

考虑风险相关性的软件风险多目标优化控制研究

吴登生¹, 李建平¹, 孙晓蕾¹, 宋浩²

(1. 中国科学院 科技政策与管理科学研究所, 北京 100190; 2. 山东财经大学 数学与数量经济学院, 济南 250014)

摘要 研究软件风险控制的理论和方法, 对提高软件开发成功率有着重要作用。面对软件风险管理的精细化要求, 已有的单目标风险控制模型难以有效管理软件风险。本文将软件风险控制成本和软件风险暴露值作为控制目标, 提出软件风险多目标优化控制模型。进一步将风险损失层面的相关性纳入风险控制模型, 建立了考虑风险损失非可加性的风险控制多目标优化模型, 刻画软件风险管理实践中存在的风险相关性问题。采用多目标粒子群算法对风险控制多目标优化模型进行求解, 并采用一个软件开发项目的风险控制问题进行案例分析。分析结果表明, 在软件风险暴露和软件风险控制成本两个目标之间近似呈现非线性置换关系, 可以根据项目中风险控制成本的实际情况, 快速找到对应的最优风险控制策略。考虑风险相关性的风险控制模型能刻画出软件风险管理实践中更加复杂的关系, 给出更加符合实际情况的风险控制策略, 对提高软件风险管理水平有着重要意义。

关键词 软件风险; 风险相关性; 风险控制; 多目标优化; 多目标粒子群算法

Multi-object risk control model for software project considering risk correlation

WU Deng-sheng¹, LI Jian-ping¹, SUN Xiao-lei¹, SONG Hao²

(1. Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, China; 2. School of Mathematics and Quantitative Economics, Shandong University of Finance and Economics, Jinan 250014, China)

Abstract Research on software risk control theories and methods play an important role in improving the success rate of software project. With the requirement of software risk refined management, the existing single-objective risk control model is difficult to manage software risks effectively. This paper regards software risk control cost and software risk exposure as control objectives, and proposes a multi-object risk control model for software project. Furthermore, it puts forward a multi-object risk control model for software project considering risk correlation. The multi-objective particle swarm optimization is applied to solve the risk control model. Moreover, the example is applied to a medium-sized software project. Case analysis results show that there is a nonlinear trade-off relationship approximately between the software risk exposure and software risk control effort. The software manager can identify the corresponding optimal risk control strategy according to the actual risk control cost. The results also indicate that the consequence from the multi-object risk control model for software project considering risk correlation can better describe the actual situation of risk management.

Keywords software risk; risk correlation; risk control; multi-objective optimization; multi-objective particle swarm optimization (MOPSO)

0 引言

近年来, 软件行业出现了大量以软件项目进度延期、预算超支和质量缺陷为典型特征的软件危机。The Standish Group 发布的《CHAOS Summary 2009》显示, 只有 32% 的软件项目是成功的, 24% 的软件项目是失败的, 剩下的 44% 的软件项目是被质疑的^[1]。从国内外软件项目的开发经验来看, 缺乏有效的软件风险

收稿日期: 2013-08-26

资助项目: 国家自然科学基金 (71201156, 91218302); 中科院科技政策与管理科学研究所重大研究任务 (Y201161Z05); 中国科学院青年创新促进会基金

作者简介: 吴登生 (1984-), 男, 安徽庐江人, 助理研究员, 博士, 研究方向: 风险管理, E-mail: wds@casipm.ac.cn.

管理是出现大量软件危机的一个重要因素。Cerpa 等人调查了 70 个失败的软件项目,总结了 18 条主要影响因素,其中与风险管理有关的因素有“风险在项目中未予重新评估、控制和管理”和“风险未列入项目计划中”,涉及到的项目分别占项目总数的 75.70% 和 70.00%^[2]。软件风险管理的目的是及时识别出软件开发过程中的风险因素,并对其进行综合评估,针对关键风险采取合适的应对措施,将风险对项目的负面影响降至最低程度,从而保证软件项目的成功。

