

# 城市基础设施项目群集成控制系统评价与优化

郑 弦<sup>1</sup>, 卢昱杰<sup>2</sup>, 李永奎<sup>1</sup>, 乐 云<sup>1</sup>

(1. 同济大学 经济与管理学院, 上海 200092; 2. 新加坡国立大学 设计与环境学院, 新加坡 117566)

**摘要:** 为了实现系统模块的合理划分和科学评价, 采用设计结构矩阵(design structure matrix, DSM)的理论和技術提出了城市基础设施项目集成控制系统模块化评价方法, 包括系统功能依赖强度、数量的确定以及传递成本和聚类成本的计算等. 针对该类系统的更新提出将遗传算法应用到 DSM 聚类算法中以寻找最优更新方案. 以长春基础设施项目群集成控制系统作为典型案例, 比较了更新前后两阶段系统版本的模块化程度, 并提出了新一代系统的更新方案, 该案例验证了 DSM 方法在系统模块划分、评价和方案优选方面的可行性和有效性, 为同类大型集成控制系统的开发和维护提供了理论和实践参考.

**关键词:** 项目群综合控制; 模块化; 设计结构矩阵

**中图分类号:** TU17

**文献标志码:** A

## Evaluation and Optimization of City Infrastructure Multi-projects Integration Control System

ZHENG Xian<sup>1</sup>, LU Yujie<sup>2</sup>, LI Yongkui<sup>1</sup>, LE Yun<sup>1</sup>

(1. School of Economics and Management, Tongji University, Shanghai 200092, China; 2. School of Design and Environment, National University of Singapore, Singapore 117566, Singapore)

**Abstract:** Aimed at implementing the rational modular partition and scientific evaluation, design structure matrix (DSM) theories and technologies were adopted to establish the evaluation and optimization methodology for city infrastructure multi-projects integration control system, encompassing the algorithm of independent intensity and number, propagation cost, clustered cost etc. In terms of the system update, the genetic algorithm was combined with the clustering algorithm of DSM so as to achieve the most optimal update scheme. Changchun infrastructure multi-projects integration control system was selected as the typical case, two versions of which before and after updating respectively were compared on degree of modularity and an update scheme

for next generation had been put forwarded. The results verify the feasibility and effectiveness of the proposed methodology on modular partition, schemes evaluation and optimal selection, which provide theoretical and practical references for the development and maintenance of large integrated control system of the same kind.

**Key words:** multi-projects integration control; modularity; design structure matrix

我国正处于快速城市化发展阶段, 大规模基础设施项目正处于建设和规划中, 管理这些项目亟需借助多项目集成信息平台以满足多主体的个性化需求<sup>[1]</sup>. 商业软件如 Oracle Primavera P6 等虽然是较好的项目管理软件, 但相对于城市基础设施项目群综合管控情境的复杂性和特殊性, 这些系统功能的完整性和适应性远远不够, 而且缺乏一些复杂项目群管理的功能, 如投资管理、前期管理、招投标管理等基础设施项目全过程管理功能等, 因此必须结合具体的管理情境进行个性化开发.

此类系统平台的开发是一个复杂工程, 具有功能多且相互关联、用户多样、需求不确定及跨组织工作流程和项目群信息集成复杂等特点. 系统需求的复杂性导致系统设计的复杂性, 同时这种复杂性增加了维护难度, 例如由于系统功能模块的高度关联性, 一个错误的产生将引起更大范围的系统错误, 这种连锁反应增加了维护时间和成本, 使系统应对变化的能力降低, 最终缩短系统的使用寿命. 因此系统设计是决定系统成功与否的关键性因素<sup>[1-3]</sup>, 良好的系统设计为利益相关者带来最大化的全生命周期价值, 包括了系统的设计开发价值和维护价值. 统计显示, 系统的维护成本占系统总投入的 40%~75%<sup>[4]</sup>. 针对以上问题, 提高系统的适应性是应对复杂环境风险、减少高昂维护成本<sup>[5]</sup>的最主要手段.

收稿日期: 2015-04-13

基金项目: 国家自然科学基金(71390523)

第一作者: 郑 弦(1989—), 女, 博士生, 主要研究方向为复杂工程项目管理. E-mail: sine2588@163.com

通讯作者: 乐 云(1964—), 男, 教授, 博士生导师, 工学博士, 主要研究方向为复杂工程项目组织. E-mail: yunle@kcpm.com.cn

模块化系统设计对促进系统适应性意义重大<sup>[6]</sup>,能促使系统有效应对各种变化和由此产生的各类风险.系统模块化的研究可以追溯至 1972 年<sup>[7]</sup>,但系统设计构架的研究还存在一些共性问题:仅分析系统的局部而不是作为整体进行研究;大多关注当前系统设计所产生的不良结果(如增加维护成本等),对系统设计构架本身的问题关注较少;系统设计的研究局限在某一阶段,缺乏对系统全生命周期价值影响的考虑;对系统设计构架的评价主要依赖于定性和手工分析.设计结构矩阵(DSM)技术能较好地解决以上系统开发中的问题.它是一种能定量反映信息关联和反馈的工具,通过充分考虑系统的整体性和关联性,进而实现系统全生命周期的优化,科学地提高系统的适应性.

