

网络结构的软件系统可靠性分配方法

杨宏伟,王焕坤,李庆全

(装甲兵工程学院 技术保障工程系,北京 100072)

摘要: 针对网络结构的软件系统,通过构造软件系统的 ANP 结构,结合用户和程序员的观点,对超矩阵的构造进行了改进,提出基于改进型 ANP 的模块重要度评估方法。以软件实用性最大化为目标函数、以可靠性下限和开发费用为约束条件,建立了可靠性分配模型。并以某工厂实力信息管理系统为例,对算法进行了验证。

关键词: 软件网络;可靠性分配;ANP;超矩阵

中图分类号: TP301.6

文献标识码: A

文章编号: 1674-7720(2012)19-0008-04

The model of software system reliability allocation with network structure

Yang Hongwei, Wang Huankun, Li Qingquan

(Department of Technical Support Engineering, Academy of Armored Force Engineering, Beijing 100072, China)

Abstract: To the software system with networked structure, this paper proposes the software dependability distribution method based on ameliorated ANP. In this method, to assess importance degree of the module in the software network through constructing ANP structure of the software system and combining the views of the user and programmer, then to get the goal that maximizing the user's availability. At the same time, considering the constraints of reliability and cost. Finally, take an information management system as an example, to prove this method further.

Key words: software network; reliability allocation; ANP; super-matrix

软件结构描述了软件实体元素方法、属性、类、包以及它们之间的相互关系,在很大程度上直接影响着软件的功能、性能、安全性及可靠性等指标。因此,在进行软件可靠性分配前,对软件系统结构进行分析是非常有必要的。

目前,在软件可靠性分配的研究中往往是基于软件系统中模块间相互独立、没有交互的假设上。然而现实情况并非如此,由于要处理繁杂的事务,现在的软件系统通常都很复杂,由许多互相作用的模块构成。近年来,一些研究者对大量面向对象软件系统的类图进行了研究,发现软件系统的结构大多数都展现出“小世界”和“无尺度”等复杂网络特性^[1]。对于这类软件的可靠性分配问题,可以用网络分析法(ANP)进行探讨。

1 软件网络观

1.1 软件的复杂性

软件系统复杂性的完整定义应当包括软件的内部结构和外部特性两个方面,其中内部结构反映软件的静

态复杂性,而外部特性则反映软件的动态复杂性。本文主要讨论软件内部结构的复杂性。

复杂系统不一定要有很大的规模,因为引起系统复杂行为的主要原因并不是元件的数量,而是元件之间的交互。所以元件之间的交互方式对复杂系统的特性及行为的影响是一个不可忽视的方面,也是复杂系统研究的关键问题之一。软件系统也是如此,单个类或模块只能完成有限的功能,而在系统范围内,所有类或模块协同交互才能完成用户期望的功能。因此,软件系统模块间的关系和交互的复杂性也是判断软件结构复杂性的重要因素^[1]。由此可知复杂软件系统的结构不再是简单的层次结构,模块之间的复杂关系使软件结构呈现网络化。

1.2 软件的网络结构

对于软件网络的抽象方式,研究人员从不同的层面均给予了不同的定义,例如节点可以是类、子程序、构件和子系统等^[2],边可以是面向对象设计中的类间关系或包间关系等^[3],也可以是面向过程设计中的函数间调用

或子程序间调用等^[4]。

2002年, Valverde^[2]等人首先研究了软件网络。他们通过逆向工程方法从程序代码得到系统的类图,然后将系统的类图作为研究对象,用网络图来表示软件系统,即网络中的节点表示类,边代表类之间的交互关系。这样,抽象的软件网络可以用一个二元组来定义:

$$\Omega_s = (W_s, E_s)$$

式中, $W_s = \{s_i | i=1, 2, \dots, N\}$, 表示由 N 个类 s_i 组成的集合; $E_s = \{e_i | i=1, 2, \dots, N\}$ 是所有类之间关系 e_i 的集合。

2 基于功能结构的可靠性分配思想

这里将软件系统理解为由多个模块组成,通过模块间相互协调、相互作用来实现系统的多种功能。软件系统的可靠性分配就是以系统实用性为目标,在可靠性指标下限和开发费用为约束条件下,在设计阶段明确各个模块的可靠性指标值,使系统在约束条件下实现实用性最大化^[5]。

