DOI: 10.3901/JME.2021.09.191

考虑变更风险的复杂产品模块划分方法^{*}

李玉鹏 谢卫星 王晓琳 任海兵
(中国矿业大学工业工程系 徐州 221116)

摘要: 模块化方法可有效控制复杂产品的研制周期与成本, 模块划分是实现模块化的基础。受顾客需求、关键技术及政策、市场等内外部动态因素的影响, 持续演化是复杂产品的必然趋势。设计变更是企业实现产品演化的基本手段, 为了降低演化的影响, 在对复杂产品进行模块划分时, 同时考虑功能、结构、变更风险三个因素。首先将复杂网络理论应用于复杂产品结构建模, 以零部件为节点, 零部件的关联关系为边, 建立复杂产品零部件网络模型。然后基于 LinkRank 的社区发现算法求解出产品模块初始划分方案。其次再利用基于网络相似度优化的社区发现算法得到最终划分方案。最后以某型号摩托车发动机为例验证了所提方法的可行性和有效性。

关键词: 复杂产品; 产品演化; 模块划分; 变更风险; 网络相似度

中图分类号: TP393

Module Partition Method for Complex Products Considering the Change Risk

LI Yupeng XIE Weixing WANG Xiaolin REN Haibing

(Department of Industrial Engineering, China University of Mining and Technology, Xuzhou 221116)

Abstract: Modularization method can effectively control the cost of research and development for complex products, and module partition is the foundation of modularization. Impacted by the dynamic internal and external factors including customer requirements, key technologies, policy and market, continuous evolution is unavoidable for complex products. Design change is the basic means for an enterprise to establish product evolution. In order to reduce the impact of evolution, three factors of function, structure and change risk are considered in the module partition of complex products. Firstly, the complex network theory is applied to the structure modelling of complex products, with components as nodes and component relationships as edges, the network model of the components for complex products is established. Then the LinkRank community discovery algorithm is used to obtain the initial module partition scheme of the product. Afterwards, the network similarity-based community discovery algorithm is used to obtain the final partitioning scheme. Finally, a case study of a motorcycle engine is presented to verify the feasibility and validity of the proposed approach.

Key words: complex products; product evolution; module partition; change risk; network similarity

0 前言

源于国家战略和重大行业需求, 工程机械、动车组、运载火箭乃至综合采煤装备等大型复杂产品在社会发展中的作用日益显著。相较于传统大规模定制产品, 复杂产品具有结构复杂、规模大、成本高、技术密集及高度定制化等典型特点^[1], 研制阶

段决定了其整体性能和总成本。复杂产品常面临多种不确定因素的影响^[2], 迫使其需要及时改进相关功能特性和物理结构加以应对。相较于产品整体, 这种改变往往是局部的细微变化。当外部因素变化累积到一定程度时, 则必须要对关键功能和零部件进行重新设计, 这一过程称为“产品演化”。尽管影响产品演化的因素复杂多变, 但可概括为顾客需求因素、关键技术因素及政策、市场等其他因素三方面。首先, 顾客需求的动态变化驱动企业对相关产品功能及结构进行改进, 以提高顾客满意度; 其次, 关键技术的进步客观上促使企业改变产品核心功能

^{*} 国家自然科学基金(51505480, 51475290)和江苏省自然科学基金(BK20150197)资助项目。20200605 收到初稿, 20210203 收到修改稿

的实现原理及物理结构,以提升产品竞争力^[3];此外,随着国家政策及市场环境对产品要求的不断提高(如:产品的低能耗、低排放、低污染等要求),迫使产品必须持续演化以符合市场标准。可见,在上述多重不确定因素的驱动下,复杂产品演化是其必然趋势,是产品成熟的必由之路。

复杂产品演化过程中,无论是局部的改进设计还是整体的变形设计,本质上都要依托设计变更技术实现^[4]。如图 1 所示,顾客需求、关键技术等驱动因素的持续变化,会带来相关功能变更需求,进而导致相关零部件的设计变更。由于复杂产品零部件间存在多重耦合关系,设计变更会以不同的风险水平向下游零部件传播,当传播累积到一定程度时,则形成新型产品。上述过程随着时间的推进不断迭代,推动着产品的更新换代,即产品演化。可见,设计变更是实现产品演化的基本手段。因此,在研发规划阶段增加对设计变更风险的考虑,对于复杂产品的演化控制,促进其快速成熟以响应各类变化因素具有重要意义。

目前,复杂产品设计逐渐普及模块化方法,可有效解决设计过程中子系统和零部件标准化程度低、组织交互频繁难以实现并行设计等难题。其中,模块划分是实现模块化设计、配置、制造和服务的基础^[5],根据应用背景的不同,分为面向新产品设计的模块划分和面向已有产品的模块划分^[6]。面向新产品设计的模块划分以顾客需求分析和产品功能分解为理论基础。LI 等^[7]基于权重有向网络构建了