DSM 算法中的聚类是实现系统模块化的有效方法,通过将 DSM 中紧密相连的行列元素归入同一类型的过程,使聚类内部的各要素之间的联系强度加强,而聚类之间的联系强度减弱,这可有效减少系统全生命期过程中变更风险和错误连锁反应.在系统优化方面,部分学者已将 DSM 方法与其他方法融合以解决复杂的优化问题,例如 Cho 和 Eppinger<sup>[8]</sup>采用 DSM 和高级仿真相结合预测项目完工时间;钱晓明等<sup>[9]</sup>利用 DSM 和遗传算法结合启发式调度规则研究了新产品并行开发过程的资源调度与优化算法;盛海涛和魏法杰<sup>[10]</sup>以遗传算法为工具,通过在优化目标函数中引入信息耦合强度等参数,有效解决了大规模设计结构矩阵的优化问题.

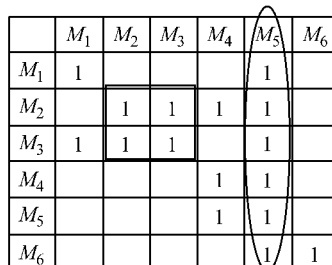
本文对信息系统结构的适应性及变更对系统结构设计的影响开展研究,采用基于任务的 DSM,即把系统的功能任务作为模型的行列元素,在功能之间的信息需求关系的基础上对过程模型进行建模和分析.虽然 DSM 技术在软件开发领域也已有相关研究,Sangal 等<sup>[11]</sup>采用 DSM 进行复杂软件构架研究;LaMantia 等<sup>[12]</sup>将设计规则理论与 DSM 结合解决系统的模块化划分问题,但目前对该技术的研究仍不够深入,缺乏结合实例对系统构架进行评价和优化的研究.鉴于此,本文将 DSM 和遗传算法结合后引入城市基础设施项目群集成控制系统的评价与优化问题,验证该方法在实践中的有效性和可行性.

## 1 DSM 在系统设计中的应用

由于 DSM 技术在过程建模、信息流规划以及优化等方面的作用,近年来被越来越多地用于航天、汽

车、建筑设计和制造业等各个领域<sup>[13-14]</sup>,具体研究内容包含设计过程管理、进度优化、流程重组和团队合作等<sup>[15-17]</sup>.在系统设计中,DSM 技术能描述某一特定设计域内各设计要素之间的相互依赖和约束等关系,能够为系统设计中的变更以及结构优化提供规划和分析的基础.尤其是在大型复杂系统设计中,加强了设计人员对系统设计过程信息流的理解,辅助系统规划设计、调度、运行和管理,提高设计效率.DSM 同样可以作为分析和管理工具,通过 DSM 对复杂系统内部各个功能模块之间的联系和依赖关系有一个直观的认识,帮助开发和管理人员进行系统设计和集成控制等工作.

具体而言,DSM 是由排列顺序相同的行列元素组成的方阵,布尔型 DSM 示意如图 1.  $M_1 \sim M_6$  等表示系统元素,在系统设计中,若  $M_2$  和  $M_3$  之间存在关联,表明  $M_2$  对应的元素需要  $M_3$  的输入,则矩阵中的第 2 行 3 列的值为 1;否则空格(或由数字 0 表示).在 DSM 矩阵中,对角线上的值无意义,对角线以下的“1”表示前信息反馈,对角线以上的“1”表示后信息前馈.若元素间既存在前馈也存在反馈,称之为耦合(例如  $M_2$  和  $M_3$ ),如果某一元素与大多数元素均相关,称此类元素为 bus(例如图 1 中  $M_5$ ).



