Software Engineering and Applications 软件工程与应用, 2017, 6(3), 49-57
Published Online June 2017 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2017.63006

Hans汉斯

The Prediction of Software-Stage Effort Based on Improved Metabolic GM (1,1) Model

Yong Wang, Peipei Han

Ocean University of China, Qingdao Shandong
Email: markwy@126.com

Received: May 10th, 2017; accepted: May 29th, 2017; published: Jun. 1st, 2017

Abstract

At present, the researches of software effort prediction mainly focus on the prediction of total effort, and the prediction of software project stage effort is less, but the software industry has a strong demand for it. So, this paper studies software-stage effort prediction by using the GM (1,1) model of grey theories, and improves the metabolic model of GM (1,1), selects the initialization dynamically, and proposes a prediction method IGM. Experiments on three different datasets demonstrate that IGM method is superior to traditional metabolic GM (1,1) model, GV method and LR model, and has greater potential.

Keywords

Software Effort, Stage Effort Prediction, Metabolic GM (1,1) Model

基于改进的新陈代谢GM (1,1)模型的软件阶段成本预测

王　勇，韩佩佩

中国海洋大学，山东　青岛
Email: markwy@126.com

收稿日期：2017年5月10日；录用日期：2017年5月29日；发布日期：2017年6月1日

摘　要

目前，关于软件成本预测的研究主要集中在对总成本的预测，对软件项目阶段成本的预测较少，然而软

文章引用: 王勇, 韩佩佩. 基于改进的新陈代谢 GM (1,1)模型的软件阶段成本预测[J]. 软件工程与应用, 2017, 6(3): 49-57. https://doi.org/10.12677/sea.2017.63006

件行业对此有强烈的需求。为此，本文研究了使用灰色理论的GM (1,1)模型进行软件阶段成本的预测，并对GM (1,1)的新陈代谢模型进行了改进，动态选择模型初始条件，并提出了一种软件项目阶段成本的预测方法IGM。在三个不同数据集上的实验证明IGM方法优于传统新陈代谢GM (1,1)模型、GV方法和LR模型，显示出较大的潜力。

关键词

软件成本，阶段成本预测，新陈代谢GM (1,1)模型

1. 引言

随着信息时代、知识经济时代的到来，计算机行业发展迅速，软件产品已经渗入到人们的生活、工作、学习当中。但随着软件规模越来越大，软件复杂度也不断增加，导致在上个世纪 60 年代中期，爆发了严重的软件危机。世界范围内大量软件项目不能按期完成或者被迫取消，很多项目虽然完成但严重超支。例如，2014 年投资 8.4 亿美元的美国健康医保网站超支 1.63 亿美元且交付日期拖延半年后仍无法正常运行。出现如此严重问题的一个重要原因是对软件成本估算不足，或在项目早期所做的总成本预测随着项目的进展变得越来越不实际[1]。因此，为减少软件失败造成的巨大损失，除了要对项目总成本进行估算外，还需要在开发过程中对项目的阶段成本进行预测，使项目在规定的时间和预算内完成。

软件成本估算是指根据软件项目的计划以及其他影响因子的信息，进行估算和确定各项活动的成本以及总成本的软件项目管理活动[2]。软件阶段成本的估算是指对开发过程中每个阶段的成本进行估算。由于在软件实际开发过程中，项目经理一般是按周、月或季度等物理时间来记录工作量，管理软件开发过程。因此本文采用月作为阶段单位，例如，整个项目持续了 10 个月，则我们称该项目共有 10 个阶段。

科学合理的估算软件项目的阶段成本，有助于项目管理人员来决定每个活动阶段的成本，进而决定软件各开发阶段的人员分配及工作计划。在开发过程中动态地预测各阶段的成本，还可以使项目管理人员发现可能出现的成本进度方面的问题，提前采取措施，减少不必要的损失。