2.1 确定目标函数

对用户来说,系统的可靠性最直接的体现是在系统的实用性,且用户对系统可用性的评价是基于对系统功能可靠性的评价。因此,这里以系统的实用性最大化为目标。

设软件可用性与软件模块以及这些模块的可靠性为线性关系,可以定义可用性为:

$$U = \sum_{i=1}^m w_{m_i} \cdot r_{m_i} \quad (1)$$

其中, w_{m_i} 为模块 i 的全局重要度, r_{m_i} 为模块 i 的可靠性。

目标函数:

$$\max U = \sum_{i=1}^m w_{m_i} \cdot r_{m_i} \quad (2)$$

由式(2)可以看出,模块 i 的重要度越大,可靠性越高,则系统的实用性也越高。

2.2 约束条件

(1) 可靠性下限约束

设 d_i 为模块 i 的可靠性下限值,称为“最低可接受的”可靠性水平^[6],它可以由软件工程师在软件产品的计划和设计阶段确定,则:

$$d_i \leq r_{m_i} \leq 1 \quad (3)$$

(2) 费用约束

一个模块的售价 v_i 是这样决定的:设 V 表示用户愿意为购买软件产品而付的价钱(即产品的市场价格),在用户对软件产品的评价过程中, w_{m_i} 为模块 i 的重要度,于是模块 i 的价格为^[7]:

$$v_i = w_{m_i} \cdot V \quad (4)$$

假定可靠性与费用呈线性关系^[8], α_i 表示对模块 i 施加可靠性为 r_{m_i} 时的一般开销, β_i 为可调整的成本开销, a 等于 1 减去软件开发者的利润率, v_i 是模块 i 的售价(为模块 i 的设计完成成本),模块 i 的实际成本一定不能超

过它,即:

$$\alpha_i + \beta_i \cdot r_{m_i} < a \cdot v_i \quad (5)$$

若 C 为有限的开发费用,则各模块可靠性成本的总和必须小于 C ,即:

$$\sum_{i=1}^m (\alpha_i + \beta_i \cdot r_{m_i}) < C \quad (6)$$

2.3 建立可靠性分配模型

综合式(2)、(3)、(5)、(6),可建立软件可靠性分配模型如下:

$$\begin{aligned} \max U &= \sum_{i=1}^m w_{m_i} \cdot r_{m_i} \\ &\begin{cases} d_i \leq r_{m_i} \\ r_{m_i} \leq 1 \\ \alpha_i + \beta_i \cdot r_{m_i} < a \cdot v_i \\ \sum_{i=1}^m (\alpha_i + \beta_i \cdot r_{m_i}) < C \end{cases} \end{aligned}$$

可靠性分配就是通过模型求解出 r_{m_i} 的值。对于模型中的参数,本文重点探讨如何确定模块的重要度 w_{m_i} 。

ZAHEDI E 和 ASHRAFI N 采用层次分析法 AHP (Analytic Hierarchy Process) 建立软件系统的层次模型,而后进行模块重要度评估^[9]。这里由于模块间的相互作用关系,系统呈现网络化的组织结构,已不能再用 AHP 方法,下面主要通过网络分析法 (ANP) 对模块重要度进行评估。

3 基于 ANP 的模块重要度评估

3.1 ANP 理论

网络分析法 ANP (Analytic Network Process) 的决策过程可以分为两大类:一是在控制层的决策,由于控制层元素相互独立,所以其决策过程与 AHP 方法相同;二是在网络层的决策,由于网络层元素间存在相互影响,其决策过程中引入了超矩阵的应用和分析。

