBDD 表示^[9,12]。在这种表示中，包含在子装配体中的零件用变量的肯定形式表示，不包含在子装配体中的零件用变量的否定形式表示，即在一个装配体的装配过程中，无论子装配体包含的零件有多少，表示子装配体的变量数都一样多，这样就增加了装配序列规划过程中对空间的需求。因此，本文改变以往使用的 OBDD 结构，研究基于符号 ZBDD 的装配可行性判定，从而更好地利用 ZBDD 表示组合集合的优越性，扩大装配序列规划求解的规模。

3 装配体模型的 ZBDD 表示

机械装配是将相互连接的零部件组合在一起，从而形成一个稳定的装配体。每个零部件都是一个坚固的刚性物体，即它的形状保持不变。两个零件不管有一个还是多个面相互接触，都称二者是相互连接的。零件之间的接触会降低相互之间的移动自由度^[13]。装配体各零件之间的逻辑和物理关系决定了接触和相互移动的信息，称其为装配知识，可以直接从装配体的计算机辅助设计 (Computer Aided Design, CAD) 模型中提取出来。

3.1 装配体联接矩阵的 ZBDD 表示

装配联接图^[14]是装配序列生成所依据的主要模型。在装配联接图中，节点表示装配体中所具有的零配件，连接弧表示零件之间的连接关系。装配联接图是一个用二元组 $G = (P, L)$ 表示的无向图。其中， P 表示零件的集合， L 表示连接边的集合。无向图的典型代数表示是关系矩阵：行、列依次按照节点编号索引；行列交叉处元素表示所对应零件对之间的连接关系 (1 表示存在连接关系，0 表示不存在连接关系)。因此，引入联接矩阵^[15]来表示联接图。

对于一个给定的装配体，可以通过对装配体的各个零件和子装配体进行编码来得到装配体联接矩阵的 ZBDD 表示。对于一个具有 n 个零件的装配体，可用长度为 n 的二进制变量 $X = (x_0, x_1, \dots, x_{n-1})$ 对零件和子装配体进行编码。每个变量表示一个零件，两个或多个变量组成的集合表示子装配体。根据装配体的联接矩阵，得到装配体联接矩阵的组合集合表示 $C(X) = \{x_i x_j \mid x_i \in X, x_j \in X, i \neq j\}$ ，其中 x_i 和 x_j 表示两个不同的零件。根据装配体联接矩阵的组合集合以及组合集合到 ZBDD 的映射关系，进而可以得到装配体联接矩阵的 ZBDD 表示。

例如，图 4 所示的简单装配体包含有 4 个零件，其联接图如图 5 所示。其中， $L = \{(a, b), (a, c), (a, d), (b, d), (c, d)\}$ 。因此，可得装配体的联接矩阵，如图 6 所示。用 4 维布尔向量 $X = (x_0, x_1, x_2, x_3)$ 对其进行编码，从而可得装配体联接矩阵的组合集合表示为 $C(x_0, x_1, x_2, x_3) = \{x_0 x_1, x_0 x_2, x_0 x_3, x_1 x_3, x_2 x_3\}$ ，进而可以得到该装配体联接矩阵的 ZBDD 表示，如图 7 所示。

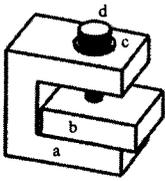


图4 实例模型图

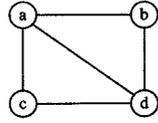


图5 实例的联接图

	a	b	c	d
a	0	1	1	1
b	1	0	0	1
c	1	0	0	1
d	1	1	1	0

图6 实例的联接矩阵

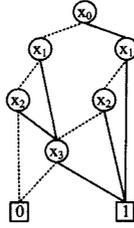


图7 联接矩阵的ZBDD表示

$x_0x_2z_5, x_0x_3z_1, x_1x_2z_0, x_1x_2z_1, x_1x_2z_2, x_1x_2z_3, x_1x_2z_5, x_1x_3z_1, x_2x_3z_1$ 。进而可以得到该装配体干涉矩阵的ZBDD表示,如图8所示。

表1 实例的干涉矩阵

零件对	T ₀	T ₁	T ₂	T ₃	T ₄	T ₅
(a,b)	1	0	1	0	0	1
(a,c)	1	1	1	1	0	1
(a,d)	0	1	0	0	0	0
(b,c)	1	1	1	1	0	1
(b,d)	0	1	0	0	0	0
(c,d)	0	1	0	0	0	0