作为软件风险管理的关键步骤,风险控制是指在风险评估的基础上,选择合适的应对措施和方法,对风险进行减缓、转移和规避,起到控制风险的作用。当风险发生后,组织需要调用相关资源进行应急管理,此时的资源使用效率和风险管理效果都难以最优化。为了更好地管理风险,需要将相关控制措施前移化,在风险没有发生之前,进行识别和度量,并进行有效控制。软件风险控制最重要的功能就是在风险识别、评估的基础上,结合组织自身的资源和管理水平,对风险控制方案进行优化。在软件风险管理研究中,已有部分针对软件风险控制的研究成果。Iversen 等人运用实物期权理论对软件风险进行建模,提出软件风险控制的若干措施^[3]。陈涛等人在螺旋模型的基础上提出了集成风险管理的软件项目开发过程模型,并在实物期权理论框架下为风险控制提供了相关措施^[4]。Ben-David 等人将项目内容、风险事件、风险缓释行动和效果整合到一个模型框架中,通过将软件项目的工作进行分解,给出各个风险缓释策略的影响,从而计算各种风险缓释方案中的风险总暴露值^[5]。潘春光等人从利润最大化角度研究了风险控制策略优选问题,提出基于成本考虑的软件风险控制策略优选模型^[6]。杨莉针对软件项目风险应对计划中风险应对措施的多目标优选问题,提出一种区间优化模型,结合考虑风险管理者的风险偏好,选出满意的风险应对措施组合^[7]。Zhang 等人利用优化模型对风险控制策略进行优选,在现有约束条件下得到最优控制策略^[8]。上述软件风险控制研究成果都是基于单个目标进行风险控制策略优化的,虽然为指导软件风险管理实践提供有益参考,但是忽略了风险控制实践中需要同时考虑风险值和控制成本这一多目标控制问题。

在软件开发实践过程中,由于人力、资金、时间等方面的约束限制,不能对所有识别出的风险进行全面控制,一般会采取有限控制的策略,即选择主要风险进行控制,如常用的风险列表 (Checklist)^[9]。所以在对软件风险进行控制时,不光要考虑风险本身暴露值的大小,同时还需要考虑进行风险控制投入的成本等其他因素,使得多个目标最优,这就引入了软件风险多目标控制问题。在软件风险多目标控制方面的成果很少。Brito 等研究多目标下风险控制策略,通过优化算法同时优化开发成本和可信度两个目标,来改善软件开发过程^[10]。Li 等人采用贝叶斯网络分析了软件风险多目标控制问题,在同时考虑风险控制成本和特定过程风险等多个目标下,优化软件风险控制策略集,取得了很好的效果^[11]。宋浩在构建风险控制模型时同时考虑了风险暴露值和风险控制成本,并将风险控制成本作为约束条件对模型进行求解^[12]。上述研究在软件风险控制过程中初步考虑了多个控制目标,在软件风险控制领域做了有益尝试,但是研究成果还不够深入,其中最重要的是没有考虑风险之间的相关性。此外,上述研究在求解软件风险多目标控制模型时,通常是将其他目标转为约束条件,将多目标优化问题转为单目标优化问题,一定程度上降低了模型的实用性。

软件风险随着时间的推移会发生风险累积,迁移或扩散等现象,彼此间会相互作用和相互影响,例如为了减少软件缺陷率却使软件成本大大超支。忽略软件风险之间的这种重要联系,会使得到的风险控制策略过于理想化。已有研究中也有一部分成果开始关注风险之间的这种相关性。Song 等人在文献调研的基础上,采用 Meta-analysis 分析的方法,研究了不同风险之间相关性^[13]。Buyukozkan 等人采用模糊测度理论来刻画风险之间的相关性,将风险之间相关性分成正相关、负相关和不相关三类^[14]。Fang 等人采用结构设计矩阵 (DSM) 来刻画风险之间的相关性,进一步构建风险网络进行实证分析^[15]。苏越良等人在研究风险控制问题时,考虑了风险因素之间复杂的非线性作用机制,提出考虑风险相关性的风险控制模型^[16]。Borgonovo 等人提出了一种新的指标来度量风险之间的交互作用^[17]。罗长青等人针对风险相关性的时变性,采用变结构 Copula 函数来刻画信用风险相关性^[18]。已有的成果对风险之间相关性进行初步的探讨,但是没有对相关性机理进行深入总结,影响了对风险管理实践的指导力度。此外,现有成果中还没有涉及到软件风险控制模型中风险相关性问题的。

所以本文在构建软件风险控制模型时,将传统的控制目标从一个拓展到多个,构建软件风险多目标控制模型,进一步对软件风险之间的相关性机理进行总结,在软件风险多目标控制模型中从风险损失程度层面考虑风险之间的相关性,对于丰富软件风险控制理论和模型,提高软件风险管理水平方面起到积极作用。

1 软件风险控制多目标优化模型

定义风险为某一特定环境下,在某一特定时间段内,某种损失发生的可能性,涉及到风险发生概率和风险损失程度两个方面.设定软件风险集为 $\{R_i | i = 1, 2, \dots, n\}$, n 为识别出来的风险个数, P_i 为第 i 种风险 R_i 发生的可能性, L_i 为第 i 种风险 R_i 发生后产生的损失;风险暴露为风险发生的可能性和产生损失的综合值,一般定义为 $RE_i = P_i \times L_i^{[19]}$.