基于 ANP 求解问题的基本流程如图 1 所示。

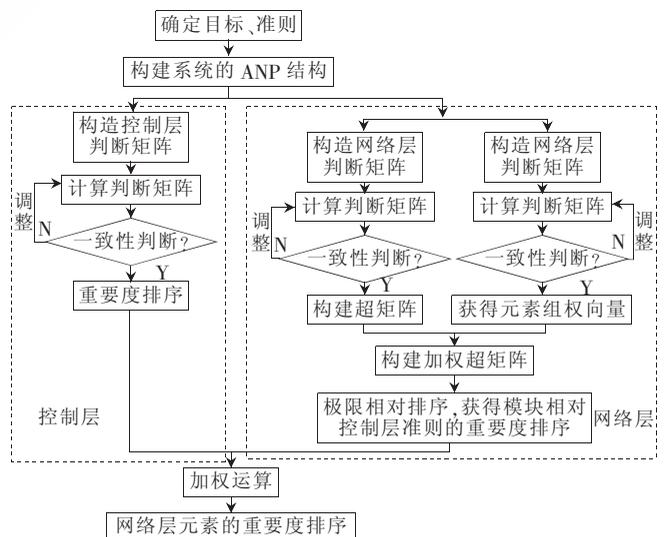


图 1 ANP 的决策流程

3.2 软件系统的 ANP 结构^[10]

作为系统功能需求的提出者,用户最明白系统需要具备哪些功能;作为系统的设计者和开发者(程序员)最熟悉系统的整体结构,从而也最了解该由哪些模块组成这些功能。因此,这里以软件用户的观点,划分 s 项功能为 f_1, f_2, \dots, f_s ; 然后以软件程序员的观点进一步将功能划分为 n 个的模块组 C_1, C_2, \dots, C_n , 共 m 个模块, 每个功能可能调用多于一个模块。

(1) 控制层

控制层的目标层是基于用户观点的, 是用户对软件可靠性与可用性的总体评估。总体评估是基于软件属性或功能, 因此, 控制层的决策准则为软件系统能够提供给用户的功能。

(2) 网络层

每个功能又是通过不同的模块组合实现的。模块之间不是简单的相互独立关系, 而是存在反馈依存关系, 形成网络结构。

软件系统的 ANP 结构如图 2 所示。

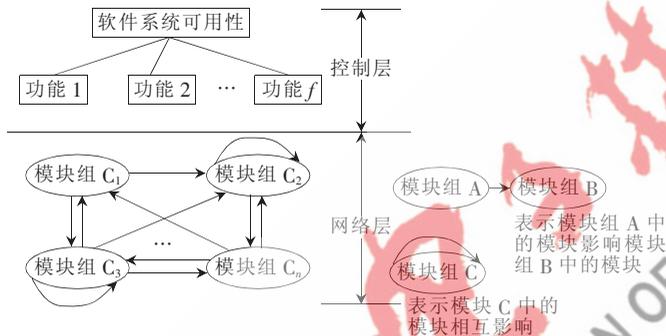


图2 软件系统的 ANP 结构

3.3 基于改进型 ANP 的模块重要度评估

结合软件系统的 ANP 网络模型, 在控制层主要由用户对各项功能对系统的重要程度作比较评估; 在网络层主要由程序员对各个模块对各项功能的重要程度作比较评估。

3.3.1 功能重要度指标 WF

软件用户通过比较两项功能重要度, 按照“1-9”标度规则, 构造判断矩阵 AF 。计算矩阵 AF 的最大特征值 λ_{\max} 对应的特征向量 WF , 计算一致性指标:

$$C.I. = \frac{\lambda_{\max} - n}{n - 1}$$

一般情况下当 $C.I. \leq 0.1$ 时, 认为判断矩阵可以接受, 否则需要调整判断矩阵。

3.3.2 模块局部重要度指标 WM^f

传统 ANP 方法在构造判断矩阵时都有两个评估准则, 在元素两两比较时需要在某目标准则条件下, 先比较两元素 i, j 相对于该准则下的第三个元素 (称为次准则) 的重要程度 w_i 与 w_j , 然后再通过 w_i/w_j 得到元素 i 与元素 j 重要度比值^[10]。这种通过两步比较的方法不是很直观, 容易混乱决策者的思维, 产生错误。相对来说一步直接比较是比较直观且较容易被理解。

下面主要介绍一种利用一步直接比较来构造网络层的超矩阵。

(1) 构造模块间判断矩阵。这里将判断矩阵分为两种: 一是以控制层功能 f_i 为单一准则的判断矩阵; 二是以网络层模块组 C_j 中元素为单一准则的判断矩阵, 并计算出排序向量矩阵。