目前，大部分的研究者的关注点在总成本预测上，对总成本预测的方法和模型已经有很多研究成果[3]-[9]，但对于软件项目阶段成本的预测的研究非常少[10] [11] [12]。MacDonell 等人通过建立线性回归模型，根据项目前期阶段的成本来预测后面开发阶段的成本，对 16 个类似软件项目的任意两个开发阶段成本做相关性检验，选择相关程度较大的两个开发阶段建立回归模型。例如，对计划和测试阶段建立了回归模型，根据前者的成本数据预测了后者的成本[10]。王勇等人提出 GV(GM (1,1)和 Verhulst 相结合)方法用来预测软件阶段成本，根据软件项目阶段成本序列的凹凸性来动态构建模型，进而预测后续阶段的成本[11]。与本文研究不同的是，MacDonell 的研究针对软件生命周期阶段，如设计、编码阶段等，而本文的阶段是以物理时间如月为单位。王勇等人的研究[11]使用了 GM (1,1)和 Verhulst 的新陈代谢模型，但模型的初始条件始终是建模序列的第一个元素。本文对初始条件进行了改进，提出了一种新的软件阶段成本预测方法 IGM，研究结果表明，IGM 具有比对比方法更好的预测性能，显示出一定的潜力。

2. 改进的新陈代谢 GM (1,1)模型

2.1. 新陈代谢 GM (1,1)模型

GM (1,1)模型主要用来模拟准指数序列，其预测原理为：是将无规律或规律性不强的的原始数据进行累加，得到规律性较强的准指数序列后建模。将建模生成的模拟序列进行累减后得到原始序列的模拟值[13]。

在利用灰色模型建模的过程中，随着时间的推移，系统逐渐发生变化，老的数据已不能反映当前系统的特征。所以在建模时，引进新信息的同时要将老信息去除，这样得到的模型称为新陈代谢模型[14]。其概念为：

设原始序列 $X^{(0)}=\left(x^{(0)}(1),x^{(0)}(2),x^{(0)}(3),\cdots,x^{(0)}(n)\right)$ 置入最新信息 $x^{(0)}(n+1)$，去掉最老信息 $x^{(0)}(1)$，称用 $X^{(0)}=\left(x^{(0)}(2),\cdots,x^{(0)}(n),x^{(0)}(n+1)\right)$ 建立的模型为新陈代谢灰色模型。

具体的建模过程为：

步骤 1：设原始序列 $X^{(0)}=\left(x^{(0)}(1),x^{(0)}(2),x^{(0)}(3),\cdots,x^{(0)}(n)\right)$，通过一次累加运算之后生成的累加序列(1-AGO)为：$X^{(1)}=\left(x^{(1)}(1),x^{(1)}(2),x^{(1)}(3),\cdots,x^{(1)}(n)\right)$。

其中：

$$x^{(1)}k=\sum_{i=1}^{k}x^{(0)}(i),\quad k=1,2,\cdots,n \tag{1}$$

$x^{(1)}$ 的均值生成序列为：

$$Z^{(1)}=\left(z^{(1)}(2),z^{(1)}(3),z^{(1)}(4),\cdots,z^{(1)}(n)\right) \tag{2}$$

其中：

$$z^{(1)}(k)=\frac{1}{2}\left(x^{(1)}(k)+x^{(1)}(k-1)\right),\quad k=2,3,\cdots,n \tag{3}$$

步骤 2：建立灰微分方程：

$$x^{(0)}(k)+az^{(1)}(k)=b \tag{4}$$

其中，a 为发展系数，b 为灰色作用量。

式(4)的白化微分方程为

$$\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t}+ax^{(1)}=b \tag{5}$$

步骤 3：求参数向量

对于式(5)中的参数向量 $\hat{a}=[a,b]^{\mathrm{T}}$ 利用最小二乘法估计得

$$\hat{a}=\left(B^{\mathrm{T}}B\right)^{-1}B^{\mathrm{T}}Y \tag{6}$$

其中：

$$B=\begin{bmatrix}-z^{(1)}(2) & 1\\ -z^{(1)}(3) & 1\\ \vdots & \vdots\\ -z^{(1)}(n) & 1\end{bmatrix},\quad Y=\begin{bmatrix}x^{(0)}(2)\\ x^{(0)}(3)\\ \vdots\\ x^{(0)}(n)\end{bmatrix}$$