	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$
$M_1$	1				1	
$M_2$		1	1	1	1	
$M_3$	1	1	1		1	
$M_4$				1	1	
$M_5$				1	1	
$M_6$					1	1

图 1 设计结构矩阵示意

Fig.1 Diagram of design structure matrix

DSM 技术应用于城市基础设施项目群集成控制系统这样基于流程和表单的大型综合管控系统的模块化划分有如下优势:①当被分析系统规模较大、依赖关系复杂众多时,方框和箭头图(如 UML 包和类图)难以直观分析,此时 DSM 显示较好的直观性;②使用 DSM 划分方法可以识别出模块结构并发现环依赖;③DSM 为系统构架设计提供了一种自动划分机制,可根据实际需要(或组织结构)对矩阵进行调整以得到满意的系统构架开发视图;④通过 DSM 进行系统模块划分后,被划分的模块可并行开发,缩短开发时间;⑤DSM 使系统构架呈现“高内聚、松耦合”的构架模式,当由于内外部环境发生变化时,可减少更新过程中错误连锁反应的产生.

## 2 基于 DSM 的系统设计模块化评价和优化方法

### 2.1 系统设计 DSM 模型构建

(1) DSM 中要素的选择. 由于本文研究的问题是系统的模块化程度, 因此采用系统的功能作为 DSM 分析要素.

(2) DSM 中依赖关系的选择. 城市基础设施项目群集成控制系统的各功能间依赖关系是信息的传递, 包括了数据、信息(狭义)和知识 3 种依赖介质, 例如用户输入的数据、图纸等, 因此统计信息和表单等均可作为输入和输出被其他功能所调用, 使功能之间产生依赖关系.

(3) DSM 中元素的依赖强度和数量. 建立了不同系统元素间依赖关系和信息流向的二元 DSM 矩阵, 它可对系统设计结构进行较好的初始分析. 为简化计算, 本文选择采用布尔型矩阵描述功能间的相互关系, 即依赖强度分为“0”和“1”. DSM 中非对角线的“1”数量之和为 DSM 依赖数量, 它反映系统元素联系的紧密程度, 紧密程度越高表示系统越复杂. 依赖密度是将依赖数量和功能数量联系起来的一个相对评价指标, 用来反映单位成对元素的依赖数量. 计算公式是: 依赖密度等于依赖数量除以矩阵的总元素对数( $N \times N$ ), 其中,  $N$  表示元素的总数量.

### 2.2 系统设计模块化评价

#### 2.2.1 传递成本

传递成本是用来揭示系统设计各功能之间的耦合关系, 进而反映单一功能变化对其他功能可能产生的直接或间接的影响, 例如: 从图 1 中功能间信息依赖关系可以看出,  $M_3$  直接依赖于  $M_1$  和  $M_2$ , 因此传递成本为 1; 而需要经过  $M_2$  后间接依赖于  $M_4$ , 因此传递成本为 2. 对传递成本的计算可借鉴可达矩阵的概念和算法<sup>[18]</sup>. 在采用设计结构矩阵转置得到邻接矩阵后可计算出可达矩阵. 传递成本的算法为: 可达矩阵中非空(或非“0”)单元格的总数量除以矩阵的总元素对数( $N \times N$ ). 该指标表示系统的耦合程度, 耦合程度越高表示系统设计的模块化程度越低, 适应性越差.

#### 2.2.2 聚类成本

由于存在于同一个聚类块中的元素传递成本较小, 在不同聚类中的元素传递成本较大, 因此聚类成本是采用差别成本的方式来反映系统设计模块化程度的优劣, 以有效补充“传递成本”单一指标评价系

统设计模块化程度的不足, 有利于系统设计方案的优化. 系统设计聚类成本的计算分为 3 步: ①对系统设计 DSM 进行聚类; ②找到大部分其他行列元素均有联系的 bus 类元素; ③根据元素所处的聚类块, 计算两两元素之间的依赖成本; ④最后将所有元素的依赖成本求和. 聚类成本算法如下:

(1) 对于元素  $M_i$  和  $M_j$ , 若  $M_j$  为 bus 类元素, 则依赖成本  $I = d_{ij}$ ,  $d_{ij}$  为依赖强度.

(2) 对于元素  $M_i$  和  $M_j$ , 若两者属于同一个聚类块, 则依赖成本  $I = d_{ij} x^\lambda$ ,  $x$  为该聚类块内元素的数量,  $\lambda$  根据计算需要取定, 一般取 2.

(3) 对于元素  $M_i$  和  $M_j$ , 若两者不属于同一个聚类块, 则依赖成本  $I = d_{ij} N^\lambda$ .