(2) 构造超矩阵。将所有网络层元素的排序向量矩阵组合, 由此得到控制层功能 f_i 下的超矩阵, 记为 W 。

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1N} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{2N} \\ w_{31} & w_{32} & w_{33} & \cdots & w_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{N1} & w_{N2} & w_{N3} & \cdots & w_{NN} \end{bmatrix}$$

(3) 构造模块组权矩阵。同样, 这里将元素组间判断矩阵分为两种: 一是以控制层功能 f_i 为单一准则的判断矩阵; 二是以网络层模块组 C_j 为单一准则的判断矩阵, 对应得到两个矩阵 A_f 和 A_c , 此时元素组权矩阵可表示为:

$$A = [A_f \quad A_c]$$

最后将模块组权矩阵 A 与超矩阵 W 相乘, 得到加权超矩阵 \bar{W} 。

对比分析, 传统 ANP 要得到 \bar{W} 共需构造判断矩阵 $(mn+n)s$ 个; 改进型 ANP 要得到 \bar{W} 共需构造判断矩阵 $(mn+sn+n+s)$ 个。

(4) 超矩阵运算。参考文献[11]运用矩阵分析的方法, 根据可约性和最大特征根的重数对超矩阵进行分类, 得到了 6 种类型的超矩阵和相应的 4 种计算方法, 这里不再赘述。

由于超矩阵的计算过程复杂, 一般要使用辅助工具计算, 因此, 本文使用 ANP 决策软件 Super Decision 进行计算, 以获得模块局部重要度指标 WM^f 。

3.3.3 求解模块全局重要度指标 WM

由 $WM = (WM^1, WM^2, \dots, WM^s) \cdot WF$ 可得模块全局重要度指标 WM 。

4 案例

本文以某一工厂实力信息管理系统为例, 对上述方法进一步说明和验证。该软件系统功能结构如图 3 所示。

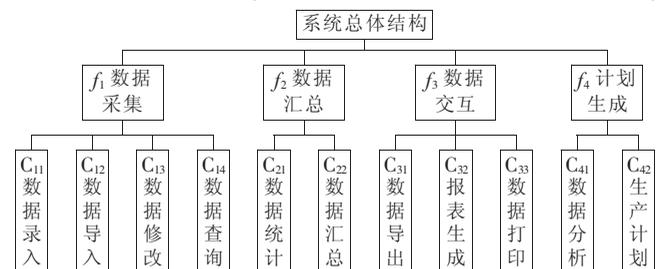


图3 工厂实力信息管理系统功能结构

系统要求各模块的可靠度不低于 0.9, 软件开发的投资为 495 000 元, 利润率为 50%, 则有: $a=0.5, C=495\ 000$ 。

其中, C_{13} 数据修改调用 C_{11} 数据录入; C_{21} 数据统计

《微型机与应用》2012 年第 31 卷第 19 期

调用 C₁₄ 数据查询; C₂₂ 数据汇总调用 C₂₁ 数据统计和 C₁₂ 数据导入; C₃₂ 报表生成调用 C₂₁ 数据统计和 C₂₂ 数据汇总; C₃₃ 数据打印调用 C₂₁ 数据统计、C₂₂ 数据汇总以及 C₃₂ 报表生成; C₄₂ 计划生成调用 C₂₁ 数据统计、C₂₂ 数据汇总和 C₄₁ 数据分析。由此, 可得该软件系统基于 ANP 的网络结构如图 4 所示。