复杂产品零部件功能关联模型,然后利用社区发现算法求解模型,得到模块划分结果;STONE 等^[8]采用功能基和时间序列功能链对产品功能进行统一描述,提出了一种三阶段启发式方法进行模块划分;高飞等^[9]建立了基于广义有向图和改进质量屋的产品功能模型,给出了基于模糊聚类的模块划分方法。面向已有产品的模块划分以零部件相关性分析为基础。部分研究中,零部件关联关系的描述工具,可用于功能模块划分中功能关联关系的表达。包括:ERIXON 等^[10]提出应用关联矩阵量化表达零部件之间的关联关系;BEEK 等^[11]研究了基于设计结构矩阵的功能关系建模方法,采用 k 均值聚类算法实现了机电产品的模块划分;QIAO 等^[12]在考虑产品全生命周期因素的基础上,设计了一种基于多维缩放和矩阵聚类的混合方法得到最优的产品模块化结构;樊蓓蓓等^[13]提出了一种基于复杂网络统计学特征的产品族模块化分析方法;侯文彬等^[14]提出一种改进的图分解算法,进而定义出车身装配结构的最佳划分方式,并依此进行车身结构模块的划分;郑维强等^[15]基于模糊关联确定零部件间的关联关系,并结合优化的粒子群算法对构建的复杂产品模块化数学规划模型进行求解,得到最优的模块化设计方案。然而以上研究均是在静态环境下,面向特定产品(功能)结构,通过对产品设计过程、功能、零部件的标化重组探求产品在品种与成本、性能间的平衡点,并未考虑复杂产品的动态演化背景。

与传统大规模制造产品相比,复杂产品具有高度复杂和深度耦合的特点,表现为由大量定制化的系统和子系统通过复杂的物理、能量和信息接口组合而成,具有显著的层次化和非线性特征;子系统数量庞大,系统要素间存在多重耦合关系,包括设计参数耦合、物理结构耦合、研发任务耦合、研发组织耦合等^[16]。而产品演化体现为设计变更,对单个或多个零部件,由于要素多重耦合关系将初始的设计变更传播至多个模块甚至整个产品,导致昂贵的结构重调和漫长的再设计周期。现有模块化方法大都着眼功能、结构及同期产品生命周期因素,而缺乏对产品演化性能,即变更风险的考量,模块化系统结构无法有效支持其演化过程控制。因此需要在考虑演化影响的基础上对复杂产品架构进行模块化重构。

综合以上分析,提出了一种考虑变更风险的复杂产品模块划分方法。首先根据零部件间的功能、结构、变更风险关联程度构建对应的复杂网络模型;

然后运用基于 LinkRank 的社区发现算法求解得到模块初始划分方案；再利用基于网络相似度优化的社区发现算法求解得到最终划分方案。所得方案可在满足功能和结构相关性约束的基础上，有效应对产品的持续演化及突变变更，缩短改进设计和变形设计的周期。

1 复杂产品零部件网络模型构建

1.1 复杂网络

复杂网络可以用一个三元组来表示，即 $CN=(V, E, W)$ ，其中： $V=(v_1, v_2, \dots, v_n)$ 表示节点集合， $E=(e_1, e_2, \dots, e_m)$ 表示边集合， $W=(w_1, w_2, \dots, w_k)$ 表示权重集合。为方便计算处理，复杂网络通常用一个 $n \times n$ (n 为节点数量)的邻接矩阵 A 表示，矩阵 A 是由元素 A_{ij} 组成，取值规则为

$$A_{ij} = \begin{cases} w_{ij}, & \text{节点 } i \text{ 和节点 } j \text{ 相连} \\ 0, & \text{其他} \end{cases} \quad (1)$$

式中， w_{ij} 表示边的权重。

1.2 复杂网络中的社区结构

对于复杂网络，GIRVAN^[17]认为网络节点间连接边关系的不均匀造成了其局部聚集特性，即复杂网络可分割为多个子网络，同一子网络中节点间的边连接较为紧密，不同子网络中节点间的边连接较为稀疏，相应地，每个子网络被称为一个社区。图 2 是一个具有三个社区结构的复杂网络实例。

图2 复杂网络的社区结构示意图

为了评判复杂网络社区结构的“好坏”程度，NEWMAN^[18]提出用网络模块度 Q 来衡量社区划分的质量，其具体计算公式如下

$$Q = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \quad (2)$$

式中， m 为网络边数； A_{ij} 为网络连接矩阵中的元素； $k_i = \sum_j A_{ij}$ ； c_i 为节点 i 所在的社区；如果节点 i 和

节点 j 属于同一社区，函数 $\delta(c_i, c_j)=1$ ，反之为 0。网络模块度 Q 的取值是介于 0 到 1 之间的自然数。

1.3 构建零部件复杂网络模型

首先对零部件间的功能、结构和变更风险相关性进行分析，给出量化准则，分别构建相应的复杂网络模型。

零部件之间的功能关联关系主要分为两类：对功能分解形成的功能从属关系和零部件相互作用产生的输入输出关系。YU 等^[19]将产品功能分为功能和子功能的层次化结构，并通过分析两零部件对其上级功能的贡献程度来确定其功能关联值。在此基础上，结合两零部件间的输入输出关系可以确定二者的功能关联强度，如表 1 所示。