### 2.3 基于 GA 的系统设计模块化优化

城市基础设施项目群集成控制系统在使用过程面临极大的内外部环境变化, 如组织变革和技术更新等, 因此系统维护过程中系统更新十分常见. 系统更新是一个持续的过程, 来保证系统的有用性, 不良的更新管理导致软件的全生命周期中出现更多的错误, 因此系统更新直接决定着系统的使用寿命<sup>[19]</sup>, 研究表明大部分的系统淘汰是由于不良的系统更新导致的. 虽然前文提出的评价算法能评估不同模块化方案的优劣, 但是不能直接进行更新方案的优选. 遗传算法(Genetic Algorithm, GA)是搜索复杂问题最优解的有效方法, 通过选择、交叉、变异等操作将择优与随机信息交换相结合, 再进行一系列的迭代过程来得到问题的最优解. 遗传算法应用到 DSM 聚类划分有利于得出系统模块化最优更新方案.

#### 2.3.1 系统模块编码及种群初始化

在系统更新前先对组织的功能需求进行再调研, 了解系统需求的变化并据此得出更新后 DSM 模型. 对于独立元素需先进行识别和分离, 因为它们与其他元素没有关系或者仅有很少的关系, 从而可简化模型, 在最后非独立元素聚类完成后再将独立元素增加到行列元素的前方. 假设系统中非独立功能为数量为  $n$  ( $n \leq N$ ), 则 DSM 模型的行列元素按照顺序依次编码为  $M_1, M_2, M_3, \dots, M_n$ . 若系统开发团队允许的系统最大聚类数为  $k$ , 则对功能进行模块化聚类划分的二维编码矩阵为  $k \times n$  矩阵  $C$ , 如式(1)所示. 编码矩阵的每一行对应一个聚类, 则有  $k$  个聚类, 分别为  $C_1, C_2, \dots, C_k$ , 行列元素的顺序与聚类前 DSM 模型顺序保持一致, 编码矩阵中每一个单元格  $s_{ij}$  的值代表  $M_j$  是否在  $C_i$  中. 以聚类  $C_1$  为例, 假设其包含的功能要素为  $M_1, M_2$ , 映射到编码结果矩阵

$C$  的第 1 行中,行列要素  $M_1, M_2$  对应的单元格  $s_{11}$  和  $s_{12}$  的值为 1,其他单元格为 0.

$$C = \begin{Bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & & \vdots \\ s_{k1} & s_{k2} & \cdots & s_{kn} \end{Bmatrix} \quad (1)$$

$$s_{ij} = \begin{cases} 1 & M_j \in C_i \\ 0 & M_j \notin C_i \end{cases} \quad (2)$$

对种群进行初始化如下:①对于一条染色体  $C$  ( $k \times n$  矩阵),每 1 个列向量中仅任取 1 个行元素置其为 1,其余为 0;②反复执行步骤①使获得足够的初始染色体.

### 2.3.2 构建适应度函数

构建适应度函数的原则是:①聚类内部元素之间联系尽可能大,聚类之间联系尽可能小;②较小聚类中的联系比较大聚类中的联系更加容易<sup>[20]</sup>.

对于 DSM 模型,设在任意一代种群中,第  $y$  个染色体聚类数目为  $k$ ,则第  $i$  个聚类中第  $j$  个基因相对于其他不为零的基因对应的权重可用式(3)计算.

$$w_{ij} = (s_{ij} \times s_{ig}) \times \sum_{g=1}^n d_{jg} \quad (3)$$

式中: $s_{ij}, s_{ig}$  为染色体  $C$  中第  $i$  个聚类中第  $j, g$  个基因对应的数值; $d_{jg}$  为该染色体第  $i$  个聚类中第  $j$  个基因对应的元素与第  $g$  个基因对应的元素在更新的 DSM 中显示的依赖强度.

如果第  $i$  个聚类中有  $x_i$  个元素,那么聚类  $C_i$  为

$$C_i = \frac{1}{x_i} \sum_{j=1}^{x_i} w_{ij} \quad (4)$$

如果该染色体实际聚类数为  $k_x$ ,那么该染色体的整体聚类值为

$$C(y) = \sum_{i=1}^{k_x} C_i \quad (5)$$

把聚类划分的染色体整体聚类值作为目标函数,根据相对系数确定适应度函数的方法,该染色体的适应度函数为

$$f(y) = \frac{C(y) - C_{\min}}{C_{\max} - C_{\min}} \quad (6)$$

式中: $C_{\min}$  为种群中目标函数的最小值; $C_{\max}$  为目标函数的最大值.

种群的选择概率按照轮盘赌的方法计算,则第  $y$  个染色体被选择的概率为

$$p(y) = \frac{f(y)}{\sum_{x=1}^h f(h)} \quad (7)$$

式中: $f(y)$  为种群中第  $x$  条染色体的适应度值; $h$  为

染色体总数.

### 2.3.3 交叉与变异

采用单点交叉,在父染色体矩阵列中任意选择一列作为交叉点,交换交叉点右侧的列元素;变异方法采取任意选取父染色体中 2 列元素进行交换(行列元素顺序不变).