已有的软件风险控制研究成果中,通常是将风险暴露最小化或者利润最大化作为其优化目标^[6-7],在优化过程中,会考虑到控制策略的成本,并将控制成本作为优化模型中的一个约束条件.本文在总结现有软件风险控制模型的基础上,将软件风险控制目标定为总风险暴露最小和风险控制成本最低.定义总风险暴露值为各个风险暴露值的总和,具体为 $REI = \sum_{i=1}^n RE_i$.每一个风险控制成本定义为 RC_i ,总风险控制成本为各个风险控制成本之和,即 $RC = \sum_{i=1}^n RC_i$.软件风险管理实践中的经验表明,要控制住暴露值越大的风险,投入的控制成本往往越高,即同时使得风险暴露值最小和风险控制成本最低成为一对相互冲突的目标,所以适用多目标优化模型来分析风险控制策略问题.

在确定风险控制目标后,引入风险控制决策变量函数 x_i ,表示对风险是否控制. $x_i = 1$: 对第 i 种风险 R_i 实施控制; $x_i = 0$: 对第 i 种风险 R_i 不实施控制.由于不实施控制的风险,其暴露值不会发生变化,也不会产生风险控制成本,所以在原先的风险控制目标函数中需要引入风险控制决策变量,具体定义见公式(1)和公式(2).

$$REI = \sum_{i=1}^n (1 - x_i) \times RE_i = \sum_{i=1}^n (1 - x_i) \times P_i \times L_i \quad (1)$$

$$RC = \sum_{i=1}^n x_i \times RC_i \quad (2)$$

此外,风险控制决策变量只能选取 0 或者 1,即构建的多目标优化模型是一个 0-1 整数规划模型,具体见公式(3).

$$\begin{aligned} \min \quad & REI \\ \min \quad & RC \\ \text{s.t.} \quad & x_i = 0 \quad \text{or} \quad 1 \end{aligned} \quad (3)$$

2 考虑风险相关性的软件风险多目标控制建模

在软件风险控制的实践过程中,还存在一种特殊的现象,即风险之间存在着相关性.传统的风险控制模型都没有考虑到这种特殊的现象,一定程度上影响了优化结果的准确性和适用性.本文在上述软件风险控制多目标优化模型的基础上,引入风险相关性刻画变量,构建考虑风险相关性的软件风险多目标控制模型,以提高风险控制模型优化结果的适用性.

2.1 风险相关性机理分析与相关性刻画

本文将风险定义为某一特定环境下,在某一特定时间段内,某种损失发生的可能性,涉及到风险发生概率和风险损失程度两个方面.风险之间的相关性机理也可以从发生概率和损失程度两个方面来总结.第一类风险相关性是从风险发生概率层面产生联系的,两个风险之间不是相互独立的,一个风险发生后增加了另一类风险发生的概率.如软件风险中“客户的经验不足”风险发生后,一定程度上会增加“某些需求发生变化”风险发生的概率.这类相关性又称为风险概率相依,已有研究成果一般采用联合概率和条件概率来刻画不同风险之间的相关性,如 Van Dorp 和李建平等人分别采用 Copula 方法刻画了不同风险之间的风险相关性^[20-21], Li 等人采用贝叶斯网络的条件概率分析了风险之间的传染、累积等相关性特征^[11].另一类风险相关性是从风险损失程度层面产生联系的,总风险损失与多个风险损失之间存在着非可加性特征,某两个风险同时发生所产生的损失要大于或者小于两个风险单独发生时所产生的损失之和.如当“客户的经验不足”风险和“成本估算不准”风险同时发生时,由于会影响到客户和开发人员之间存在的沟通效果,其损失肯定要大于两个风险单独发生时的之和.该类相关性可以称为风险损失非可加性,已有学者尝试采用模糊测度来度量不同风险之间的相关性^[14].