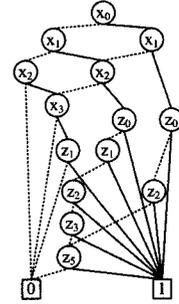


图8 干涉矩阵的ZBDD表示

3.2 装配体干涉矩阵的ZBDD表示

装配体的联接矩阵只能说明某两个零件之间的联接关系,但并不足以保证两者能够装配在一起。一个可行的装配操作还必须要满足参与装配的两个零件或子装配体存在在某个方向上的移动自由度,即存在一条在某个方向上的无冲突装配路径。Gottipolu 和 Ghosh 定义了装配体的干涉矩阵^[16],用于表示零件之间的可相互移动关系。通过干涉矩阵,可以判断出零件之间是否存在无冲突的装配路径。

两个零件 a 和 b 之间的移动向量函数可以表示为 $T_{ab} = (T_0, T_1, T_2, T_3, T_4, T_5)$, 其被称作转移函数或者 T -函数, 定义为:

$$T_{ab} = T_i \rightarrow \{0, 1\}, i=0, 1, 2, 3, 4, 5$$

T_{ab} 的第 i 个分量 $T_i = 1 (i=0, 1, 2, 3, 4, 5)$ 表示零件 b 对于 a 存在在方向 i 上的移动自由度, 即零件 b 在方向 i 上有一条到零件 a 的无冲突装配路径; $T_i = 0 (i=0, 1, 2, 3, 4, 5)$ 表示零件 b 对于 a 不存在在方向 i 上的移动自由度, 即零件 b 在方向 i 上与零件 a 存在干涉, 无法实施装配。其中, $0, 1, 2, 3, 4, 5$ 分别对应于三维笛卡尔坐标系统的 3 个坐标轴的 6 个方向: $+X, +Y, +Z, -X, -Y, -Z$ 。方向 $0, 1, 2$ 分别表示 X, Y, Z 轴的正方向, 方向 $3, 4, 5$ 分别表示 X, Y, Z 轴的负方向。另外, 如果零件 b 对于 a 存在在方向 i 上的移动自由度, 则零件 a 对于零件 b 存在在方向 $((i+3) \bmod 6)$ 上的移动自由度。因此, 给出 T -函数的前半部分足以判断零件之间的相互移动关系。

对于一个给定的装配体, 可以通过对其零件、子装配体和方向进行编码得到干涉矩阵的 ZBDD 表示。对于一个具有 n 个零件的装配体, 可用长度为 n 的二进制变量 $X = (x_0, x_1, \dots, x_{n-1})$ 对零件和子装配体进行编码。然后, 对坐标轴的各个方向进行编码, 三维坐标共有 6 个方向, 所以使用 6 位二进制变量 $Z = (z_0, z_1, z_2, z_3, z_4, z_5)$ 为其编码。根据装配体的干涉矩阵, 可以得到装配体干涉矩阵的组合集合, 表示为 $T(XZ) = \{x_i x_j z_k \mid x_i \in X, x_j \in X, z_k \in Z, i \neq j\}$, 其中 x_i 和 x_j 表示两个不同的零件。根据装配体干涉矩阵的组合集合以及组合集合到 ZBDD 的映射关系, 进而可以得到装配体干涉矩阵的 ZBDD 表示。

例如, 图 4 所示的简单装配体的干涉矩阵如表 1 所列。由表 1 可得, $T = (x_0, x_1, x_2, x_3, z_0, z_1, z_2, z_3, z_4, z_5) = \{x_0x_1z_0, x_0x_1z_2, x_0x_1z_5, x_0x_2z_0, x_0x_2z_1, x_0x_2z_2, x_0x_2z_3,$

4 基于符号 ZBDD 的装配可行性判定

一个子装配体包含一个或者多个零件, 其中, 每一个零件至少与其他零件有一个接触关系。对于一个具有 n 个零件的装配体, 可用长度为 n 的二进制变量 $X = (x_0, x_1, \dots, x_{n-1})$ 对零件和子装配体进行编码。其中, 每一个变量 x_i 对应一个零件 i , 两个或多个变量组成的集合表示子装配体。变量 x_i 的出现与否, 表示零件 i 是否已装配。因此, 装配状态可以使用组合集合来表示。例如, 图 4 所示的简单装配体具有 4 个零件, 可用 4 个二进制变量 x_0, x_1, x_2, x_3 分别为它们编码。 $\{abc\}$ 为一个子装配体, 可用变量集合 $\{x_0x_1x_2\}$ 来表示。初始装配状态和结束装配状态可用 $\{x_0, x_1, x_2, z_3\}$ 和 $\{x_0x_1x_2z_3\}$ 来分别表示。