表1 零部件功能关联强度评价标准

关联强度	功能关联值	描述
强	1.0	两零部件均为同一功能下的从属功能，且其中一零部件功能对另一零部件功能起辅助作用
中	0.6	两零部件均为同一功能下的从属功能，二者存在输入输出关系，且均为上级功能实现的必要功能
弱	0.2	两零部件属于不同功能的从属功能，但二者之间存在输入输出关系
无	0	两零部件属于不同功能的从属功能，且二者之间不存在输入输出关系

产品功能是通过物理结构所实现的，功能的实现和物理结构紧密相连。零部件之间的结构关联关系主要包括接触方式、结合方式、结合所使用工具类型及连接自由度等四个方面，其量化评价标准如表 2 所示。这四个评价指标被赋予相同的权重因子，两零部件之间的最终结构关联值可通过对上述评价指标的取值进行加权求和获取。

表2 零部件物理结构关联强度评价标准

关联强度	结构关联值	接触方式	结合方式	工具类型	连接自由度
极强	1.0	多面接触	不可拆卸	大型工具	1 自由度
强	0.8	多点接触	可拆卸	特制工具	2 自由度
中	0.6	单面接触	旋入	小型工具	3 自由度
弱	0.4	线接触	插入	螺丝刀	4 自由度
极弱	0.2	点接触	放置	手工	5 自由度

CLARKSON 等^[20]利用设计结构矩阵分析了在产品设计中零部件的变更对其他零部件的变更影响和变更可能性，并预测了变更产生的传播风险。图 3 展现了变更影响、变更可能性及变更风险之间的关系。

图 3 变更影响、变更可能性和变更风险关系图

定义 1: 零部件变更影响表示当零部件 i 的变更传播到零部件 j 时, 其受到影响需要重做的设计工作的比例(表示为 p_{ij})。

定义 2: 零部件变更可能性表示当零部件 i 发生变更时, 与之具有变更联系的零部件 j 发生变更的概率(表示为 l_{ij})。

定义 3: 零部件变更风险表示零部件 i 和 j 之间变更传播的风险, 等于零部件变更影响和变更可能性的乘积(表示为 r_{ij})。

变更影响的取值由相关设计人员参考产品变更的历史记录得出, 取值范围为 $[0, 1]$ 。变更可能性 l_{ij} 由产品设计变更数据库通过计算条件概率获取

$$l_{ij} = P(v_j | v_i) = \frac{P(v_i \cap v_j)}{P(v_i)} = \frac{P(v_j | v_i) P(v_i)}{P(v_i)} = \frac{P_{ji} P(v_i)}{P(v_i)} \quad (3)$$

式中, $P(v_i)$ 为节点 i 对应的零件发生变更的概率。再根据定义 3 可计算出变更风险关联值

$$r_{ij} = p_{ij} \times l_{ij} \quad (4)$$

在对零部件间的功能、结构及变更风险关联分析后, 将零部件抽象成复杂网络中的节点, 零部件间的联系抽象成边(包括有向边和无向边), 零部件间的关联值作为边的权重, 分别构建零部件复杂网络模型, 包括功能网络模型 $CN_{function}$ 、结构网络模型 $CN_{structure}$ 和变更风险网络模型 CN_{change} , 如图 4 所示。

图 4 零部件复杂网络模型

其中, $CN_{function}$ 和 $CN_{structure}$ 为无向加权网络, CN_{change} 为有向加权网络, 其边的方向是从零部件自身指向受其影响的零部件。此外, 各网络模型中边的权重定义如下

$$w_{ij} = \begin{cases} f_{ij}, & \text{在 } CN_{function} \text{ 中} \\ s_{ij}, & \text{在 } CN_{structure} \text{ 中} \\ r_{ij}, & \text{在 } CN_{change} \text{ 中} \end{cases} \quad (5)$$

式中, f_{ij} 为零部件 i 和零部件 j 之间的功能关联值; s_{ij} 为零部件 i 和零部件 j 之间的结构关联值; r_{ij} 为零部件 i 和零部件 j 之间的变更传播风险。

2 考虑变更风险的模块划分

产品模块划分通常是在分析零部件间的关联因素后, 得到其综合关联关系后再进行模块划分。但在实际问题中, 由于零部件变更风险关联的取值相对功能和结构关联的取值较小, 而传统赋权方法难以避免决策者主观性的影响, 导致模块划分方案不合理, 后续实例分析中具体展示该现象。为了避免这种现象的出现, 先分别对各关联关系求解得到模块初始划分方案, 再以各初始方案与最终方案的网络综合相似度最高为目标对后者进行优化, 如图 5 所示。

图 5 考虑变更风险的模块划分方法框架

2.1 基于 LinkRank 和社区发现的模块初始划分

从上述构建的三个复杂网络模型中进行社区发现需要考虑三个问题：评价社区发现结果质量的目标函数；有效识别最优社区发现结果的优化方法；该方法是否适用于有向网络和无向网络。