在风险相关性机理分析的基础上,需要引入相应的方法对相关性进行刻画.由于本文中风险控制多目标优化模型注重风险控制成本,所以只从风险损失程度层面来刻画风险之间的相关性(风险损失非可加性).在度量非可加性方面,模糊测度一直是通用的方法^[22].模糊测度用较弱的单调性来代替可加性,其主要特征是非可加性,常用的模糊测度包括一般模糊测度、 g_λ 模糊测度、 K 可加模糊测度,不同模糊测度之间刻画能力和复杂度差异较大^[23].作为 K 可加模糊测度中的一种,2可加模糊测度在复杂性和表示能力之间做了相应的折中,较好地解决了复杂性和精度之间的矛盾,且其参数值恰好与模糊测度的Shapley值相等,易于分析各个属性的重要性及相互作用关系,已在实践中被广泛采用^[24-25],故本文选用2可加模糊测度来刻画风险损失之间的相关性.

2.2 考虑风险损失非可加性的风险控制多目标优化模型

在上述风险控制多目标优化模型中要考虑风险损失的这种非可加性,就需要将存在相关性风险同时发生时的影响纳入风险控制优化模型中.设定 $L_{i \cup j}$ 为第 i 种风险 R_i 和第 j 种风险 R_j 同时发生后产生的损失, $P_{i \cup j}$ 表示第 i 种风险 R_i 和第 j 种风险 R_j 同时发生的概率.本文只考虑风险损失层面的相关性,假设风险发生概率是独立的,即 $P_{i \cup j} = P_i \times P_j$.

设定 $L_{i \cup j}$ 为第 i 种风险 R_i 和第 j 种风险 R_j 同时发生后产生的损失,根据两个风险同时发生时损失($L_{i \cup j}$)与单独发生时损失之和($L_i + L_j$)之间的大小关系可以将风险损失非可加性分成三种.当 $L_{i \cup j} > L_i + L_j$,表示风险 R_i 和风险 R_j 之间在损失程度上存在正向相互关系,表现为损失的超可加性质;当 $L_{i \cup j} < L_i + L_j$,表示风险 R_i 和风险 R_j 之间在损失程度上存在负向相互关系,表现为损失的次可加性质;当 $L_{i \cup j} = L_i + L_j$,表示风险 R_i 和风险 R_j 之间在损失程度上无相互关系存在,呈独立状态表现出可加性.已有的风险控制模型中,都是假设不同风险损失之间是可加的.本文在软件风险多目标控制模型中,将风险损失之间的非可加性考虑进来,总风险暴露值控制目标除了考虑各个暴露值的总和 $\sum_{i=1}^n RE_i$ 之外,还需要考虑风险损失非可加性的风险暴露值 REC .此外,根据上述设定,在计算 REC 时要剔除原先单个风险暴露的影响,具体如公式(4)所示.

$$REC = \sum_{i,j \in Q} (1-x_i)(1-x_j) \times (P_i \times P_j) \times (L_{i \cup j} - L_i - L_j) \quad (4)$$

式中 Q 表示存在相关性的风险集合, $i, j \in Q$ 表示风险 R_i 和风险 R_j 之间存在相关性(风险损失非可加性); $(1-x_i)(1-x_j)$ 表示两个风险之间只要有一个风险被控制,就不存在这种相关性作用; $P_i \times P_j$ 表示两个风险在发生概率上是独立的,不存在相关性; $(L_{i \cup j} - L_i - L_j)$ 表示已经剔除原先风险暴露的影响.公式(4)中主要是在风险控制模型中考虑了风险损失的非可加性.在软件开发实践中,由于资源复用等特征,部分风险之间存在着损失的非可加性,而从概率层面来看,风险之间又是相互独立的.