根据以上 3 个步骤即可得到最优的系统更新方案,保证系统开发人员能较独立开展系统更新,提供系统的适应性.

## 3 实证分析

### 3.1 长春市基础设施项目群集成控制系统介绍

自 2007 年以来,长春市针对政府投资项目采用了集中代建模式,成立了长春市政府投资建设项目管理中心(简称“建管中心”),承担基础设施等政府投资项目代建管理.随着长春市城市基础设施项目急剧增加,给建管中心带来了极大的管理挑战.为了提升基础设施项目群管控水平,开发长春市基础设施项目群集成控制系统成为必然选择.

依据建管中心定位,其基础设施项目群管理的职能和任务涉及前期、勘察设计、招投标、施工准备、施工和验收及移交阶段,因此涉及前期立项与手续管理、勘察设计管理、造价管理、招投标与合同管理、征地与拆迁管理、施工管理、验收与移交管理等职能,这些管理工作既有一定独立性也有很大相关性.在组织方面,既涉及建管中心各职能部门和项目部,也涉及外部政府监管部门、投资(融资)机构、监理单位、总包等,呈现出复杂的多主体多项目多阶段特征.以合同管理子系统为例,该业务内容与各职能部门、项目部、外部政府监管部门和融资公司均有关联,子模块与流程管理、招标管理、计量管理、造价管理和进度管理等子模块均有依赖性,通过模块化的架构设计有助于软件的功能集成、开发和维护管理.

### 3.2 基于 DSM 的系统模块化比较

由于建管中心内外部环境变化,长春市基础设施项目群集成控制系统从首次开发至今共经历了 3 个阶段,第 3 阶段的 V3.0 实际上为第 1 阶段 V1.0 的更新.与系统 V1.0 相比,系统 V3.0 在系统设计上进行了较好的优化,使系统功能、模块划分和流程设计更具合理性和适应性.因此,先采用基于 DSM 的系统设计模块化划分和评价方法来比较系统 V1.0 和 V3.0 的模块化程度.

### 3.2.1 数据收集和模型建立

通过搜集完备的长春市基础设施项目群集成控制系统 V1.0 和 V3.0 资料,并对该系统需求调研人员、系统开发人员和项目经理的多次访谈,初步建立了基于信息依赖关系的 V1.0 和 V3.0 的系统功能 DSM. 由于功能数量之多,本文仅选取了系统功能层级里的二级功能作为分析要素建立 DSM 模型. 系统 V1.0 包含 69 个二级功能,系统 V3.0 包含 47 个. 将各项基本数据录入 ProjectDSM,得到表示功能模块之间依赖关系的初始 DSM 模型. 再通过调研取得了 2 个系统功能的模块化聚类方式,如图 2 和图 3 所示. 可以看出,系统 V3.0 的功能 DSM 在聚类后,除去与 bus 类元素的依赖关系外,聚类块之间的依赖关系已非常少,但存在规模过大的聚类块,这是系统设计改进的一个重点. 系统 V1.0 的 DSM 在聚类后,虽然聚类块之间的依赖关系明显减少了,但聚类块数量过多,且同样存在规模较大的聚类块.

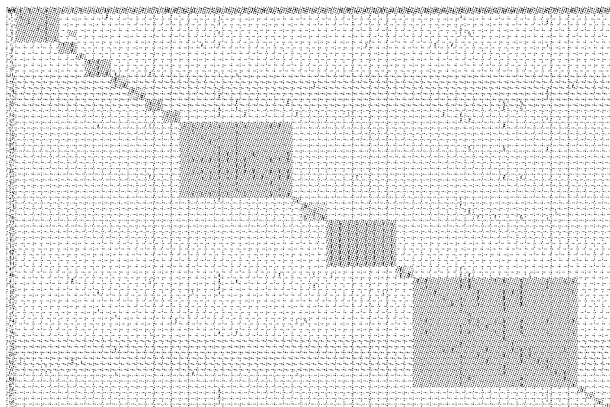


图 2 系统 V1.0 聚类后 DSM 模型

Fig.2 Clustered DSM model of V1.0 system

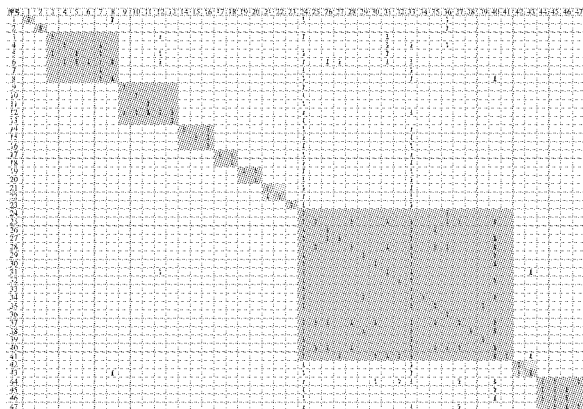


图 3 系统 V3.0 聚类后 DSM 模型

Fig.3 Clustered DSM model of V3.0 system

通过对 2 个系统版本的 DSM 建模和聚类可以计算出相应的依赖关系数量、依赖关系密度、传递成

本和聚类成本等. 此外,还收集了 2 个系统版本的源文件数量、系统代码行数和部分性能指标(如:网页平均响应时间、错误个数等). 具体比较如表 1.