式(2)中定义的模块度是用来评估复杂网络社区发现结果质量的常用目标函数，但其只适用于无向网络。在此基础上，KIM 等^[21]提出了一种基于 LinkRank 的模块度来评价有向网络社区发现结果。其中，LinkRank 是指网络中边的重要度，表示在固定状态下，网络中的随机步骤从节点 i 到节点 j 的概率，可通过网络节点重要度 PageRank 计算得出。PageRank 被定义为在某个网页上随机点开一个超链接，它到达特定页面的概率^[28]。计算 PageRank 的公式如下

$$\pi^T = \pi^T G \quad (6)$$

式中， π^T 为矩阵 G 的固定行矢量，也叫 PageRank 矢量， π^T 中的每个元素 π_i 表示在该固定状态下，随机步骤拜访节点 i 的概率^[21]。矩阵 G 被称为 Google 矩阵，它表示网络中随机步骤转移的概率矩阵，其中每个元素 G_{ij} 表示随机步骤从节点 i 移动到节点 j 的概率。Google 矩阵被定义为

$$G_{ij} = a \frac{w_{ij}}{w_i^{out}} + \frac{1}{N}(\alpha a_i + 1 - \alpha) \quad (7)$$

式中， w_{ij} 为网络邻接矩阵中的元素； w_i^{out} 为节点 i 的出度； α 为跟随超链接打开网页的概率； N 为网页数量；此外，当节点 i 为悬挂节点时， $a_i=1$ ，反之 $a_i=0$ ；当 $w_i^{out}=0$ 时， $w_{ij}/w_i^{out}=0$ ；在获取网络节点重要度 PageRank 后，可计算网络边的 LinkRank 值

$$L_{ij} = \pi_i G_{ij} \quad (8)$$

式中， π_i 为 PageRank 矢量中第 i 个元素的值， G_{ij} 为 Google 矩阵中元素的值。

KIM 等基于 LinkRank 将社区定义为一组随机游走更容易停留的节点^[21]，与传统模块度定义不同，其提出的模块度定义如下

$$Q_{linkrank} = \sum_{i,j} [L_{ij} - \pi_i \pi_j] \delta(c_i, c_j) \quad (9)$$

式中， L_{ij} 为节点 i 和节点 j 之间 LinkRank 值， $\pi_i \pi_j$ 为从节点 i 移动到节点 j 随机游走的概率；如果节点 i 和节点 j 属于同一社区，函数 $\delta(c_i, c_j)=1$ ，反之为 0。同时，KIM 等^[21]证明了这种模块度定义适用于无向

网络和有向网络。

LinkRank 社区发现算法本质上是对网络模块度(式(8))的优化问题，传统的优化算法，例如贪心算法^[23]和快速展开算法^[24]都可以用来解决这个问题。但复杂网络的复杂性容易使得其陷入局部最优解，故采用一种全局优化算法——模拟退火算法(Simulated annealing, SA)^[25]。

LinkRank 社区发现算法主要由 LinkRank 值获取和网络社区发现两部分构成。算法流程如图 6 所示。通过上述的 LinkRank 社区发现算法，可以分别对 1.3 中构建的 $CN_{function}$ 、 $CN_{structure}$ 和 CN_{change} 网络模型求解，得到三个产品模块初始划分方案。

图6 LinkRank 社区发现算法流程图

2.2 基于网络相似度优化的模块再划分

由于考虑的关联因素不同，单独考虑零部件间功能、结构和变更风险的关联得到的产品模块初始划分方案显然各不相同，最终的综合划分方案应尽量和三个初始划分方案相接近。为了衡量初始划分方案和综合划分方案的接近程度，提出了网络综合相似度(S_{com})的概念，通过对其进行优化来获取最终划分方案， S_{com} 计算过程如下。

步骤 1：将三个初始划分方案分别转换为三个无向网络 $I_k(k=1, 2, 3)$ ，网络中的边权定义如下

$$w_{ij} = \begin{cases} 1, & \text{节点 } i \text{ 和 } j \text{ 在同一个社区} \\ 0, & \text{其他} \end{cases} \quad (10)$$

特别地, 当某社区只有一个节点 i 时, $w_{ii}=1$ 。

步骤 2: 对于两个节点数目相同的无向网络, 其相似程度可以转换为其邻接矩阵间的相似程度。这里以 Euclidean 距离定义单个网络 I_k 和综合网络 I 之间的相似度关系, 定义网络相似度 S_{CN} 计算如下

$$S_{CN} = \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{1}{n} \sum_{j=1}^n (I_{kij} - I_{ij})^2} \quad (11)$$

式中, n 为网络中节点的数量, I_{ij} 和 I_{kj} 分别为综合网络 I 和单个网络 I_k 中节点 i 和节点 j 之间的取值。

步骤 3: 综合相似度 S_{com} 被定义为三个初始划分方案和综合划分方案的综合接近程度, 如式(12)所示。

$$S_{com} = \sum_{k=1}^3 w_{I_k} \cdot S_{CN}^{I_k} \quad (12)$$

式中, w_{I_k} 为划分方案 I_k 的偏重因子, 该系数越大表明综合划分方案偏向于该划分方案的程度越高; $S_{CN}^{I_k}$ 为划分方案 I_k 和综合划分方案的相似程度。

