

文章编号:1000-582X(2012)03-033-07

面向产品族配置的约束网络模型及配置方法

董元发¹, 郭 钢¹, 付 豪¹, 孙 敏²

(1. 重庆大学 机械工程学院, 重庆 400044;
2. 中国嘉陵工业股份有限公司(集团), 重庆 400032)

摘 要:针对面向产品族的产品配置问题,建立了客户需求族模型,引入了需求元与配置元的概念;构建了需求元与配置元之间以及配置元与配置元之间基于特征属性匹配的约束网络模型。按需求元权重依次进行约束传播,通过预处理技术缩小与需求元有约束关系的配置元值域空间,减小了配置求解的搜索空间,采用非二元弧一致回溯算法对于未确定的配置元进行约束求解,得到满足客户需求的配置结果集。以某底盘产品的配置设计实例验证了该方法的有效性。

关键词:产品配置;客户需求族;需求元;配置元;谓词逻辑;约束网络;非二元弧一致
中图分类号:TP391 **文献标志码:**A

Product family configuration-oriented constraint network model and configuration

DONG Yuan-fa¹, GUO Gang¹, FU Hao¹, SUN Min²

(1. College of Mechanical Engineering, Chongqing University, Chongqing 400044, P. R. China;
2. China Jialing Industrial Co., Ltd. (Group), Chongqing 400032, P. R. China)

Abstract: Concepts of requirement elements and configuration elements and the model of customer requirements family are proposed for the configuration problem of product family. The constraint relationships are taken into account to establish a feature attribute mapping-based constraint network, which is between configuration elements, as well as requirement element and configuration element. The scale of the configuration problem solving has been reduced through the ordinal constraint diffusion that according to the weight of requirement elements, and also through the range reduction of configuration element related to requirement element by pretreatment. The set of rapid configuration results is obtained by constraint solving for undetermined configuration element, applied the non-binary uniformity backtracking algorithm. Ultimately the feasibility of the proposed approach is validated by means of an example of a chassis configuration design.

Key words: product configuration; customer requirement family; requirement element; configuration element; predicate logic; constraint network; non-binary uniformity

收稿日期:2011-10-15

基金项目:国家科技支撑计划项目(2006BAF01A41);国家 863/CIMS 主题资助项目(2007AA04Z1B1)

作者简介:董元发(1988-),男,重庆大学博士研究生,主要从事新产品开发战略方面的研究。

郭钢(1960-),男,重庆大学教授,博士生导师,(E-mail)cqguogang@163.com。

如何按客户的个性化需求,在企业已有系列产品基础上,利用功能模块的搭配和零部件间的可替换关系,快速配置出产品,实现快速生产和按期交货,是目前大规模定制企业普遍关注的热点问题。

目前,国内外学者对此问题开展了一系列研究工作,主要集中在客户需求结构化建模^[1]、产品配置知识的表达^[2]、产品配置方案搜索与优化算法等^[3-4]方面。目前常见的产品配置方法主要有基于规则、基于实例及基于约束等。文献[5-6]利用 IF-THEN 形式的产生式规则描述配置约束信息;文献[7]基于语义网络表达产品配置知识,便于配置知识的重用与共享;基于规则的方法随着配置领域知识的日益复杂与知识更新的日益频繁,知识库难以维护。文献[8-10]对产品族内的配置实例进行形式化描述,利用基于实例的方法检索出与客户需求匹配的产品实例。文献[11-12]将配置问题转化为约束满足问题,通过取各对象域中的不同域值进行组合,并满足预先定义的对象间的所有约束,从而生成各种不同的产品配置。文献[13]提出了一种可以灵活集成不同求解器的配置软件框架。文献[14]通过存储和管理不同层次的配置知识,建立了虚拟配置模型,简化配置系统的实现。此外,还有基于相关性分析的协同配置方法^[15]。近年来,由于基于约束满足的方法在配置知识表达能力和求解方面的良好表现,受到越来越多的关注。但传统的约束满足求解算法针对单个客户需求

时,会不必要的增大搜索空间和时间开销。

面向客户需求族的产品配置问题可归纳为以下几个问题的求解:1)客户需求分解与结构化描述;2)产品族配置模型与零部件配置规则;3)客户需求族模型与产品族配置模型间的映射与求解。笔者通过引入需求元与配置元的概念,提出了一种基于约束网络的产品配置模型及求解算法,可为构建大规模定制行业的智能配置器奠定良好基础。