新的考虑风险损失非可加性的风险控制多目标优化模型中,第一个控制目标为总风险暴露最小,即 $REI + REC$ 最小,具体见公式(5).

$$RE = REI + REC = \sum_{i=1}^n (1-x_i) \times P_i \times L_i + \sum_{i,j \in Q} (1-x_i)(1-x_j) \times (P_i \times P_j) \times (L_{i \cup j} - L_i - L_j) \quad (5)$$

新的考虑风险损失非可加性的风险控制多目标优化模型具体如公式(6)所示.

$$\begin{aligned} \min \quad & RE \\ \min \quad & RC \\ \text{s.t.} \quad & x_i = 0 \quad \text{or} \quad 1 \end{aligned} \quad (6)$$

3 基于多目标粒子群算法的模型求解

本文构建的软件风险多目标控制模型实质上是一个多目标优化问题,传统的模型求解方法是将其目标变为约束,将多目标优化问题转化为单目标优化问题,这一定程度上影响了模型的适用范围.近年来,进化多目标优化算法已经成为求解多目标优化模型的主流,包括基于人工免疫系统的多目标优化、基于粒子群优化的多目标优化和基于遗传算法的多目标优化等^[26],其中多目标粒子群算法(multi-objective particle swarm optimization,简称MOPSO)是将帕累托最优与粒子群算法进行综合应用到多目标优化上的一种启发式算法,相比较其他进化多目标优化算法,具有进化机制简单、易于实现、收敛速度快等特点^[27].此外,MOPSO能够得到非劣解集的一个子集构成Pareto前沿面,便于分析软件风险多目标控制模型中不同目标之间的关

系, 决策者可以根据组织的实际情况, 在最优解前沿面上快速找到最佳控制方案^[28]. 因此, 本文采用 MOPSO 算法来求解构建的考虑风险相关性的软件风险多目标控制模型.

MOPSO 引入了外部非劣解集合来存储所有的非劣解, 从一开始不断将非劣解加入到这个集合中, 在每次循环过程中选择其中一个非劣解作为全局最优解. 而每个随机的粒子都会有一个到目前为止的个体最优解, 再结合全局最优解就可以形成下一轮粒子进化的速度和方向. 不断循环以上的过程就可以计算多目标优化模型的帕累托最优解. MOPSO 算法位置更新公式仍相似于单目标优化, 如公式 (7) 和 (8) 所示.

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}} \times t$$

(7)

$$v(t+1) = \omega \times v(t) + c_1 \times rand(0,1) \times (x_{\text{best}} - x) + c_2 \times rand(0,1) \times (x_{\text{gbest}} - x)$$

(8)

其中, T_{\max} 表示最大迭代次数, t 表示当前迭代次数, ω_{\max} 、 ω_{\min} 是 ω 的最小值和最大值, c_1 表示粒子自身最优的影响因子, c_2 是全局最优粒子的影响因子, $v(t)$ 、 $v(t+1)$ 分别表示第 t 和第 $t+1$ 次迭代的速度, x_{best} 指的是粒子自身的最优值, 而 x_{gbest} 指的是全局最优粒子值.

4 案例分析

现有一个软件开发项目需要进行风险控制, 该项目为一个中型软件项目, 其开发商是一个通过 CMMI4 级认证的软件企业. 为了对软件风险进行控制, 前期在充分调研已有软件风险列表的基础上^[29-30], 综合形成了该项目使用的风险列表, 将软件风险分为需求风险、用户风险、开发商风险、项目风险、技术风险和环境风险 6 大类, 并进一步细分为 24 个风险, 具体如表 1 所示. 针对已确定的风险列表发放问卷调查, 对每一个风险发生的可能性和损失进行调研, 共发放调查问卷 110 份, 回收有效问卷 93 份. 风险发生的可能性、损失取调查数据的均值. 并请项目经理与主要开发人员结合项目实际情况对风险的控制成本进行估值. 风险发生可能性是一个概率值, 最大值为 1, 最小值为 0; 风险损失是指该风险发生后对软件项目的整体带来的损失值, 用货币来表示, 单位为百元; 风险控制成本主要是人力成本, 单位为人日. 具体风险名称以及其相对应的参数如表 1 所示.