类似地, 以综合相似度代替网络模块度作为评价社区发现结果的质量函数进行社区结构优化, 即可得到产品模块划分的最终结果, 其算法流程如图 7 所示。

图 7 网络相似度社区发现算法流程图

3 实例分析

摩托车发动机的研制综合了电气、材料、机械、系统工程等相关学科, 属于产品的核心部件。此外, 摩托车发动机易受顾客需求变化和技术更新等因素影响, 包括更强的动力、更低的油耗、更大的排量等, 进而导致产品演化, 例如排气量更大的需求使得气缸体的尺寸和材料发生变化, 从而需要对发动机再设计。因此, 选取摩托车发动机为例进行模块划分, 证明所提方法的有效性。

该发动机由汽缸盖、气门、气门固定螺母等 30 种零部件组成, 如图 8 所示。依照 1.3 节所述方法对零部件间 3 类关联关系进行赋值, 功能关联关系、结构关联关系、变更风险关联关系分别如图 9a、图 9b 及图 9c 所示。进而分别建立零部件功能、结构、变更风险网络模型, 如图 10 所示。

图 8 摩托车发动机示例图

1. 油道; 2. 气缸垫片; 3. 气缸体; 4. 喷油器; 5. 气缸套; 6. 气门固定轴; 7. 气门; 8. 气门固定螺母; 9. 气门调整螺钉; 10. 气环; 11. 挺杆导管; 12. 凸轮; 13. 凸轮轴; 14. 汽缸盖; 15. 连杆轴颈; 16. 连杆; 17. 连杆轴承; 18. 活塞销; 19. 油环; 20. 活塞; 21. 飞轮; 22. 曲轴后臂; 23. 曲柄销; 24. 曲轴前壁; 25. 滤网; 26. 限压阀; 27. 机油罩; 28. 曲轴箱; 29. 机油盘; 30. 机油泵

获取零部件各网络模型后, 利用基于 LinkRank 的社区发现算法对零部件进行模块初始划分, 根据图 5 的求解流程, 分别对 3 个网络模型进行计算。算法设定初始温度为 100 度、冷却速率为 0.99、终止温度为 0.1 度。由图 11 可知经过 25 次迭代, 3 次计算均收敛, 得到模块初始划分方案如表 3 所示。

[illegible]

(a) 功能关联关系矩阵

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0.3	0	0.3	0.6
2	0	0	0	0.8	0.2	0.6	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0
3	0	0.8	0	0.6	0.6	0	0	0	0	0	0	0	0.6	0.8	0	0	0	0	0	0.6	0	0	0	0	0.4	0	0	0	0.4	0
4	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0.6	0.6	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0.8	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0.8	0	0.8	0.8	0.9	0.8	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0.6	0.8	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0.8	0.5	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0.8	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0.4	0	0.6	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0.8	0.8	0.9	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.8	0.2	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0.6	0	0	0.3	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0.8	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.8	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0.3	0.8	0.3	0.9	0.9	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.6	0.4	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0.7	0.6	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.7	0	0.7	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.6	0.7	0	0	0	0	0	0	0
25	0.8	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0
28	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0.3	0	0.4	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0
30	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0

(b) 结构关联关系矩阵

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0
2	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0.1	0	0	0	0	0	0.2	0.2	0	0	0	0
3	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4
5	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1
6	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0.2	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0.4	0.1	0.2	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0.1	0	0	0.2	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0.2	0.2	0	0.2	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0.3	0.2	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0.4	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0.1	0.5	0.2	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0.5	0	0.3	0
15	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0.2	0.2	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0.4	0.1	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.2	0.2	0	0.1	0	0	0	0	0	0	0
19	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0.2	0.1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.3	0	0	0	0	0	0
22	0	0	0	0.3	0.2	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0.1	0.2	0	0	0.2	0	0	0	0.2	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0.2	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0.2	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0.1	0	0.4	0.2	0	0	0	0	0	0
29	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0
30	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0.3	0