1 客户需求族模型

客户需求来自客户订单或预测市场,由于每个客户对同一类产品都有不同的个性化需求,但从一个客户群来看,总可以找到一些具有相同或相似需求的客户群,称为细分市场客户群。由此,通过大量市场调查,可获得某类产品的细分市场客户群 1、客户群 2、客户群 N,每个客户群的需求都可用一个需求结构树来表示。在同一客户群中其需求是完全相同的,在不同客户群中,有完全相同的需求,也有部分相同的需求,还有不同的个性化需求。因此,笔者把多个客户群的需求按完全相同需求、部分相同需求和个性化需求定义为 3 个类别,每个类别中的需求又分为不同的需求元(如需求 A_1 、需求 A_2 、需求 B_1 、需求 B_2 等),这样就可把多个客户群的需求用需求结构矩阵树来描述,定义为客户需求族模型,如图 1 所示。



图 1 表征多个客户群需求结构矩阵的客户需求族模型

2 产品族模型

在生产实际中,许多产品在用途和功能上基本一致,但它们在功能参数、性能指标、尺寸规格等方面存在一定差异,以满足同类客户的不同需求,这类产品称为系列产品。表达一个系列产品的 BOM 树状结构,是在基型产品的基础上,保留共用零部件不变,在部分共用件和专用件的配置位置上,将系列产品中每个型号产品所属的部分共用件和专用件加在基型产品可替换件的同一位置上,形成一个复合式产品 BOM 结构,称为产品族 GBOM。产品族 GBOM 中零部件间的上下、左右隶属关系,描述了一个系列产品中通用件(又叫基础件)、部分通用件和专用件在系列产品中的配置关系,以及系列产品中各型号产品的演化过程、不同规格型号产品间零部件的配置差异、功能特征属性和适用范围,对于按不同顾客订单需求选择合适的产品具有重要意义。

3 面向客户需求族的产品族配置模型

产品配置是指以客户需求和产品信息模型为输入,以产品的最终配置结果为输出的一类设计活动。产品族 GBOM 结构中具有某种特定功能的零部件称为模块;某个模块或模块组合形成的特定功能与客户某一需求形成对应关系,模块与模块之间又存在许多相互依赖或者互斥的关系,这些关系一般均可以用约束的形式表达,约束之间的相互关联则构成了产品配置的约束网络。在此将客户需求族中的每个独立需求定义为需求元(Requirement Element),产品族 GBOM 结构中的每个可配置的模块定义为配置元(Configuration Element),由此,客户需求族与产品族 BOM 间的配置关系,可演化为需求元与配置元之间基于特征属性映射的约束网络。

3.1 需求元

客户需求是整个配置过程的输入,包含对产品功能、结构、性能等方面的要求,考虑客户需求的特点及后续配置结果优化,笔者将客户需求描述为一个需求元向量组:

$$R = \begin{pmatrix} T \\ O \\ F \\ D \\ W \end{pmatrix}^T = \begin{pmatrix} t_1 & O_1 & F_1 & D_1 & w_1 \\ t_2 & O_2 & F_2 & D_2 & w_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_n & O_n & F_n & D_n & w_n \end{pmatrix}, \quad (i = 1, 2, \dots, n) \quad (1)$$

式中 $T = (t_1, t_2, \dots, t_n)^T$, 表示需求元的需求类型,0 表示功能需求,1 表示性能需求,2 表示结构需求,3

表示心理需求,4 表示其他; O 指需求名称,便于标识和客户理解; F 表示需求元对应的零部件特征属性, $F_i = \{f_i^1, f_i^2, \dots, f_i^n\}$, f_i^n 表示第 i 个需求元关联的第 n 个特征属性; D 表示该需求元对应的值域, $D_i = \{d_i^1, d_i^2, \dots, d_i^n\}$, d_i^n 表示第 i 个需求元的第 n 个取值; W 表示需求在客户心中的权重,是影响最终配置结果的重要参数,个性化需求比共性需求权重值高, $W = (w_1, w_2, \dots, w_n)^T$, 其中 $w_i \in (0, 1)$, w_i 表示第 i 个需求元的权重值。