一个装配体的装配过程包含一系列的装配操作, 每一个装配操作把两个小零件或子装配体组装成一个较大的子装配体; 并且, 假设一旦两个零件组装在一起, 这种状态将保持到装配结束。当一个装配操作的两个零件存在沿某个方向的无冲突装配路径时, 该装配操作是几何可行的。另外, 如果能够建立起装配零部件之间的连接关系, 那么该装配操作也是机械可行的。给定两个子装配体 S_1 和 S_2 , 如果集合 $S_3 = S_1 \cup S_2$ 表示的子装配体存在, 那么称 $(\{S_1, S_2\}, S_3)$ 为一个可行装配操作, 用 τ 表示。由于 $S_2 = S_3 - S_1$, 因此 τ 可用一个长度为 $2n$ 的二进制变量 (X, Y) 来表示, 其中 $X = (x_0, x_1, \dots, x_{n-1})$ 用于表示 $S_1, Y = (y_0, y_1, \dots, y_{n-1})$ 用于表示 S_2 。例如, 对于图 4 所示装配体, 如果 $S_1 = \{ab\}, S_2 = \{c\}$, 则 $S_3 = \{abc\}$, 那么该装配任务可表示为 $\tau = \{x_0x_1y_0y_1y_2\}$ 。

基于以上描述, 给出了基于 $C(X)$ 和 $T(XZ)$ 的符号 ZBDD 表示的装配操作的可行性判定方法。记 $C(X)$ 的 ZBDD 表示为 $ZBDD_C, T(XZ)$ 的 ZBDD 表示为 $ZBDD_T$ 。

用集合 S_1 和 S_2 表示两个零件或者子装配体, 当 $S_1 \cap S_2 = \emptyset$ 时, 可通过算法 1 和算法 2 对二者装配操作的可行性进行判定。

算法 1 给定零件或者子装配体 S_1 和 S_2 , 判定 S_1 和 S_2 的装配操作是否满足连接关系约束

- 步骤 1 分别建立 S_1 和 S_2 所代表的零件或子装配体的组合集合 $F_1(X)$ 和 $F_2(X)$ 。
- 步骤 2 计算 $F_1(X)$ 和 $F_2(X)$ 的笛卡尔积, 记为 $F_{12}(X)$, 即 $F_{12}(X) = F_1(X) \times F_2(X)$, 并创建 $F_{12}(X)$ 所对应的 ZBDD, 记为 $ZBDD_{12}$ 。
- 步骤 3 执行 $ZBDD_{12} \cap ZBDD_C$, 若结果不为空, 则 S_1 和 S_2 所代表的两个零件或子装配体满足连接关系约束; 否则, 不满足连接关系约束。

连接关系约束的满足是装配操作得以实施的必要条件, 它并不能保证装配操作的几何可行性, 必须还要满足优先关系约束。为了判定装配两个零件或子装配体的装配操作是否满足优先关系约束, 给出了判定算法 2。

算法 2 给定零件或者子装配体 S_1 和 S_2 , 判定 S_1 和 S_2 的装配操作是否满足优先关系约束

- 步骤 1 分别建立 S_1 和 S_2 所代表的零件或子装配体的组合集合 $F_1(X)$ 和 $F_2(X)$ 。
- 步骤 2 计算 $F_1(X)$ 和 $F_2(X)$ 的笛卡尔积, 记为 $F_{12}(X)$, 即 $F_{12}(X) = F_1(X) \times F_2(X)$, 并创建 $F_{12}(X)$ 所对应的 ZBDD, 记为 $ZBDD_{12}$ 。
- 步骤 3 对于 $ZBDD_{12}$ 中的第一条路径 $x_0 x_1 \dots x_{n-1}$, 在 $ZBDD_T$ 中搜索对应的路径 $x_0 x_1 \dots x_{n-1} z_0 z_1 z_2 z_3 z_4 z_5$, 截取 $z_0 z_1 z_2 z_3 z_4 z_5$, 创建相应的 ZBDD, 记为 $ZBDD_{DS}$ 。创建用于存放待装配零部件之间的无冲突装配路径的 ZBDD, 并令 $ZBDD_D = ZBDD_{DS}$ 。
- 步骤 4 对于 $ZBDD_{12}$ 中剩余的每一条路径 $x_0 x_1 \dots x_{n-1}$, 在 $ZBDD_T$ 中搜索对应的路径 $x_0 x_1 \dots x_{n-1} z_0 z_1 z_2 z_3 z_4 z_5$, 截取 $z_0 z_1 z_2 z_3 z_4 z_5$, 创建相应的 ZBDD, 记为 $ZBDD_{DS}$, 然后执行 $ZBDD_D = ZBDD_D \cap ZBDD_{DS}$ 。
- 步骤 5 若 $ZBDD_D$ 不为空, 则 S_1 和 S_2 所代表的两个零件或子装配体满足优先关系约束; 否则, 不满足优先关系约束。