步骤 4：求解方程

继续求解微分方程(5)，得到：

$$\hat{x}^{(1)}(t)=Ce^{-at}+\frac{b}{a} \tag{7}$$

其中：C 为待定常数。

为求解常数 C，需要选定一个初始值，假定 $\hat{x}^{(1)}=x^{(1)}(1)=x^{(0)}(1)$，则有

$$\hat{x}^{(1)}(1)=Ce^{-a}+\frac{b}{a}=x^{(0)}(1)$$

$$C=\left[x^{(0)}(1)-\frac{b}{a}\right]e^{a} \tag{8}$$

代入式(7)得

$$\hat{x}^{(1)}(t)=\left[x^{(0)}(1)-\frac{b}{a}\right]e^{-a(t-1)}+\frac{b}{a} \tag{9}$$

则传统 GM (1,1)模型的时间响应式为

$$\hat{x}^{(1)}(k)=\left[x^{(0)}(1)-\frac{b}{a}\right]e^{-a(k-1)}+\frac{b}{a},\quad k=1,2,\cdots,n \tag{10}$$

步骤 5：再求出模型的还原数据 $\hat{x}^{(0)}(k)$

$$\hat{x}^{(0)}(k)=\hat{x}^{(1)}(k)-\hat{x}^{(1)}(k-1),\quad k=1,2,\cdots,n \tag{11}$$

2.2. 改进的新陈代谢 GM (1,1)模型

传统的 GM (1,1)模型公式如式(10)中始终存在 $x^{(0)}(1)$，这就等于让拟合曲线必须过点(1，$x^{(0)}(1)$)。最小二乘的原理并不要求拟合曲线过第一个数据点，针对初始条件的选取，党耀国等人按照灰色系统理论中新信息优先原理，选取最新点即 $X^{(1)}$ 的最后一个分量 $x^{(1)}(n)$ 作为初始条件[15]。这个改进在一定的条件下可以提高预测精度，但由于最小二乘拟合的曲线理论上并不要求通过某一个点，所以本文提出了一种更全面的初始条件选择方法。依次选取累加序列 $X^{(1)}$ 中的各元素作为初始条件，在训练集上建模，选择其中误差最小的元素 $x^{(1)}(i)$ 作为初始条件，在测试集上预测，这个元素所在的位置索引 i 记为 I。

设 $B,Y,\hat{a}$ 如传统 GM (1,1)模型所定义，取误差最小的元素 $x^{(1)}(i)$ 作为初始条件，则灰微分方程 $x^{(0)}(k)+az^{(1)}(k)=b$ 的白化方程 $\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t}+ax^{(1)}=b$ 的时间响应式为：

$$\hat{x}^{(1)}(k)=\left[x^{(1)}(i)-\frac{b}{a}\right]e^{-a(k-i)}+\frac{b}{a},\quad k=1,2,\cdots,n \tag{12}$$

还原值为：

$$\hat{x}^{(0)}(k)=\hat{x}^{(1)}(k)-\hat{x}^{(1)}(k-1),\quad k=2,3,\cdots,n \tag{13}$$

称式(12)和(13)为改进的新陈代谢 GM (1,1)模型，即 IGM (1,1)模型。

3. 基于改进的新陈代谢 GM (1,1)模型的软件阶段成本预测过程

本文中利用改进的新陈代谢 GM (1,1)模型即 IGM (1,1)模型来建模预测软件项目阶段成本，并使用误差补偿的技术来提高预测精度，并将该软件阶段成本的预测方法称为 IGM 方法。