表 1 软件风险特征属性及控制成本

风险大类	风险名称	风险编号	可能性	损失	控制成本
需求风险	客户的经验不足, 对相关知识不了解	R1	0.57	573	48
	某些需求不正确	R2	0.62	827	34
	某些需求发生变化	R3	0.76	1400	156
	某些系统需求之间有冲突	R4	0.39	907	42
用户风险	用户不合作	U1	0.33	840	158
	用户的部门之间有冲突	U2	0.33	318	69
	用户高层领导的参与和支持不够	U3	0.26	127	52
	用户商业或组织环境不稳定	U4	0.42	191	105
开发商风险	开发人员缺少相应的知识和技能	D1	0.55	636	87
	依赖关键人物	D2	0.48	509	122
	开发团队的士气低	D3	0.31	318	132
	开发团队人员不稳定, 流动率高	D4	0.37	255	140
	开发团队总人数多, 导致难协调沟通不畅	D5	0.41	382	52
项目风险	项目经理的相关经验不丰富	P1	0.46	64	34
	进度压力过大	P2	0.57	636	87
	对成本估计不足	P3	0.25	573	98
	项目人员的角色和任务定义不清楚	P4	0.33	827	73
技术风险	使用了新的技术	T1	0.43	382	105
	需要和其他的系统集成	T2	0.53	445	110
	技术过于复杂	T3	0.33	127	128
	软件和硬件供应商的质量不高	T4	0.31	318	122
环境风险	需要和除用户以外的其他组织机构联系	E1	0.27	111	87
	开发商内部或开发商和用户之间有冲突	E2	0.24	318	78
	对顾问、分包商、供应商控制监管不严	E3	0.19	64	64

利用 Matlab 编程求解 MOPSO 算法, 根据已有研究成果在参数设置方面的建议^[31]并结合试算结果, 相关参数设置如下: $T_{\max} = 10000$, $\omega_{\max} = 1.2$, $\omega_{\min} = 0.1$, $c_1 = c_2 = 0.8$, 非劣解集合容量设置为 100, 初始粒子数为 200, 非劣解筛选密度为 0.05. 算法的计算结果为一组风险控制的帕累托最优解集, 集合中每一个解在两个控制目标上都不能同时优于其他解. 根据计算结果, 一共找到了 44 个帕累托最优解, 解集中对应的风险暴露值与风险控制成本控制目标的关系如图 1 所示.

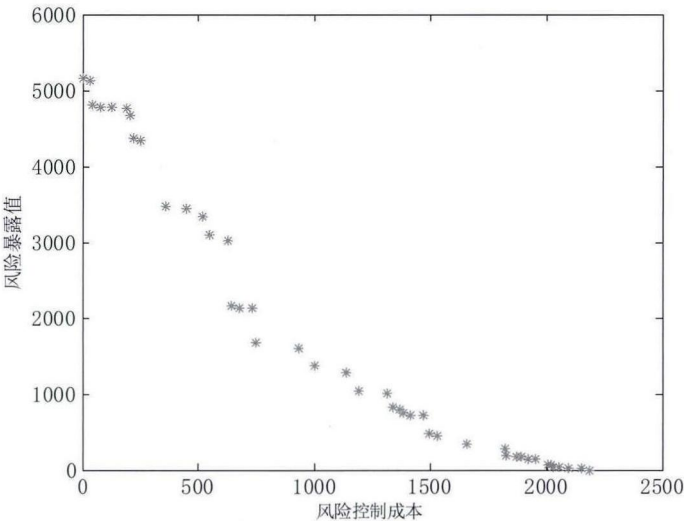


图 1 帕累托最优解集中风险控制目标关系图

从图 1 中可以看出, 风险暴露值控制目标和风险控制成本控制目标近似呈现非线性置换关系. 当控制成本为 0 时, 说明所有的风险都没有得到控制, 此时总风险暴露值最大, 为 5163.3; 当总风险暴露值为 0 时, 表示所有的风险都已经得到有效控制, 此时的风险控制成本最大, 为 2183. 在软件风险管理实践中, 通常情况是有一定的资源来进行风险控制, 但是无法控制所有的风险, 这就需要选择适合的风险对象来进行控制, 已达到最优风险控制效果. 图 1 中的最优解前沿面清晰给出了不同风险控制成本下的最优控制策略, 可以帮助决策者快速做出决策, 制定出软件风险控制应急措施.

现假设开发公司针对该项目有两个风险控制预案, 分别是高投入风险控制预案 (风险控制成本为 1600) 和低投入风险控制预案 (风险控制成本为 800). 为了比较两种不同风险控制预案中最优风险控制策略的差异, 在图 1 中帕累托最优解集中查找风险控制成本与控制预案中风险投入的近似解, 分别找到了第 9 号最优解 (风险控制成本为 1659) 和第 23 号最优解 (风险控制成本为 747). 表 2 中为两种风险控制预案下最优风险控制策略.