(c) 变更风险关联关系矩阵

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
1	0	0.1	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0.1	0	0	0	0	0	0.4	0.3	0.2	0	0.3	0.4		
2	0.1	0	0.5	0.1	0.4	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0	0.2	0	0		
3	0	0.5	0	0.2	0.5	0	0	0	0	0	0	0	0.2	0.7	0	0	0	0	0	0.2	0	0	0	0	0.1	0	0	0.2	0.1	0.1		
4	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2		
5	0	0.4	0.5	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.1	0	
6	0	0	0	0	0	0	0.5	0.4	0.2	0.2	0.1	0.1	0.2	0	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0.5	0	0.6	0.5	0.6	0.4	0.2	0.2	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0.4	0.5	0	0.3	0.2	0.2	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0.2	0.5	0.4	0	0.2	0.3	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0.2	0.5	0.3	0.2	0	0.2	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0.1	0.5	0.1	0.2	0.2	0	0.2	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0.1	0.2	0.1	0.1	0.1	0.2	0	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0.1	0	0.2	0	0	0.1	0.2	0.1	0.1	0.1	0.2	0.7	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0.5	0.7	0.4	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0.2	0.4	0.2	0	0	
15	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0.6	0.4	0.2	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0.1	0.1	0	0.7	0	0.5	0.2	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.5	0	0.2	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.2	0.2	0	0	0.6	0.1	0	0	0	0	0	0	0	0	0	0	0
19	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	0.3	0.2	0.2	0	0.2	0.1	0	
20	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0.3	0.5	0.4	0.7	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.4	0.4	0	0	0	0	0	0	0	0
22	0	0	0	0.1	0.1	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0.3	0	0.4	0.4	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.4	0	0.5	0	0	0	0.1	0	0	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.4	0.5	0	0	0	0	0	0.1	0	0	0
25	0.4	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0.2	0.1	0	0.1	0.1	0
26	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0.2	0	0.2	0	0.2	0.5	0	0
27	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0.2	0	0	0.2	0	0	0.1	0.2	0	0	0.5	0.2	0	0
28	0	0.1	0.4	0	0.2	0	0	0	0	0	0	0	0	0.1	0	0	0.1	0	0	0	0	0	0.1	0.1	0	0	0	0	0	0	0	0
29	0.3	0	0.2	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0.2	0	0	0	0	0	0.1	0.2	0.6	0	0	0	0.2	0
30	0.4	0	0.1	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0.1	0	0	0	0	0	0.1	0.5	0.2	0	0.4	0	0	0

图 10 零部件网络模型

表 3 模块初始划分结果

模块序号	方案 1 (基于功能网络模型)	方案 2 (基于结构网络模型)	方案 3 (基于变更风险网络模型)
M1	2, 3, 5, 14, 28	2, 3, 4, 5, 14, 27, 28, 29	3, 4, 5, 14, 22, 27, 29, 30
M2	6, 7, 8, 9, 10, 11, 12, 13	6, 7, 8, 9, 10, 11	7, 8, 9, 10, 11
M3	21, 22, 23, 24	21, 22, 23, 24	17, 18, 20, 21, 23, 24, 28
M4	15, 16, 18, 17, 20	15, 16, 17, 18, 19, 20,	1, 2, 19, 25, 26
M5	4	1, 25, 26, 21, 30	6, 12, 13, 15, 16,
M6	1, 19, 25, 26, 27, 29, 30	12, 13	—
模块数	6	6	5
模块度	0.644 9	0.643 7	0.339 8

从表 3 可见，方案 1 中将喷油器(4)单独划分为一个模块，而喷油器通常安装在汽缸盖(14)上且容易与其一起发生变更，因此在方案 2 和方案 3 中和汽缸盖同属一个模块；方案 2 中机油盘(29)和曲轴箱(28)在同一模块，但机油盘的功能是收集和储存回流的润滑油，因此在方案 1 中和滤网(25)、机油泵(30)等组成发动机润滑模块；方案 3 中曲轴后臂(22)和汽缸盖被划分在同一模块，但这两零部件在功能和结构上联系较弱，因而在方案 1 和方案 2 中被划分为不同的模块。总之，在模块初始划分时，仅考虑零部件间单一联系得到的 3 个发动机初始模块划分方案都存在一定缺陷，接下来以初始划分方案为基础进行模块再划分。

图 11 模块度迭代图

依据式(10)将以上模块划分方案转化为三个无向无权网络，再由设计人员给出三个方案的权重 $w_{jk}=(0.3,0.3,0.4)$ ，并基于网络相似度的社区发现算法

获取模块划分最终方案。为说明方法的合理性和有效性，根据图 7 中的求解流程，分别对考虑和不考虑变更风险两种情况进行计算。其中，综合相似度的变化如图 12 所示，经过 30 次迭代 2 次计算均收敛，最终结果如表 4 所示。

图 12 综合网络相似度迭代图

表 4 模块划分结果

模块名称	不考虑变更风险	考虑变更风险
M1	2, 3, 5, 14, 28	3, 4, 5, 14, 27, 28, 29, 30
M2	6, 7, 8, 9, 10, 11	6, 7, 8, 9, 10, 11
M3	21, 22, 23, 24,	21, 22, 23, 24
M4	15, 16, 17, 18, 19, 20	1, 2, 19, 25, 26
M5	4	15, 16, 17, 18, 20
M6	1, 25, 26, 27, 29, 30	12, 13
M7	12, 13	—
模块数	7	6
模块度	0.643 9	0.596 3

从图 12 可知, 不考虑变更风险时, 得到的最终划分方案与初始划分结果的相似度更大, 而考虑变更风险后最终划分方案与三个初始划分方案相似度较小。这是因为在考虑变更风险后, 一些变更风险关联较大的零部件被划分至同一模块, 如 M1 中的喷油器(4)、机油盘(29)等。