利用 IGM 方法预测软件项目阶段性成本的基本过程为：

1) 选取建模子序列；

2) 对建模序列进行光滑性处理；

3) 在训练集上用 IGM (1,1)模型建模预测，得到误差最小的 I 和误差补偿值；

4) 在测试集上，以 I 所在的元素为初始条件，建立 IGM (1,1)模型并预测；

5) 用误差补偿值来修正预测结果，得到最优预测值和预测误差。

在本文真实的数据集上预测过程如图 1 所示：

下面将详细介绍建模子序列的选取、数据的光滑性处理、建模预测、求误差补偿值、测试集的建模预测及误差补偿、计算预测误差的过程。

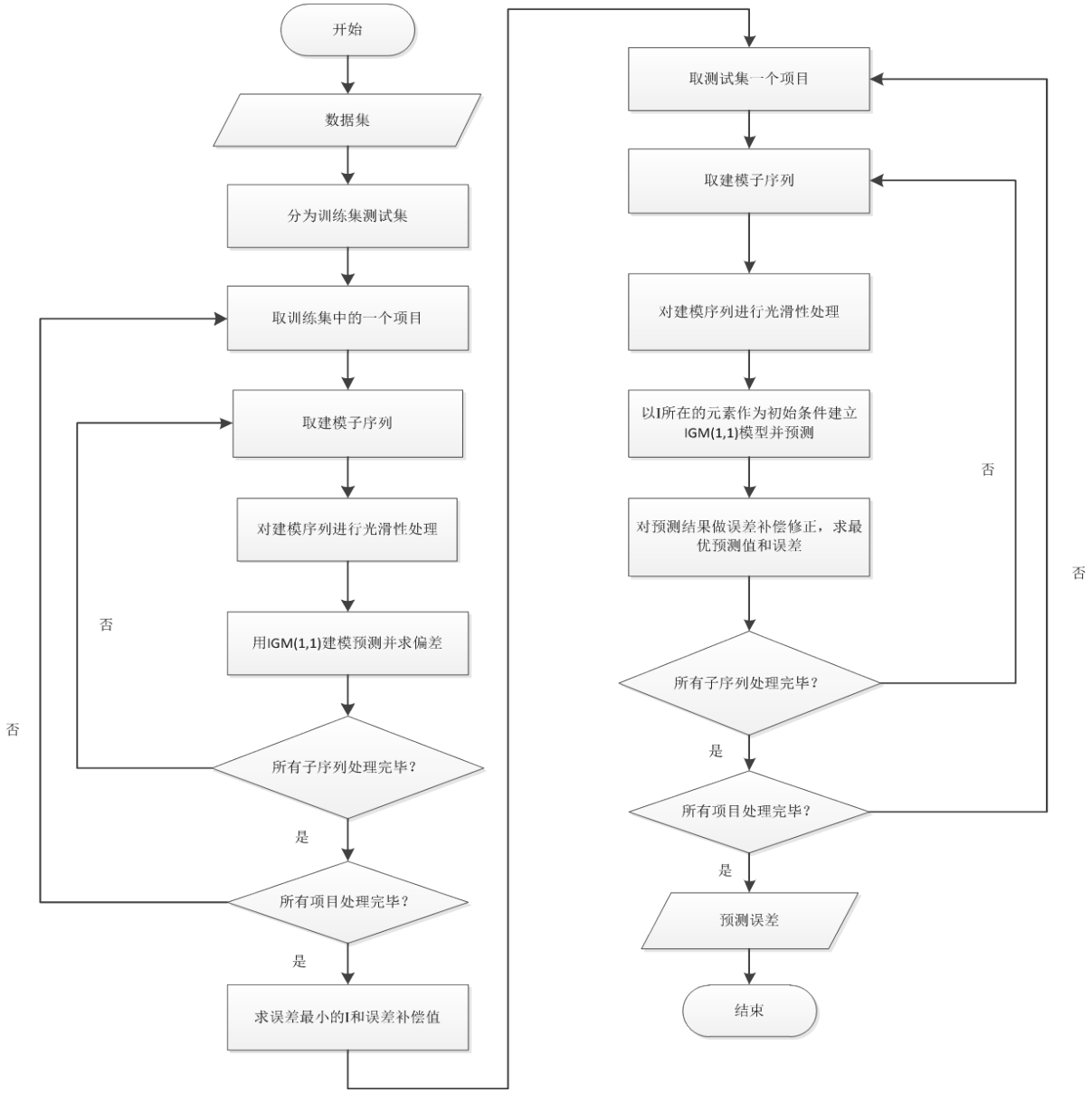

Figure 1. The prediction process of IGM method
图 1. IGM 方法的预测过程

1) 选取建模子序列

在本文的软件项目月工作量预测中，由于在软件开发的前期工作量数据较少，所以我们只选择 3 个月的工作量数据建模。当有新数据加入时，就舍弃最老的一个数据，始终保持使用最近 3 个阶段的数据，形成滚动预测。

例如，某一个项目的规模为 n 个月，则需要对 n 个工作量数据转变成 $(n-3)*4$ 的矩阵形式，对矩阵中每一行的前三个数据进行建模，利用第四个数据来验证误差大小。

假设一个持续时间为 n 个月的项目工作量序列为： $X=(x_1,x_2,\cdots,x_n)$，将其转变成的 $(n-3)*4$ 的矩阵为：

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_3 & x_4 & x_5 \\ \vdots & \vdots & \vdots & \vdots \\ x_{(n-3)} & x_{(n-2)} & x_{(n-1)} & x_{(n)} \end{bmatrix}$$

2) 建模序列的光滑性处理