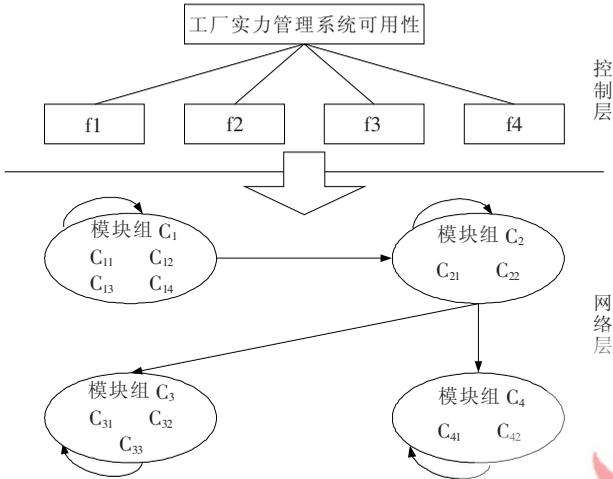


图 4 工厂实力信息系统基于 ANP 的网络结构

4.1 模块重要度计算

(1) 采用问卷调查的形式收集用户的评估意见, 根据第 3.3.1 节, 得:

$$WF = (0.335\ 4, 0.304\ 9, 0.246\ 3, 0.113\ 4)^T$$

(2) 同样, 也采用问卷调查的形式收集程序员评估意见, 而后分别用传统的 ANP 方法和改进型 ANP 方法进行超矩阵的构造, 最后用 Super Decision 软件进行辅助决策, 得到结果如表 1 所示。

表 1 模块重要度评估表

| | 传统 ANP | 改进型 ANP |
|-----------------|---------|---------|
| C ₁₁ | 0.079 3 | 0.077 3 |
| C ₁₂ | 0.137 3 | 0.126 3 |
| C ₁₃ | 0.026 6 | 0.028 6 |
| C ₁₄ | 0.201 7 | 0.202 7 |
| C ₂₁ | 0.178 1 | 0.177 1 |
| C ₂₂ | 0.139 9 | 0.150 9 |
| C ₃₁ | 0.038 1 | 0.039 3 |
| C ₃₂ | 0.047 9 | 0.046 9 |
| C ₃₃ | 0.028 1 | 0.031 1 |
| C ₄₁ | 0.078 9 | 0.078 7 |
| C ₄₂ | 0.043 9 | 0.040 9 |
| 构造判断矩阵个数 | 200 | 68 |

从表中的结果可知, 传统 ANP 与改进型 ANP 所得到的评估结果基本一致, 但是在超矩阵的构造过程中改进型 ANP 的计算量明显要低于传统的 ANP。

4.2 模型求解

设软件产品的售价为 1 000 000 元, 应用式 (4) 可计算出 (以下计算均取单位 1 000 元):

$$(v_1\ v_2\ v_3\ v_4\ v_5\ v_6\ v_7\ v_8\ v_9\ v_{10}\ v_{11}) = (38.56\ 63.15\ 14.30\ 101.35\ 88.55\ 75.54\ 19.65\ 23.45\ 15.55\ 39.35\ 20.45)$$

《微型机与应用》2012 年第 31 卷第 19 期

本文取:

$$(\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6\ \alpha_7\ \alpha_8\ \alpha_9\ \alpha_{10}\ \alpha_{11}) =$$

$$(6\ 20\ 3\ 14\ 37\ 55\ 25\ 27\ 22\ 10\ 18)$$

$$(\beta_1\ \beta_2\ \beta_3\ \beta_4\ \beta_5\ \beta_6\ \beta_7\ \beta_8\ \beta_9\ \beta_{10}\ \beta_{11}) =$$

$$(19\ 23\ 14\ 27\ 26\ 25\ 17\ 15\ 13\ 26\ 14)$$

将上述求得的参数代入可靠性分配模型中, 得:

$$\max U = 0.077\ 3rm_1 + 0.126\ 3rm_2 + 0.028\ 6rm_3 + 0.202\ 7rm_4 + 0.177\ 1rm_5 + 0.150\ 9rm_6 + 0.039\ 3rm_7 + 0.046\ 9rm_8 + 0.031\ 1rm_9 + 0.078\ 7rm_{10} + 0.040\ 9rm_{11}$$