表 1 2 个系统版本的评价指标项对比

Tab.1 Comparison of two system versions on evaluation indexes

对比项	集成控制系统 V1.0	集成控制系统 V3.0	相对变化/%
源文件数量	1 556	1 442	7.33
代码行数	17 235	15 942	7.50
二级功能数量	69	47	31.88
功能总数量	89	106	-19.10
功能的平均源文件数量	17.48	13.603	22.18
功能的平均代码行数	193.65	150.40	22.33
依赖关系数量	371	214	42.32
依赖关系密度	0.078	0.097	-24.36
传递成本	31.78%	28.02%	11.83
聚类成本	432 688	68 488	84.17
网页平均响应时间	210 ms	140 ms	33.33
测试期平均每月错误个数	11	6	45.45

注:相对变化是 V1.0 数据基于 V3.0 的变化率.

### 3.2.2 结果分析

从表 1 可见:系统 V3.0 的功能的平均源文件数量、功能的平均代码行数均低于 V1.0,而网页平均响应时间、错误个数也明显少于系统 V1.0,说明系统 V1.0 存在冗余或无用的系统源文件和代码,从而降低系统效率;而 V3.0 通过优化系统设计,减少了源文件数量和代码行数,提升了系统性能,该对比再次证实系统设计对系统性能的重要性,良好的系统设计能大大减少系统维护成本,提高用户满意度.

对于系统 V1.0 和系统 V3.0 的依赖关系数量和依赖关系密度,系统 V3.0 中的功能依赖关系减少了近 40%,这说明更新后系统 V3.0 进行了较好的模块划分,使功能之间的联系紧密程度降低了,从而降低了系统的复杂程度,增强系统适应性. 但对比依赖关系密度,系统 V3.0 高于系统 V1.0,主要原因是系统 V3.0 中 bus 类元素增加了,若去掉 bus 类元素与其他元素的关联,系统 V3.0 的依赖关系密度会大幅降低,小于 V1.0 密度.

功能系统 V1.0 比系统 V3.0 的依赖关系复杂,但通过计算发现 2 个系统版本的传递成本相差仅 3%,这说明系统 V3.0 功能之间的间接依赖关系较多,只有发现这种隐藏依赖关系,才能更好开展模块化系统设计. 而对于聚类成本这一指标,系统 V1.0 比系统 V3.0 多出 6 倍,说明系统 V1.0 虽然采用了模块化设计方法,但是模块间的关联性依然过多,功能组合需要优化,以减少功能依赖性,提高系统适应

性.改进后的系统 V3.0 对功能的开发顺序和组合进行了更佳的设计,通过调整模块划分方式,减少了维护成本,增大系统的全寿命周期价值.通过以上数据对比,充分说明了系统 V3.0 的模块化程度显著高于系统 V1.0,验证了 DSM 技术在系统模块化划分和评价上的有效性.

### 3.3 DSM 和 GA 结合的系统更新方案优选

在 V3.0 系统使用 1 年后,建管中心为了精细化、规范化和高效化基础设施管理,在管理模式、部门与职能、人员岗位与数量等方面均进行了一系列的变革.例如,在管理模式上,建管中心成立了项目管理公司;在部门与职能方面,部分部门进行了职责调整,包括将征地与拆排迁部、前期计划部的部分职能并入了总工办,征地与拆迁部和前期计划部合并为征地拆迁部(前期部)等.在这样的组织改革背景下,再次需要对系统进行更新.本文拟针对系统从 V3.0 到 V3.1 的更新,将 DSM 和 GA 技术结合得出最优的更新方案.