例如 $R_i = \{2, \text{卡车外型}, (\text{驾驶室外型}, \text{风窗系统外型}, \text{侧围系统外型}, \text{顶盖系统外型}, \text{卧铺系统外型}), (\text{平顶}, \text{高顶}), 0.3\}$ 表示结构需求类的卡车外型选择范围有平顶和高顶,卡车外型关联以下零部件特征属性:驾驶室外型、风窗系统外型、侧围系统外型、顶盖系统外型、卧铺系统外型,卡车外型在客户需求中的权重值为 0.3。

3.2 配置元

根据公理化设计理论,可将每一个功能元对应的零部件组合视为一个组合部件。实际配置过程中,产品族 GBOM 中的模块只分解到顶层部件,它包含着模块自身属性及模块与模块间的装配关系;配置元可以描述为一个五元组 $E = \{A, F, T, D, C\}$, 其中各符号含义如下:

1) $A = \{\text{Code}, \text{Type}, \text{Name}, \text{Version}, \text{Amount}, \text{Cost}, \text{ParentID}, \text{Childs}\}$, A 是有限变量的集合,用来表征配置元的属性信息,便于其他信息系统集成。 Type 为选项型变量,标识该配置元的通用程度, $\text{Type}=0$ 时为完全通用件, $\text{Type}=1$ 时为部分通用件, $\text{Type}=2$ 时为专用件;如图 2 所示驾驶室系统是专用件,版本号为 00, 编码为 50ZAX12100, 父节点编码为 ZAX121, 成本 10 000 元, 它含有 4 个子件。

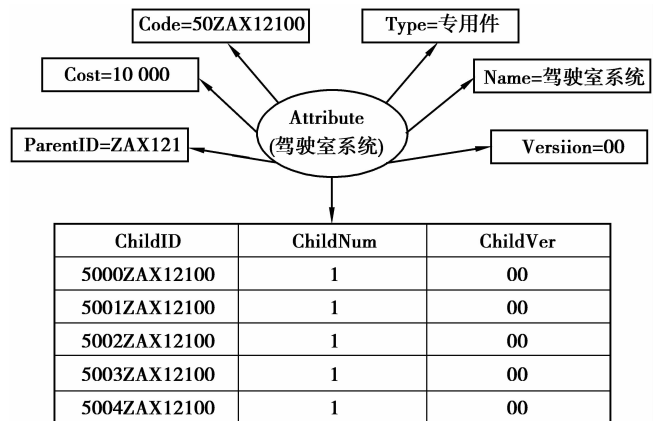


图 2 驾驶室系统配置元属性集合

2) $F = \{f_1, f_2, \dots, f_n\}$, 其中 $f_i = \{f_i^1, f_i^2, \dots, f_i^m\}$, f_i^m 表示第 i 个配置元的第 m 个特征属性, 与需求元中的特征属性对应, 特征属性是对配置元功能、性能和结构等方面的概括, 对应着客户某方面的需求。

3) $T = \{0, 1, 2\}$, 为选项型变量, 当 $T=0$ 时表示该配置元为必选单元, 配置时必须选择; 当 $T=1$ 时表示该配置元为可配置单元, 需要从配置元值域 D 中选择一个替换当前配置元; 当 $T=2$ 时表示该配置元为可选单元, 依据客户需求决定是否选择该配置元。

4) $D_i = \{d_i^1, d_i^2, \dots, d_i^m\}$, d_i^m 表示第 i 个配置元值域的第 m 个取值, d_i^1 为默认值, 特别地当 $T=0$ 时 $n=0$, 即 $D=\emptyset$; $T=2$ 时 $n=1$ 。

5) $C = \{Cobject, Ctype, Value\}$ 是一个约束集合, $Cobject$ 代表约束主体, 指对该配置元产生约束的各个对象, 包括客户需求及其他配置元; $Ctype$ 指约束类型, 约束类型包含关联约束、相容约束、互斥约束、可选约束和数量约束等。

3.3 需求元-配置元约束网络

谓词逻辑是一种重要的知识表示方法, 约束是由约束变量及其约束关系构成, 可以通过如下谓词和函数定义。

- Confirm(x, X): 确定 X 中 x 为当前取值;
- Reject(y, Y): 从集合 Y 中剔除子集 y ;
- Choose(x): 选中 x ;
- Get(D', D): 从集合 D 中得到子集 D' ;
- Number(x, n): x 的数量为 n ;
- $D(x)$ 和 $d(x)$ 分别表示 x 的值域和 x 的具体取值;
- $D'(x)$ 表示 $D(x)$ 的子集;