表 2 两种风险控制预案中最优风险控制策略

序号	风险编号	高投入控制员	低投入控制预案	序号	风险编号	高投入控制员	低投入控制预案
1	R1	1	1	13	D5	1	0
2	R2	1	1	14	P1	1	0
3	R3	1	1	15	P2	1	1
4	R4	1	1	16	P3	1	1
5	U1	1	0	17	P4	1	1
6	U2	1	0	18	T1	0	0
7	U3	1	0	19	T2	1	0
8	U4	1	0	20	T3	0	0
9	D1	1	1	21	T4	1	1
10	D2	1	0	22	E1	0	0
11	D3	1	0	23	E2	1	0
12	D4	0	0	24	E3	0	0

根据前文的定义, 最优解中决策变量为 1 表示该风险在控制策略中得到控制. 表 2 中的结果表明, 当采用高投入风险控制预案时, 有 19 个风险得到控制, 此时的风险暴露值为 342.65; 当采用低投入风险控制策略时, 有 9 个风险得到有效控制, 此时的风险暴露值为 1679.16. 当风险控制资源发生变化时, 可以根据风险投

入成本值在帕累托最优解集中找到对应最佳风险控制策略。

进一步根据前文风险相关性的定义以及专家调查结果, 识别出 5 对存在相关性的风险集合, 具体如表 3 所示。根据风险同时发生时损失和与单独发生时损失和对比情况, 可以将其分成两类。一类是指存在正相关关系的风险集, 如风险 R3 和 U1, 当风险“用户不合作”与“某些需求发生变化”同时发生时, 因其协调成本的增加, 其风险损失值为 2620, 大于两个风险单独发生时损失和 (2240)。另一类是指存在负相关关系的风险集, 如风险 D1 和 P4。因这两个风险都是关于项目人员, 当同时发生时, 由于存在关键人员, 其损失值 (1100) 要小于两个风险单独发生时损失和 (1463)。已经识别的 5 对存在相关性的风险集中有 3 对是属于正相关, 2 对属于负相关。

表 3 具有相关性的软件风险集对其同时发生时损失

序号	存在相关性的风险集	风险编号	单个风险损失	风险同时发生时损失和
1	某些需求发生变化	R3	1400	2620
	用户不合作	U1	840	
2	开发团队的士气低	D3	318	510
	项目经理的相关经验不丰富	P1	64	
3	使用了新的技术	T1	382	450
	技术过于复杂	T3	127	
4	开发人员缺少相应的知识和技能、必备的培训	D1	636	1100
	项目人员的角色和任务定义不清楚	P4	827	
5	依赖关键人物	D2	509	1380
	进度压力过大	P2	636	

在进行风险控制时, 考虑风险之间存在的这种相关关系, 有助于得到更加符合实际情况的风险控制策略, 提高组织的风险管理水平。根据计算结果, 考虑风险相关性后, 一共找到了 46 个帕累托最优解, 比没有考虑风险相关性时多找到 2 个帕累托最优解集。将考虑风险相关性前后得到的帕累托最优解进行对比, 可以得到如图 2 中结果。

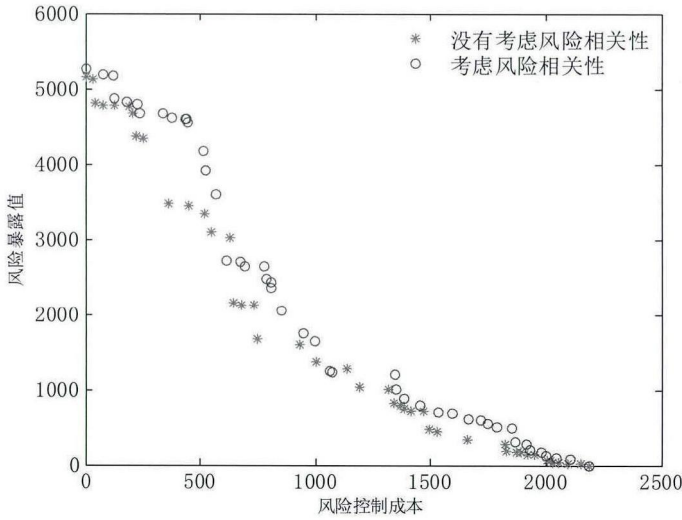


图 2 考虑风险相关性前后的帕累托最优解集中风险控制目标对比图

从图 2 中的结果可以看出, 在考虑风险之间存在的这种相关性后, 所得到的最优解与原先模型的最优解有着较大的区分。根据图中的对比结果, 将最优解前沿面按照风险控制成本坐标分成 4 段, 分别是 $[0, 200)$ 、 $[200, 620)$ 、 $[620, 1000)$ 、 $[1000, 2183]$ 。当风险控制成本在区间 $[0, 200)$ 和 $[1000, 2183]$ 时, 两种模型得到的最优解差异不大。