首先进行需求调研,更新后的二级功能子模块数量为 59,再对各个部门的部分员工进行半结构的访谈后得出子模块间的依赖强度,从而得到 V3.1 系统的布尔型 DSM 模型.假设系统开发团队允许的最大聚类数为 8,则染色体编码为  $8 \times 59$  矩阵.将种群大小设定为 200,交叉概率为 0.9,变异概率为 0.1,终止代数数为 200 代,在 MATLAB 软件中执行选择、交叉和变异算子后得出收敛过程曲线如图 4 所示,曲线表明执行到 155 代时收敛,收敛时最优目标函数值为 20.36.将最优染色体解码至 DSM 模型中,得到聚类后的 DSM 模型如图 5 所示.以第 1 个聚类块为例,二级功能模块 33(工程计量管理子模块)、35(工程洽商管理子模块)、39(形象进度管理子模块)、44(工程计量汇总子模块)、45(工程洽商汇总子模块)、49(计划监控子模块),它们并非属于同一个业务部门的职能,但是由于二级功能模块间相互的关联性,此 6 个子模块组合成一个更新模块,由一个开发团队负责更新.如此使聚类块内部联系得到了增强,依赖强度较弱的元素已从聚类块中分离出来,聚类块之间的联系大大减小,系统开发人员可以据此进行更新的分工,避免模块之间的强依赖性和错误的连带反应,提高了系统的适应性.

通过将 DSM 和 GA 结合进行更新方案的优选后,各二级功能模块在系统更新时的组合方式更加均衡和合理,以合同管理功能模块(图 5 中编号 23)为例,在 V3.0 系统版本中,由于该子模块与流程管

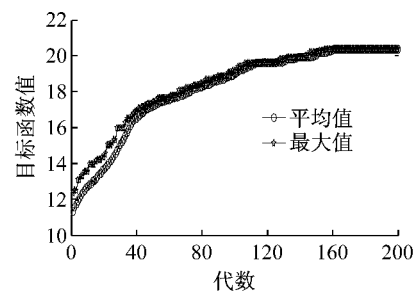


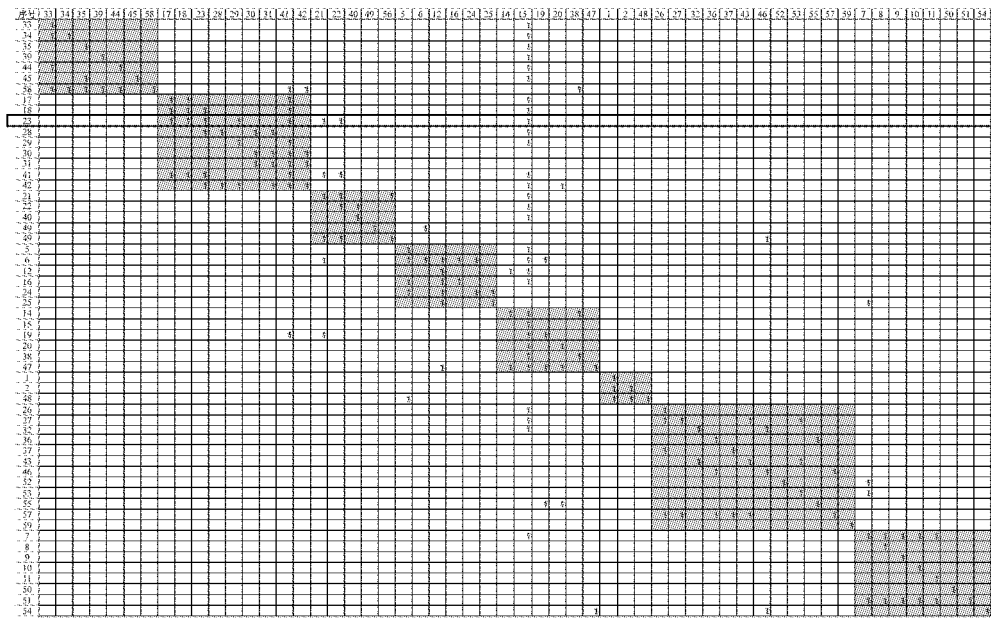
图 4 收敛曲线

Fig.4 Convergence curves

理、招标管理、计量管理、造价管理和进度管理等子模块均有依赖性,因此聚类形成了一个包含 18 个二级功能模块的大功能集合,包括排迁任务管理、拦标价管理、排迁内业信息、履约保证金、标段划分管理、结算管理等(见图 3),这类大功能块并不利于系统更新,因此本次更新的一个重点是对该大聚类块的解耦.此外,V3.1 系统进行更新的主要原因是功能模块的内容进行了增加或整合,例如合同管理子模块中增加了合同金额的自动统计功能,因此功能模块之间的依赖关系有了较大变动,需要提出最合理的更新方案.经过采用 GA 算法进行模拟和优化后,V3.1 系统的更新方案中大功能集合模块被重新划分,合同管理功能模块仅与其他 8 个二级功能模块一起被更新,包括征收资金管理(编号 28)、履约保证金(编号 29)、提请款管理(编号 30)以及其他若干与合同资金相关的功能,这极大地提高了更新效率.此外,更新后的系统在数据的调用方面更加灵活和快捷,这提高系统读取数据的效率和计算速度.