则关联约束:

$$C_L = (\exists D(R_i)) (\exists D(E_j)) \wedge \text{Confirm}(d(R_i), D(R_i)) \rightarrow \text{Get}(D'(E_j), D(E_j));$$

相容约束:

$$C_R = (\exists D(E_p)) (\exists D(E_q)) \wedge \text{Confirm}(d(E_p), D(E_p)) \rightarrow \text{Get}(D'(E_q), D(E_q));$$

互斥约束:

$$C_c = (\exists D(E_i)) (\exists D(E_j)) \wedge \text{Confirm}(d(E_i), D(E_i)) \rightarrow \text{Reject}(D'(E_j), D(E_j));$$

可选约束:

$$C_x = (\exists D(R_i)) (\exists E) \wedge \text{Confirm}(d(R_i), D(R_i)) \rightarrow \text{Choose}(E);$$

数量约束:

$$C_s = (\exists D(E_i)) (\exists d(E_j)) \wedge \text{Confirm}(d(E_i), D(E_i)) \rightarrow \text{Number}(d(E_j), n);$$

其中 R_i 表示第 i 个需求元, E_j 表示第 j 个配置元。

在这里用图的方式表达, 约束变量和图节点存在对应关系, 约束和边存在对应关系, 将变量作为节点, 约束表示为节点间的边, 根据变量与约束之间的相互关系的连接构成配置问题的约束网络图(见图 3), 下面给出约束网络的构造步骤。

- 步骤 1: 提取产品配置模型中的需求元、特征属性、配置元得到约束网络中的 R, F, E 等节点;
- 步骤 2: 将约束变量 R, F, E 表示为节点, 将变量间的约束作为边;
- 步骤 3: 对需求元按权重值排序, 由权重值最高的需求元开始, 参照关联约束, 依次连接 $R-F-E$;
- 步骤 4: 参照配置元与配置元之间的具体约束, 连接 $E-E$ 边构成需求元-配置元约束网络。

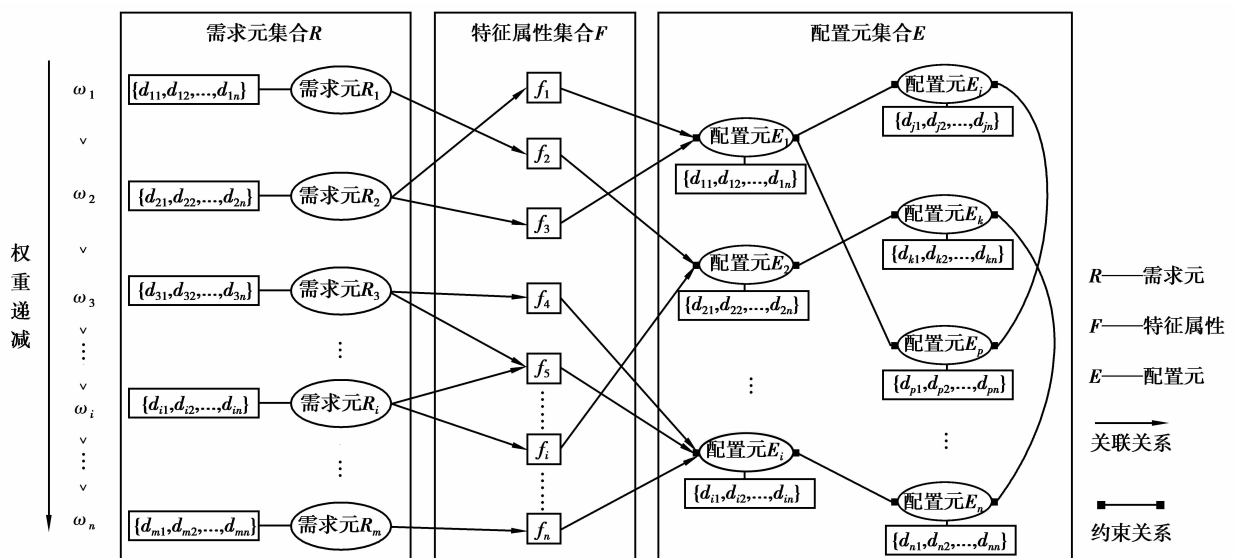


图 3 需求元-配置元约束网络映射模型

如图3所示,节点 R 表示需求元,节点 F 表示特征属性,节点 E 表示配置元连接节点 R 、 F 、 E 的 $R_i - f_j - E_k$ 边表示 R_i 与 E_k 通过特征属性 f_j 构成关联约束,记为 $h(R_i, f_j, E_k)$;连接两个配置元节点 $E_m - E_n$ 边表示 E_m 与 E_n 构成其他约束,记为 $C(E_m, E_n)$ 。