如果 S_1 和 S_2 所代表的两个零件或子装配体既满足连接关系约束, 又满足优先关系约束, 那么两者的装配操作是可行的; 否则, 不可行。另外, 如果 S_1 和 S_2 所代表的都为单个零件, 则只满足算法 1 时两者的装配操作即为可行的。若可行, 则装配操作可表示为 $\tau = \{S_1, S_3\}$ 。其中, $S_3 = S_1 \cup S_2$ 。

因此, 对于一个给定的装配体, 可以根据算法 1 和算法 2 对其所有可能可行的装配操作进行判定, 从而找出所有可行的装配操作, 进而根据所有可行的装配操作找出该装配体完备的可行装配序列。

以图 4 所给装配体为例, 对上述算法的有效性进行验证。首先, 取 $S_1 = \{a\}, S_2 = \{b\}$, 根据算法 1 判断二者是否满足连接关系约束。

步骤 1 建立 S_1 和 S_2 的组合集合, 可得 $F_1(X) = \{x_0\}, F_2(X) = \{x_1\}$ 。

步骤 2 计算 $F_1(X)$ 和 $F_2(X)$ 的笛卡尔积, 可得 $F_{12}(X) = F_1(X) \times F_2(X) = \{x_0 x_1\}$ 。另外, 创建该组合集合对应的 $ZBDD_{12}$ 。

步骤 3 执行 $ZBDD_{12} \cap ZBDD_C$, 可得 $ZBDD_{12} \cap ZBDD_C = \{x_0 x_1\}$, 结果不为空, 因此二者满足连接关系约束。因为此时 S_1 和 S_2 所代表的都为单个零件, 所以只满足算法 1 时两者的装配操作即为可行的。因此, 可得 $S_3 = \{ab\}, \tau_1 = \{x_1 y_0 y_1\}$ 。

接下来, 取 $S_1 = \{ab\}, S_2 = \{c\}$, 根据算法 1 判断二者是否满足连接关系约束。

步骤 1 建立 S_1 和 S_2 的组合集合, 可得 $F_1(X) = \{x_0, x_1\}, F_2(X) = \{x_2\}$ 。

步骤 2 计算 $F_1(X)$ 和 $F_2(X)$ 的笛卡尔积, 可得 $F_{12}(X) = F_1(X) \times F_2(X) = \{x_0 x_2, x_1 x_2\}$ 。另外, 创建该组合集合对应的 $ZBDD_{12}$ 。

步骤 3 执行 $ZBDD_{12} \cap ZBDD_C$, 可得 $ZBDD_{12} \cap ZBDD_C = \{x_0 x_2\}$, 结果不为空, 因此满足连接关系约束。

然后, 根据算法 2 判断 S_1 和 S_2 是否满足优先关系约束。首先, 通过算法 1 中的步骤 1 和步骤 2, 求得 $ZBDD_{12} = \{x_0 x_2, x_1 x_2\}$ 。其次, 根据步骤 3 在 $ZBDD_T$ 中搜索与 $x_0 x_2$ 对应的路径, 有 $x_0 x_2 z_0, x_0 x_2 z_1, x_0 x_2 z_2, x_0 x_2 z_3, x_0 x_2 z_5$ 。由此可得, $ZBDD_D = \{z_0, z_1, z_2, z_3, z_5\}$ 。最后, 根据步骤 4 对剩余路径 $x_1 x_2$ 进行处理, 同理可得 $ZBDD_D = \{z_0, z_1, z_2, z_3, z_5\}$, 即二者可在 z_0, z_1, z_2, z_3 方向或者 z_5 上进行装配。