原始数据序列是影响灰色模型预测精度的重要因素之一。过于离散的数据直接建模，一般不容易拟合而造成预测误差很大。所以要对建模序列进行光滑性处理，光滑性处理可以减少序列中的噪声波动。采用的方法为 n 阶移动加权平均，它对数据序列有修匀和平滑的作用。

假设原序列为 $X=(x_1,x_2,x_3)$，经过光滑性处理后的序列为：

$$\frac{2x_1+x_2}{3},\frac{x_1+x_2+x_3}{3},\frac{x_2+2x_3}{3}$$

3) 建模预测

对经过光滑性处理的建模序列在训练集上运用 IGM (1,1)模型进行建模预测，得到误差最小的 I。

4) 求误差补偿值

对训练集中每个项目的每一个建模子序列进行建模预测，并计算出每一次预测的偏差，将训练集中所有的项目都按此过程操作一遍。最后，将所有的偏差值排序，为排除极值的影响，舍弃偏差序列头尾的 10%，将中间的 80%求出均值作为误差补偿值 $Bias_0$。

5) 在测试集上建模预测及误差补偿

在测试集中，用 I 所在的元素作为初始条件建模预测并利用训练时得到误差补偿值 $Bias_0$ 来修正预测结果，得到最优预测。 最优预测值 $=\text{IGM}(1,1)$ 模型预测值$/(1-Bias_0)$。

6) 计算预测误差

对测试集中的每个项目的每个建模序列求出预测误差，每个项目的预测误差是该项目中所有建模序列预测误差的平均值。该数据集的预测误差为所有项目预测误差的均值。

4. 实验与结果分析

为了检验 IGM 方法的实用性，本文将 IGM 方法应用在三个不同工业类型项目的数据集上进行了验证。

(1) 数据集。本文使用的数据集来自 EDS 公司，共包括 14000 多个真实的软件工程项目，是世界上最大的软件工程数据集之一。数据来自 30 多个国家和地区，包括美国(56.7%)、澳大利亚(7%)、瑞士(5%)、加拿大(4.6%)等，开发语言 130 多种，有 COBOL (10.9%)、JCL (9.4%)、SQL (5.2%)、Visual Basic (4.5%)等，并涉及多个工业类型，有金融、制造、运输等。研究中我们把项目按工业类型分类，以其中最典型