3.4 需求元-配置元约束网络求解算法

约束满足问题(CSP-Constraint Satisfaction Problem)求解通过对一组给定值域的变量设定一组有限的约束,每个约束规定了这组变量允许取的域值组合,产品配置是根据客户需求通过取各配置元中的不同域值进行组合,并满足预先定义配置元间的所有约束,从而生成各种不同的产品配置的过程,可以将配置问题映射到CSP中的变量和值域,对配置问题的求解转换为等价的CSP求解;但是传统的CSP求解算法针对单个客户需求时,会不必要的增大搜索空间和时间开销,笔者首先按需求元权重值进行关联约束传播,通过预处理技术缩小与需求元有约束关系的配置元值域空间,减小CSP问题规模,然后基于非二元弧一致求解剩余配置元获得配置结果集。

假设存在 m 个需求元与 n 个配置元,它们之间的约束关系矩阵可以表示为

$$\mathbf{M}_R = \begin{bmatrix} f(R_1, E_1) & \cdots & f(R_1, E_n) \\ \vdots & f(R_i, E_j) & \vdots \\ f(R_m, E_1) & \cdots & f(R_m, E_n) \end{bmatrix},$$

式中: R_i 为第 i 个客户需求元; E_j 为第 j 个配置元; D_j 为第 j 个配置元的值域。

$f(R_i, E_j)$ 中的 f 是关联约束函数,表示 R_i 通过第 j 个配置元 E_j 的关联约束对 D_j 作用后的集合,特别地当 E_j 与 R_i 不存在关联约束关系时, $f(R_i, E_j) = D_j$;则第 j 个配置元值域 D_j 经过一级约束网络求解后 $D'_j = \bigcap_{i=1}^m f(R_i, E_j)$;假设 $D'_j = \{d_1, d_2, \dots, d_n\}$,当 $n = 0$ 时,即 $d_j = \emptyset$ 表示客户需求间存在冲突,需要调整;当 $n = 1$ 时表示该配置元取值已确定;当 $n > 1$ 时表示该配置元取值未完全确定。将已经确定取值的配置元与必选单元称为已配置元,取值仍未确定的配置元称为未配置元。配置推理流程如下:

STEP1,如果未配置元集合为空,输出求解结果,结束程序;否则,选取未配置元集合中一个配置元为当前配置元,从未配置元集合中剔除该配置元。

STEP2,选取当前配置元值域中一个域值为当前配置元的取值,从当前值域中剔除当前取值,更新

当前配置元值域。

STEP3,对当前配置元约束进行非二元弧一致检验,如果发现未配置元集合中有配置元值域为 \emptyset ,则当前配置元取值不合适,转STEP4;如果一致性检查通过,将当前配置元并入已配置元集合中,转STEP1。

STEP4,如果当前配置元值域为 \emptyset 且未配置元集合为 \emptyset ,则此配置问题无可行解,提示用户修改需求,如果当前配置元值域不为 \emptyset ,转STEP2;如果当前配置元值域为 \emptyset 且未配置元集合非空,转STEP5。

STEP5,回溯到上一次赋值的配置元,令该配置元为当前配置元,转STEP2。

经过以上约束求解,可以得到最终配置结果集,按价格进行排序提供给用户选择,完成配置过程。

4 算例验证与应用实例

为了验证配置元模型与算法的有效性,选取半挂牵引车底盘作为配置实例进行验证,由于篇幅限制,案例中对数据进行了适当的简化。假设客户需要自重轻、省油高效、动力强劲、驾乘舒适的半挂牵引车底盘的产品,经转换可以将用户需求表达为 $\{Z$ 系列驾驶室、高驾驶室顶棚、后桥速比为5.286 $\}$,如图4所示底盘产品的配置模型,其中驾驶室系统、风窗系统、前围系统、变速箱系统、后桥系统及传动轴系统为虚拟节点,仪表系统与空调系统为必选单元。