因为 $ZBDD_D$ 不为空, 根据步骤 5 可知, 二者满足优先关系约束。因此, 此时 S_1 和 S_2 所代表的子装配体满足优先关系约束。故有 $S_3 = \{abc\}, \tau_2 = \{x_0 x_1 y_0 y_1 y_2\}$ 。

最后, 取 $S_1 = \{abc\}, S_2 = \{d\}$ 。分别根据算法 1 和算法 2 判定二者是否满足连接关系和优先关系约束。根据算法 1 得 $ZBDD_{12} \cap ZBDD_C = \{x_0 x_3, x_1 x_3, x_2 x_3\}$, 结果不为空, 因此满足连接关系约束。根据算法 2 得 $ZBDD_D = \{z_1\}$, 由此可知二者也满足优先关系约束, 并且可在方向 z_1 上进行装配。故有 $S_3 = \{abcd\}, \tau_3 = \{x_0 x_1 x_2 y_0 y_1 y_2 y_3\}$ 。此时, S_3 是一个完整的装配体, 因此, 得到一个可行的装配序列 $\tau_1 \tau_2 \tau_3$ 。最后, 把可行的装配序列添加到 ZBDD 中保存。

重复执行上述判定过程, 即可得到装配体完备的几何可行装配序列。图 9 给出了所求得的图 4 所示实例的可行装配序列的 ZBDD 表示。

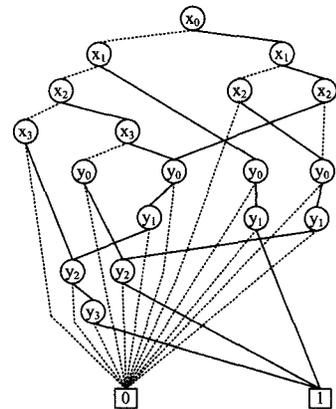


图 9 可行装配序列的 ZBDD 表示

结束语 本文研究了基于符号 ZBDD 的装配可行性的判定方法, 给出了装配体联接矩阵和干涉矩阵的 ZBDD 描述, 建立了基于符号 ZBDD 的装配体模型。在此基础上, 给出了基于符号 ZBDD 的连接约束和优先约束的可满足性判定算法, 基于该算法, 可以得到装配体的所有可行装配操作。最后通过装配实例, 验证了算法的有效性。今后的研究重点是进一步开展基于符号 ZBDD 的装配序列的生成、评价与优化, 扩大装配序列规划问题的求解规模。

(下转第 49 页)

不降低系统性能和用户体验的基础上提高缓存命中率,合理使用缓存空间。下一步,我们将更深入研究移动社交网络中用户的行为对缓存的影响,以优化现有的缓存替换算法。

参考文献

- [1] Jain D K, Sharma S. Growth Rate of Cached Data Items at clients in Mobile Ad Hoc Networks [C] // 2014 IEEE Global Conference on Wireless Computing and Networking (GCWCN). IEEE, 2014: 157-159
- [2] Yang L, Qin Y, Zhou X, et al. Social Relation Based Cache Distribution Policy in Wireless Mobile Networks [J]. Journal of Networks, 2014, 9(9): 2279-2288
- [3] Ma Hong-yuan, Wang Bin. Query Results Caching and Prefetching in Web Search Engines Based on User Characteristics [J]. Journal of Chinese Information Processing, 2012, 26(6): 19-26 (in Chinese)
马宏远, 王斌. 基于用户特性的搜索引擎查询结果缓存与预取 [J]. 中文信息学报, 2012, 26(6): 19-26
- [4] Yan Xia. Caching Based on Actual User Behavior [D]. Beijing: University of Science and Technology of China, 2011 (in Chinese)
夏琰. 基于实际用户行为分析的缓存研究 [D]. 北京: 中国科学技术大学, 2011
- [5] Wang Y, Wu J, Xiao M. Hierarchical cooperative caching in mo-