接下来在变更传播涉及模块数量方面对上述两个结果进行对比。气缸体是摩托车发动机的重要零部件, 且在现实生活中容易发生变更, 故选取节点编号为 3 的气缸体为初始变更节点, 初始变更影响设置为 0.30, 运用文献[26]中的方法, 得到最短路径为 3→14→27→22→4→30。在不考虑变更风险时此次变更涉及的模块数量为 4 个(M1, M3, M5, M6), 而在考虑变更风险后涉及的模块数量仅为 2 个(M1, M3)。究其原因是在变更的传播往往发生在变更风险关联较大的零部件间, 在模块划分时考虑变更风险这一因素会使得易于发生变更传播的零部件被划分至同一模块, 从而减少变更所涉及的模块数量。随着产品复杂度的不断增加, 每次变更涉及的模块数量也会不断增加。在模块划分时考虑零部件间的变更风险关联关系, 可以将变更控制在少量的模块中, 有效地降低变更带来的影响, 提高设计的敏捷性。

表 5 方法对比结果

模块名称	传统模块划分方法	考虑变更风险的模块划分方法
M1	2, 3, 4, 5 14, 28	3, 4, 5, 14, 27, 28, 29, 30
M2	6, 7, 8, 9, 10, 11, 12, 13	6, 7, 8, 9, 10, 11
M3	21, 22, 23, 24	21, 22, 23, 24
M4	1, 19, 25, 26, 27, 29, 30	1, 2, 19, 25, 26
M5	15, 16, 17, 18, 20	15, 16, 17, 18, 20
M6	—	12, 13
模块数	5	6
模块度	0.564 5	0.596 3

下面, 对提出的考虑变更风险的模块划分方法和传统的模块划分方法进行对比。传统的模块划分方法是对零部件间各关联关系加权求和处理得到综合关联关系矩阵(图 9d), 再进行模块划分求解。运用第 2.1 节中的模块划分方法对综合关联关系矩阵求解得到模块划分方案, 如表 5 所示。对比结果发现个别零部件划分结果存在差异, 这是由于实际问题中变更风险关联关系整体取值较小造成的。以机油盘(29)为例, 从图 9 中可以看出与其相关的变更风险关联取值最大为 0.3, 而功能和结构关联的最大值为 0.8, 这使得 29 和 1、19、25、26 间的综合关联强度明显要强于和 3、14、22、5 的, 进而导致在模块划分时被划分至前者。而考虑变更风险的模块划分方法是单独划分完成后再利用网络相似度进行划分, 避免了该问题, 结果更为合理。

在上述理论方法基础上, 研究人员可构建产品模块化知识库, 例如: 规则库、结构库、关系库、方法库, 开发相应的计算机辅助模块化概念设计系统, 主要功能包括产品信息录入、产品结构建模与表达、产品模块划分及方案评价等, 以提高模块化设计效率, 减少人为操作及计算工作量。

4 结论

(1) 针对以往对产品演化背景下模块划分研究的不足, 在分析零部件关联关系时加入了变更风险这一因素, 并给出了量化准则, 同时建立了复杂产品零部件多网络模型。

(2) 提出了一种考虑变更风险及基于网络相似度优化的复杂产品模块划分方法。获取的模块划分方案能帮助设计人员更好地应对产品的持续演化及突发变更, 缩短改进设计和变形设计的周期。

(3) 用某型号摩托车发动机模块划分实例对上述方法的应用进行描述和验证。

所提方法仅从变更风险的角度对零部件间进行模块划分, 但并未考虑变更传播路径问题, 未来将在模块划分时将考虑变更传播路径的影响。

参 考 文 献

- [1] HANSEN K L, RUSH H. Hotspots in complex product systems: Emerging issues in innovation management[J]. Technovation, 1998, 18: 555-561.
- [2] ECKERT C, CLARKSON P J, ZANKER W. Change and