以驾驶室系统为例, $D' = \{III$ 系列高顶, Z 系列高顶 $\} \cap \{Z$ 系列高顶, Z 系列平顶 $\} = Z$ 系列高顶,同理可得 E_3 。选取风窗系统(高顶), E_4 选取前围系统(Z 系列)。此时仍然有变速箱系统与传动轴系统没有确定,问题转化为约束满足问题,未配置元集合为 $\{$ 变速箱系统,传动轴系统 $\}$,已配置元集合为 $\{$ 仪表系统,驾驶室系统,风窗系统,前围系统,后桥系统,空调系统 $\}$;当变速箱系统选择HY7-130B04时,进行非二元弧一致检查,与后桥系统(速比5.286)不相容,所以变速箱系统选择8JS125TA(G6784),同理传动轴系统选择(15 000 端面齿)8JS125TA箱用传动轴系统;于是得到客户产品配置为 $\{$ 仪表系统,驾驶室系统(Z 系列高顶),风窗系统(高顶),前围系统(Z 系列),8JS125TA(G6784)变速箱,后桥系统(速比5.286),(15 000 端面齿)8JS125TA箱用传动轴系统,空调系统 $\}$;图5为在快速响应客户的产品开发平台上的实际产品配置过程。

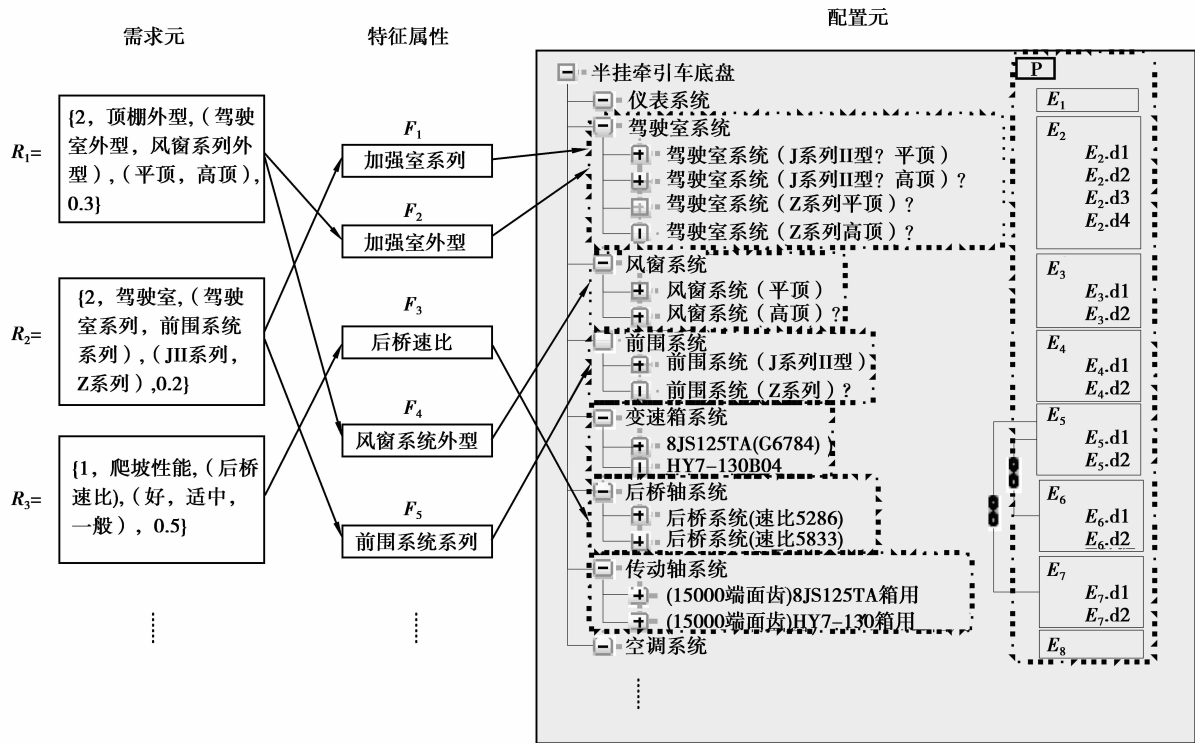


图 4 某半挂牵引车底盘配置模型



图 5 快速响应客户的产品开发平台中产品配置过程

5 结束语

针对面向产品族的产品配置方法,通过分析客户需求与产品族的特点建立了客户需求族模型、产品族配置模型;引入了需求元与配置元的概念,用谓词逻辑公式表示各种约束关系,构建了客户需求元与产品族配置元间、配置元与配置元间基于特征属性匹配的约束网络模型;通过预处理缩小与需求元有关约束关系的配置元值域空间,减小了配置求解的搜索空间和时间,采用非二元弧一致回溯算法对于未确定的配置元进行约束求解,得到配置结果