的 3 个数据集为代表进行实验。由于本实验需要用连续的 3 个月的成本数据建模，预测第 4 个月的成本，因此小于或等于 3 个月的项目因数据不足，将从数据集中删除。各实验数据集的项目数量如表 1 所示。

(2) 确认方法。本实验采用交叉验证中的 Hold-Out 方法。将每个数据集的项目随机均分为五份，选取其中的四份作为训练集，剩余的一份作为测试集。首先对训练集上的每个项目用 IGM 方法预测，并求出误差最小的 I 和误差补偿值，在测试集中，以 I 所在的元素作为初始条件，建立 IGM (1,1)模型预测，并利用误差补偿值对预测结果进行修正，得到最优预测。

(3) 评估标准。本文为验证 IGM 方法的性能，使用在软件成本预测领域通常用到的 MMRE、Bias 两种指标作为评价标准。

Bias 表示预测值与真实值之间的偏离程度，对被预测值 x_i，对应的 *Bias* 定义如下：

$$Bias_i = \frac{x_i - \hat{x}_i}{x_i} \times 100\% \tag{14}$$

其中，x_i 为真实值，$\hat{x}_i$ 为预测值，*Bias* 越接近 0，表示预测的精度越高。

相对误差(Magnitude of Relative Error, MRE)定义如下：

$$MRE_i = \frac{\left|x_i - \hat{x}_i\right|}{x_i} \times 100\% \tag{15}$$

多次预测的平均相对误差(Mean Magnitude of Relative Error, MMRE)定义如下：

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \tag{16}$$

(4) 基准方法。为了比较 IGM 方法的性能，在实验中采用其他三种方法作为基准方法进行相同的实验。基准方法一：采用传统的新陈代谢 GM (1,1)模型，记为 GM1；基准方法二：采用王勇等人提出的 GV 方法；基准方法三：采用 MacDonell 在[10]中用到的线性回归(LR)模型。四种方法均在相同的数据集上进行实验。

(5) 实验结果。四种方法所得到的 MMRE 和 Bias 两项指标的值，分别列在表 2，表 3 中。

Table 1. The number of items per data set
表 1. 各数据集的项目数量

数据集	工业类型	项目数量
1	金融	234
2	制造	1036
3	运输	177

Table 2. The MMRE of four methods on different data sets
表2. 四种方法在不同数据集上的 MMRE

数据集	MMRE (%)			
	IGM	GM1	GV	LR
1	45.42	54.94	46.38	50.12
2	74.64	125.15	79.55	88.63
3	53.05	70.73	56.43	63.30

Table 3. The Bias of four methods on different data sets
表3. 四种方法在不同的数据集上的 Bias

数据集	Bias (%)			
	IGM	GM1	GV	LR
1	29.93	−29.87	31.30	32.86
2	33.91	−97.63	39.64	50.50
3	36.05	−41.48	35.83	36.83

实验结果显示，在 3 个数据集上，IGM 方法相比其他三种方法取得了更好的预测精度。从表 2 的 MMRE 指标来看，IGM 方法在 3 个数据集上都表现出一定的优势，表现全部优于其余三种方法。相比其他三种方法，MMRE 最少降低了 20.96%，2.11%，10.35%。其中数据集 2 上四种方法得到的 MMRE 都比较大，检查数据发现原始数据序列有异常波动，导致预测误差偏大。从表 3 的 Bias 指标来看，IGM 方法在数据集 2 上优势明显，均优于其他三种方法。在数据集 1 上，IGM 方法的 Bias 值比 GM1 略差，但相差不多，只有 0.06%。在数据集 3 中，GV 与 IGM 两种方法的 Bias 指标取值接近，GV 比 IGM 低 0.22%，略占优势。

总体来看，在本文的 3 个数据集上，IGM 方法在 MMRE 和 Bias 两个指标上都取得了较好的成果，均超过了传统的新陈代谢 GM (1,1)模型，GV 方法，LR 模型，有较大的潜力。