## 4 结语

城市基础设施项目群集成控制系统是一类复杂适应性系统,需应对系统功能、用户和需求的不确定性和管理环境的复杂性,提高模块化程度能增强系统适应性.本文提出了基于 DSM 的系统设计模块化划分、评价和更新优化方法,通过建立系统功能之间信息依赖关系的 DSM,采用 DSM 聚类方法实现了系统设计的模块化划分,根据聚类后的 DSM 计算依赖关系数量、依赖关系密度、传递成本和聚类成本等,并结合系统性能指标,可以对系统模块化程度进行比较和评价;此外,通过将 GA 算法与 DSM 聚类算法相结合可得出最优的更新方案,提高更新效率,延长使用寿命.本文以长春市基础设施项目群集成控制系统为例,通过比较更新前后 2 个版本系统设计的模块化程度并结合组织变更需求提出了下一代



注:方框内为“合同管理”二级功能模块,编号为 23.

图 5 最优更新方案的 DSM 模型

Fig.5 DSM mode of the optimal updated scheme

系统的最优变更方案,验证了该方法在提高系统适应性和全生命周期价值上的有效性,该方法有助于同类集成控制系统在开发和维护过程中进行系统模块化划分、方案评价和优选.

#### 参考文献:

- [1] Lee S K, Yu J H. Success model of project management information system in construction [J]. Automation in Construction, 2012, 25: 82.
- [2] Engel A, T R Browning. Designing systems for adaptability by means of architecture options [J]. Systems Engineering, 2008, 11(2): 125.
- [3] McCormack A, Rusnak J, Baldwin C Y. Exploring the structure of complex software designs: an empirical study of open source and proprietary code [J]. Management Science, 2006, 52(7): 1015.
- [4] Sosa M E, Browning T, Mihm J. Studying the dynamics of the architecture of software products [C] // ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Las Vegas; American Society of Mechanical Engineers, 2007: 329-342.
- [5] Tan Y, V S Mookerjee. Comparing uniform and flexible policies for software maintenance and replacement [J]. IEEE Transactions on Software Engineering, 2005, 31(3): 238.
- [6] Davis S A Slaughter. Software development practices, software complexity, and software maintenance performance: A field study [J]. Management Science, 1998, 44(4): 433.
- [7] McCormack A D. Product-development practices that work: How internet companies build software [J]. Sloan Management Review, 2001, 42(2): 75.
- [8] Cho S, Eppinger S D. A simulation-based process model for managing complex design projects [J]. IEEE Transactions on Engineering Management, 2005, 52(3): 316.
- [9] 钱晓明, 王宁生, 唐敦兵. 基于 DSM 的产品开发过程资源优化算法[J]. 系统工程学报, 2008, 23(2): 238.  
QIAN Xiaoming, WANG Ningsheng, TANG Dunbing. Algorithms for resource optimization of PDP based on DSM [J]. Journal of Systems Engineering, 2008, 23(2): 238.
- [10] 盛海涛, 魏法杰. 基于遗传算法的设计结构矩阵优化方法研究 [J]. 中国管理科学, 2007, 15(7): 7.  
SHENG Haitao, WEI Fajie. The research and application of genetic-based design structure matrix optimization algorithm [J]. Chinese Journal of Management Science, 2007, 15(7): 7.
- [11] Sangal N, Jordan E, Sinha V, et al. Using dependency models to manage complex software architecture [J]. In ACM Sigplan Notices, 2005, 40(10): 167.
- [12] LaMantia M J, Cai Y, MacCormack A D, et al. Analyzing the evolution of large-scale software systems using design structure matrices and design rule theory: Two exploratory cases [C] // Seventh Working IEEE/IFIP Conference on Software Architecture. Vancouver; IEEE, 2008: 83-92.
- [13] Steward D V. The design structure system—A method for managing the design of complex systems [J]. IEEE Transactions on Engineering Management, 1981, 28(3): 71.
- [14] Eppinger S D, Whitney D E, Smith R P, et al. A model-based method for organizing tasks in product development [J]. Research in Engineering Design, 1994, 6(1): 1.
- [15] Smith R P, Eppinger S D. Identifying controlling features of engineering design iteration [J]. Management Science, 1997, 43(3): 276.
- [16] Austin S, Newton A, Steele J, et al. Modelling and managing project complexity [J]. International Journal of Project Management, 2002, 20(3): 191.

(下转第 1922 页)