集。以半挂牵引车底盘为例,给出了配置模型在产品配置设计过程中的实际应用。下一步将在配置结果优化方面,综合考虑工艺、成本、质量、供应链等因素进行深入探索和研究。

参考文献:

[1] 付启敏,刘伟. 基于常用配置元的在线产品配置方法[J]. 计算机集成制造系统, 2010, 16(8): 1607-1613.
 FU QI-MIN, LIU WEI. Product configuration method based on the common configuration elements [J]. Computer Integrated Manufacturing Systems, 2010, 16(8): 1607-1613.

- [2] LI Y, CHEN Y. An apriori-based knowledge mining method for product configuration design[J]. *Advanced Materials Research*, 2010, 139-141: 1490-1493.
- [3] 轩辕思思, 姜兆亮, 李岩, 等. 基于双层配置体系的产品配置算法研究[J]. *计算机集成制造系统*, 2009, 15(5): 893-899.
XUANYUAN SI-SI, JIANG ZHAO-LIANG, LI YAN, et al. Product configuration solution based on double layer configuration system[J]. *Computer Integrated Manufacturing Systems*, 2009, 15(5): 893-899.
- [4] OSTROSI E, FOUGERES A J. Optimization of product configuration assisted by fuzzy agents[J]. *International Journal on Interactive Design and Manufacturing*, 2011, 5(1): 29-44.
- [5] ZENG F S, JIN Y. Study on product configuration based on product model [J]. *The International Journal of Advanced Manufacturing Technology*, 2007, 33(7/8): 766-771.
- [6] 李妮娅, 张健, 刘大有. 基于广义产品结构的产品配置[J]. *计算机集成制造系统*, 2010, 16(1): 17-23.
LI NI-YA, ZHANG JIAN, LIU DA-YOU. Product configuration based on generic product structure[J]. *Computer Integrated Manufacturing Systems*, 2010, 16(1): 17-23.
- [7] YANG D, MIAO R, WU H W, et al. Product configuration knowledge modeling using ontology web language[J]. *Expert Systems with Applications*, 2009, 36(3): 4399-4411.
- [8] XU Z W, LIANG Z Z, SHENG Z Q. Extended object model for product configuration design [J]. *International Journal of Automation and Computing*, 2010, 7(3): 289-294.
- [9] 但斌, 覃燕红, 王江平. 基于实例和 CSP 的产品配置方法[J]. *重庆大学学报*, 2008, 31(5): 511-514.
DAN BIN, QIN YAN-HONG, WANG JIANG-PING. Product configuration method based on an authentic real case and a constraint satisfaction problem[J]. *Journal of Chongqing University*, 2008, 31(5): 511-514.
- [10] FELFERNIG A, MANDL M, TIIHONEN J, et al. Personalized user interfaces for product configuration[C]// *Proceedings of the 15th ACM International Conference on Intelligent User Interfaces*, February 7-10, 2010, New York, USA. [S. l.]: Association for Computing Machinery, 2010: 317-320.
- [11] 单汨源, 袁际军. 基于非二元约束满足的配置问题求解方法[J]. *计算机工程*, 2008, 34(7): 205-208.
SHAN MI-YUAN, YUAN JI-JUN. Configuration problem solving method based on non-binary constraint satisfaction[J]. *Computer Engineering*, 2008, 34(7): 205-208.
- [12] BESSIERE C, MESEGUER P, FREUDER E C, et al. On forward checking for non-binary constraint satisfaction[J]. *Artificial Intelligence*, 2002, 141(1/2): 205-224.
- [13] FALKNER A, HASELBOCK A, SCHENNER G, et al. Modeling and solving technical product configuration problems[J]. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2011, 25: 115-129.
- [14] RAFFAELI R, MENGONI M, GERMANI M, et al. An approach to support the implementation of product configuration tools [C]// *Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, August 30-September 2, 2009, San Diego, California, USA. [S. l.]: ASME Digital Library, 2010, 5: 559-570.
- [15] MENDONCA M, COWAN D, MALYK W, et al. Collaborative product configuration: formalization and efficient algorithms for dependency-analysis[J]. *Journal of Software*, 2008, 3(2): 69-82.

(编辑 张 莘)