$$0.9 \leq rm_i \leq 1 \quad (i=1, 2, \dots, 11)$$

$$20 + 19rm_1 \leq 38.56$$

$$41 + 23rm_2 \leq 63.15$$

$$1 + 14rm_3 \leq 14.30$$

$$75 + 27rm_4 \leq 101.35$$

$$63 + 26rm_5 \leq 88.55$$

$$51 + 25rm_6 \leq 75.54$$

$$s.t. \quad 3 + 17rm_7 \leq 19.65$$

$$9 + 15rm_8 \leq 23.45$$

$$3 + 13rm_9 \leq 15.55$$

$$14 + 26rm_{10} \leq 39.35$$

$$7 + 14rm_{11} \leq 20.45$$

$$287 + 19rm_1 + 23rm_2 + 14rm_3 + 27rm_4 + 26rm_5 +$$

$$25rm_6 + 17rm_7 + 15rm_8 + 13rm_9 + 26rm_{10} + 14rm_{11} \leq 495$$

通过求解上述模型, 可以得出此软件系统的可靠性分配的目标值为:

$$RM = (0.982\ 0.963\ 0.900\ 0.976\ 0.984\ 0.978$$

$$0.900\ 0.963\ 0.900\ 0.931\ 0.900)$$

$U = 0.961\ 3$, 即为约束条件下可得到的最大软件实用性。从结果中可知, 可靠性指标较高的模块为: C₁₁、C₁₂、C₁₄、C₂₁、C₂₂、C₃₂。因此, 在软件系统的开发过程需要投入较大人力物力以保证其完成的质量, 才能最终满足用户对软件实用性的要求。

目前软件可靠性分配研究中, 对于具有网络结构的软件可靠性分配讨论比较少。本文首先对软件的网络结构的存在性、普遍性进行了说明; 然后结合 ANP 理论讨论了具有网络结构的软件可靠性分配方法, 并对传统的 ANP 方法进行了改进; 最后将该方法应用于案例中进行验证, 得出如下结论: (1) 在 ANP 结构中, 没有依赖与影响关系的限制, 能更准确地描述软件系统网络结构; (2) 在 ANP 决策中, 综合了用户和程序员的观点, 降低了决策者单方面的主观因素的影响; (3) 通过改进 ANP 中超矩阵的构造, 方法更加清晰、直观、容易理解, 且在一定程度上减少了计算量; (4) ANP 方法能够有效地解决网络结构的软件系统可靠性分配问题。

参考文献

[1] 何克清, 马于涛, 刘娟, 等. 软件网络[M]. 北京: 科学出版

- 社, 2008.
- [2] VALVERDE S, CANEHO R, SOLE R. Seale free networks from optimal design [J]. EuroPhysics Letters, 2002, 60(4): 512-517.
- [3] SOLE R V, VALVERDE S. Information theory of complex networks: on evolution architectural constraints [J]. Lect Notes Physical, 2004, 65(1):189-207.
- [4] MYERS C R. Software systems as complex networks: Structure, function, and resolvability of software collaboration graphs[J]. Physical Review E, 2003, 68(4): 046-116.
- [5] 徐仁佐. 软件可靠性工程[M]. 北京:清华大学出版社, 2007.
- [6] 左云霞. 基于遗传算法与 AHP 的软件可靠性分配方法 [D]. 西安:西北工业大学, 2007: 13-18.
- [7] 徐仁佐, 向剑文, 肖英柏. 面向多用户软件系统的可靠性分配的故障树分析法 [J]. 小型微型计算机系统, 2001, 22(3): 329-332.
- [8] MARY E H, MING Z, OHLSSON N. Planning models for software reliability and cost [J]. IEEE Transaction on Software Engineering, 1998, 24(6):420-434.
- [9] ZAHEDI F. ASHRAFI N. Software reliability allocation based on structure, utility, price and cost [J]. IEEE Transaction Software Engineering, 1991, 17(4):345-356.
- [10] SAATY T L, VARGAS L G. Decision making with the analytic network Process [M]. Springer Science Business Media, LLC, 2006.
- [11] 吴志彬, 陈义华. ANP 中超矩阵排序算法研究 [D]. 中国控制与决策学术年会论文集, 2006: 1235-1242.

(收稿日期: 2012-05-09)

作者简介:

杨宏伟, 男, 1960 年生, 教授, 博士, 主要研究方向: 复杂管理系统建模与仿真。

王焕坤, 男, 1987 年生, 硕士, 主要研究方向: 系统运行与优化。

李庆全, 男, 1969 年生, 博士, 副教授, 主要研究方向: 物流管理。