5. 结论

本文对传统新陈代谢 GM (1,1)模型进行了改进，动态选择模型初始条件，并提出了一种软件项目阶段成本的预测方法 IGM。在 3 个大规模软件工程数据集上进行了验证，实验证明 IGM 方法优于传统的新陈代谢 GM (1,1)模型、GV 方法、LR 模型，显示出较好的预测性能，可作为软件阶段成本预测的一种可选方法。

基金项目

本论文得到国家自然科学基金面上项目(61170312)及软件工程国家重点实验室开发基金项目(SKLSE2012-09-14)的支持。

参考文献 (References)

[1] Jrgensen, M. and Shepperd, M. (2007) A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, **33**, 33-53. https://doi.org/10.1109/TSE.2007.256943

[2] Shepperd, M. (2007) Software Project Economics: A Roadmap. *Proceedings of the FoSE 2*007: *Future of Software Engineering*, Minneapolis, 23-25 May 2007. https://doi.org/10.1109/fose.2007.23

[3] Shepperd, M. and Schofield, C. (1997) Estimating Software Project Effort Using Analogies. *IEEE Transactions on Software Engineering*, **23**, 736-743. https://doi.org/10.1109/32.637387

[4] Tadayon, N. (2005) Neural Network Approach for Software Cost Estimation. *Proceedings of the International Conference on Information Technology*: *Coding and Computing*, Las Vegas, 4-6 April 2005. https://doi.org/10.1109/itcc.2005.210

[5] Srinivasan, K. and Fisher, D. (1995) Machine Learning Approaches to Estimating Software Development Effort. *IEEE Transactions on Software Engineering*, **21**, 126-137. https://doi.org/10.1109/32.345828

[6] Briand, L.C., Emam, K.E. and Surmann, D. (1999) An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques. *Proceedings of the* 1999 *International Conference on Software Engineering*, 22 May 1999.

[7] Molokken, K. and Jorgensen, M. (2003) A Review of Software Surveys on Software Effort Estimation. *Proceedings of*

the 2003 *International Symposium on Empirical Software Engineering*, Rome, 30 September-1 October 2003. https://doi.org/10.1109/isese.2003.1237981

[8] Hughes, R.T. (1996) Expert Judgement as an Estimating Method. *Information & Software Technology*, **38**, 67-75. https://doi.org/10.1016/0950-5849(95)01045-9

[9] 何晓阳，王亚沙. 基于模型的软件成本估计方法[J]. 计算机研究与发展, 2006, 43(5): 777-783.

[10] Macdonell, S.G. and Shepperd, M.J. (2003) Using Prior-Phase Effort Records for Re-Estimation during Software Projects. *Proceedings of the* 9*th International Software Metrics Symposium*, Sydney, 3-5 September 2003. https://doi.org/10.1109/metric.2003.1232457

[11] Wang, Y., Song, Q., Macdonell, S., *et al.* (2009) Integrate the GM (1,1) and Verhulst Models to Predict Software Stage Effort. *IEEE Transactions on Systems, Man and Cybernetics Part C*: *Applications and Reviews*, **39**, 647-658. https://doi.org/10.1109/TSMCC.2009.2020690

[12] 王勇，宋擒豹，沈钧毅. 根据序列变化率预测软件阶段成本[J]. 计算机学报, 2009, 32(7): 1346-1355.

[13] 刘思峰，杨英杰，吴利丰. 灰色系统理论及其应用[M]. 北京：科学出版社, 2014.

[14] 陈霞，邱桃荣，魏玲玲. GM (1,1)模型和新陈代谢模型的应用比较[J]. 微计算机信息, 2008, 24(12): 163-165.

[15] 党耀国，刘思峰，刘斌. 以 x~((1))(n)为初始条件的 GM 模型[J]. 中国管理科学, 2005, 13(1): 133-136.