bile opportunistic social networks [C] // Proc. of IEEE GLOBE-COM, 2014

- [6] Kumar V R, Swati M. Cache Replacement Algorithms for Coordinated Cooperative Social Wireless Networks [J]. International Journal of Computer Science and Mobile Computing, 2014, 3(10): 718-725
- [7] Liu Qian. Friend Recommendation Based on Social Network and Location Information [D]. Hangzhou: Zhejiang University, 2013 (in Chinese)
刘乾. 基于社交网络和地理位置信息的好友推荐方法研究 [D]. 杭州: 浙江大学, 2013
- [8] Branzei S, Larson K. Social distance games [C] // The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3. International Foundation for Autonomous Agents and Multiagent Systems, 2011: 1281-1282
- [9] http://en.wikipedia.org/wiki/Mobile_social_network
- [10] http://en.wikipedia.org/wiki/Six_degrees_of_separation
- [11] Madhukar A, Özyer T, Alhajj R. Dynamic cache invalidation scheme for wireless mobile environments [J]. Wireless Networks, 2009, 15(6): 727-740
- [12] Rathore R, Prinja R. An Overview of Mobile Database Caching [J]. CiteSeerX, doi, 2007, 10(1.100): 9481
- [13] Leong H V, Si A. On Adaptive Caching in Mobile Databases [C] // Proceedings of the 1997 ACM Symposium on Applied Computing, 1997: 302-309

(上接第 31 页)

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] Wang Jun-feng, Li Shi-qi, Liu Ji-hong, et al. Computer Aided Assembly Planning: a Survey [J]. Journal of Engineering Graphics, 2005, 26(2): 1-6 (in Chinese)
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] Uma R N, Wein J, Williamson D P. On the Relationship between Combinatorial and LP-Based Lower Bounds for NP-hard Scheduling Problems [J]. Theoretical Computer Science, 2006, 361(2): 241-256
- [4] Fazio D T, Whitney D E. Simplified Generation of All Mechanical Assembly Sequences [J]. IEEE Journal Robotics and Automation, 1987, 3(6): 640-658
- [5] 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
- [6] Bryant R E. Symbolic Boolean Manipulation with Ordered Binary Decision Diagram [J]. ACM Computing Surveys, 1992, 24(3): 293-318
- [7] Xu Zhou-bo, Gu Tian-long, Zhao Ling-zhong. A Novel Symbolic ADD Algorithm for Maximum Flow in Networks [J]. Journal on Communications, 2005, 26(2): 1-8 (in Chinese)
徐周波, 古天龙, 赵岭忠. 网络最大流问题的一种新的符号 ADD 求解算法 [J]. 通信学报, 2005, 26(2): 1-8
- [8] 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

- [9] Gu Tian-long, Liu Hua-dong. Symbolic OBDD-based technique for generating assembly sequences [J]. Computer Integrated Manufacturing Systems, 2008, 14(2): 321-328 (in Chinese)
古天龙, 刘华东. 基于符号有序二叉决策图的装配序列生成技术 [J]. 计算机集成制造系统, 2008, 14(2): 321-328
- [10] Minato S. Zero-suppressed BDDs and Their Applications [J]. International Journal on Software Tools for Technology Transfer, 2001, 3(2): 156-170
- [11] Li Feng-ying, Gu Tian-long, Chang Liang, et al. Timed Petri and ZBDD Based Approach for Assembly Sequence Planning [J]. Computer Science, 2012, 39(2): 170-174 (in Chinese)
李凤英, 古天龙, 常亮, 等. 一种基于赋时 Petri 网和 ZBDD 的装配序列规划方法 [J]. 计算机科学, 2012, 39(2): 170-174
- [12] Gu Tian-long, Xu Zhou-bo, Yang Zhi-fei. Symbolic OBDD Representations for Mechanical Assembly Sequences [J]. Computer-Aided Design, 2008, 40(4): 411-421
- [13] Hu Min, Wang Yan-wei, Nie Bin, et al. Assembly Sequence Planning Based on Contact-Relation Analysis [J]. Journal of Computer-Aided Design & Computer Graphics, 2014, 26(8): 1374-1384 (in Chinese)
胡敏, 王彦伟, 聂斌, 等. 基于接触关系分析的装配序列规划 [J]. 计算机辅助设计与图形学学报, 2014, 26(8): 1374-1384
- [14] Homem D E, Mello L S, Sanderson A C. Representation of Mechanical Assembly Sequence [J]. IEEE Transactions on Robotics and Automation, 1991, 7(2): 211-227
- [15] Gottipolu R B, Ghosh K. A Simplified and Efficient Representation for Evaluation and Selection of Assembly Sequences [J]. Computer in Industry, 2003, 50(2): 251-264
- [16] Gottipolu R B, Ghosh K. An Integrated Approach to the Generation of Assembly Sequences [J]. International Journal of Computer Application in Technology, 1995, 8(3): 125-138