- customisation in complex engineering domains[J]. *Research in Engineering Design*, 2004, 15 (1): 1-21.
- [3] ORBACH Y, FRUCHTER G E. Forecasting sales and product evolution: The case of the hybrid/electric car[J]. *Technological Forecasting & Social Change*, 2011, 78(7): 1210-1226.
- [4] LI Yuliang, ZHAO Wei, ZHANG Jie. Resource-constrained scheduling of design changes based on simulation of change propagation process in the complex engineering design[J]. *Research in Engineering Design*, 2019, 30(1): 21-40.
- [5] 侯亮, 唐任仲, 徐燕申. 产品模块化设计理论, 技术与应用研究进展[J]. *机械工程学报*, 2004, 40(1): 56-61.
HOU Liang, TANG Renzhong, XU Yanshen. Review of theory, key technologies and its application of modular product design[J]. *Journal of Mechanical Engineering*, 2004, 40(1): 56-61.
- [6] 祁卓娅, 王建正, 韩新民. 模块柔性划分方法[J]. *机械工程学报*, 2007, 43(1): 87-94.
QI Zhuoya, WANG Jianzheng, HAN Xinmin. Flexible Module division method[J]. *Journal of Mechanical Engineering*, 2007, 43(1): 87-94.
- [7] LI Yupeng, WANG Zhaotong, ZHONG Xiaoyu, et al. Identification of influential function modules within complex products and systems based on weighted and directed complex networks[J]. *Journal of Intelligent Manufacturing*, 2018(5):1-16.
- [8] STONE R B, WOOD K L, CRAWFORD R H. A heuristic method for identifying modules for product architectures[J]. *Design Studies*, 2000, 21(1): 5-31.
- [9] 高飞, 肖刚, 潘双夏, 等. 产品功能模块划分方法[J]. *机械工程学报*, 2007, 43(5):29-35.
GAO Fei, XIAO Gang, PAN Shuangxia, et al. Method of product function module partition[J]. *Journal of Mechanical Engineering*, 2007, 43(5): 29-35.
- [10] ERIXON G, VON YXKULL A, ARNSTROEM A. Modularity-the basis for product and factory reengineering[J]. *CIRP Annals-Manufacturing Technology*, 1996, 45(1): 1-6.
- [11] BEEK T J V, ERDEN M S, TOMIYAMA T. Modular design of mechatronic systems with function modeling[J]. *Mechatronics*, 2010, 20(8): 850-863.
- [12] QIAO L, EFATMANESHNIK M, RYAN M, et al. Product modular analysis with design structure matrix using a hybrid approach based on MDS and clustering[J]. *Journal of Engineering Design*, 2017, 28(6):1-24.
- [13] 樊蓓蓓, 祁国宁. 基于复杂网络的产品族结构建模及模块分析方法[J]. *机械工程学报*, 2007, 43(3): 187-198.
FAN Beibei, QI Guoning. Product family structure modeling and module analysis method based on complex network[J]. *Journal of Mechanical Engineering*, 2007, 43(3): 187-198.
- [14] 侯文彬, 单春来, 于野, 等. 面向模块化平台的模块划分及共享模块筛选方法[J]. *机械工程学报*, 2018, 54(1):188-196.
HOU Wenbin, SHAN Chunlai, YU Ye, et al. Modular platform-oriented method for module identification and selection[J]. *Journal of Mechanical Engineering*, 2018, 54(1): 188-196.
- [15] 郑维强, 刘振宇, 刘达新, 等. 基于模糊关联的复杂产品模块化设计方法及其应用[J]. *机械工程学报*, 2015(5): 130-142.
JIA Weiqiang, LIU Zhenyu, LIU Daxin, et al. Modular design and application for complex product based on fuzzy correlation analysis[J]. *Journal of Mechanical Engineering*, 2015(5): 130-142.
- [16] BRAHA D, BAR-YAM Y. The structure and dynamics of complex product design[C]//Complex Engineered Systems. Springer Berlin Heidelberg, 2006: 40-71.
- [17] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. *Proceedings of the National Academy of Sciences*, 2002, 99(12): 7821-7826.
- [18] NEWMAN, M E J. Analysis of weighted networks[J]. *Physical Review E*, 2004, 70(5): 056131.
- [19] YU Suiran, YANG Qingyang, TAO Jing, et al. Product modular design incorporating life cycle issues-group genetic algorithm (GGA) based method[J]. *Journal of Cleaner Production*, 2011, 19(9):1016-1032.
- [20] CLARKSON P J, SIMONS C, ECKERT C. Predicting change propagation in complex design[J]. *Journal of Mechanical Design*, 2000, 136(8):52-68.
- [21] KIM Y, SON S W, JEONG H. Finding communities in directed networks[J]. *Physical Review E*, 2010, 81(1): 016103.
- [22] BRIN S, PAGE L. Reprint of: The anatomy of a large-scale hypertextual web search engine[J]. *Computer Networks*, 2012, 56(18): 3825-3833.
- [23] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. *Physical Review E*, 2004, 69(6): 066133.
- [24] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast unfolding of communities in large networks[J].

Journal of Statistical Mechanics: Theory and Experiment, 2008, 10: P10008.

- [25] KIRKPATRICK S, JR G C, VECCHI M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.
- [26] MA Songhua, JIANG Zhaoliang, LIU Wenping, et al. Design property network-based change propagation prediction approach for mechanical product development[J]. Chinese Journal of Mechanical Engineering, 2017, 30(3): 676-688.

作者简介: 李玉鹏, 男, 1985 年出生, 博士, 副教授, 硕士研究生导师。主要研究方向为复杂产品系统设计理论及群决策。

E-mail: ypeng_li@163.com

谢卫星, 男, 1994 年出生, 硕士研究生。主要研究方向为复杂产品模块化及设计变更影响分析。

E-mail: 1105413061@qq.com

王晓琳(通信作者), 女, 1977 年出生, 博士, 副教授, 硕士研究生导师, 主要研究方向为复杂系统设计、企业管理等。

E-mail: graciouswxi@126.com

任海兵, 男, 1978 年出生, 博士, 副教授, 硕士研究生导师。主要研究方向为复杂系统设计、生产计划与控制等。

E-mail: cumt_rhb@163.com