当风险控制成本在区间 $[200, 620)$ 时, 从最优解结果中可以看出, 此时表 3 中存在相关性的风险中有 R3 和 P1 列入风险控制对象, 相关性结果表明, 这两个风险集对都是存在正相关关系的, 即同时发生后的损失要大于单独发生时的损失之和。所以最优解呈现出图 2 中的情况, 即相同控制成本下, 考虑相关性的风险控制模型得到的风险值要大于没有考虑相关性风险控制模型的结果。当风险控制成本在区间 $[620, 1000)$ 时, 上述存在相关性的风险集中, 主要控制对象为 P4、D1、R3 和 P1, 其相关性有正相关也有负相关, 所以会出

现图2中该区间段,两个模型最优结果先是没有考虑相关性模型的风险值大,后是考虑相关性模型的风险值大。

实证结果表明,在考虑风险相关性前后,软件风险控制优化模型的结果存在较大的差异,传统的研究忽略了这种差异会给软件风险管理带来许多问题。本文构建的考虑风险相关性软件风险控制优化模型能刻画出软件实践中更加复杂的关系,给出更加符合实际情况的风险控制策略,对提高软件风险管理水平有着重要意义。

5 结语

软件风险控制一直是软件风险管理中的关键步骤,本文构建了同时考虑风险暴露值和风险控制成本的多目标风险控制模型,并采用MOPSO算法来求解该多目标优化模型。此外,针对软件风险管理实践中存在的风险相关性问题,总结了其相关性机理,并在已构建的多目标风险控制模型的基础上,进一步建立了考虑风险损失非可加性的风险控制多目标优化模型。结合软件开发实践,选择一个软件开发项目的风险控制问题进行案例分析。分析结果表明,在软件风险暴露和软件风险控制成本两个目标之间近似呈非线性置换关系,当投入的风险控制成本越多,风险暴露值越低。此外,可以根据项目中风险控制成本的实际情况,快速找到对应的最优风险控制策略。对比考虑风险相关性前后的多目标风险控制模型的结果,可以发现因为较多地考虑风险之间的相关性,在不同控制成本区间段内,两个模型的最优结果呈现出较大的差异,进一步分析该差异来源于考虑的风险相关性类型,表明提出的考虑风险相关性软件风险控制优化模型能刻画出软件实践中更加复杂的关系,其结果能更好地刻画风险管理的实际情况。

参考文献

- [1] The Standish Group. CHAOS Summary 2009[EB/OL]. http://www.standishgroup.com/newsroom/chaos_2009.php, 2009.
- [2] Cerpa N, Verner J M. Why did your project fail?[J]. Communications of the ACM, 2009, 52(12): 130-134.
- [3] Iversen J H, Mathiassen L, Nielsen P A. Managing risk in software process improvement: An action research approach[J]. MIS Quarterly, 2004, 28(3): 395-433.
- [4] 陈涛, 丛国栋, 于本海, 等. 基于风险管理的软件开发过程模型及其复合实物期权分析 [J]. 管理工程学报, 2010, 24(2): 61-67.
Chen Tao, Cong Guodong, Yu Benhai, et al. Software process model based on risk management and its real options analysis[J]. Journal of Industrial Engineering and Engineering Management, 2010, 24(2): 61-67.
- [5] Ben-David I, Raz T. An integrated approach for risk response development in project planning[J]. Journal of the Operational Research Society, 2001, 52(1): 14-25.
- [6] 潘春光, 陈英武. 基于CMMI的软件项目风险缓解策略优化模型研究 [J]. 计算机科学, 2007, 34(4): 279-281.
Pan Chunguang, Chen Yingwu. Research on the optimization model of risk reduction actions in the CMMI-based SPRM[J]. Computer Science, 2007, 34(4): 279-281.
- [7] 杨莉, 李南. 软件项目风险应对措施优选的区间模型及其算法 [J]. 控制与决策, 2011, 26(4): 530-534.
Yang Li, Li Nan. Interval model and its algorithm for optimal selecting of software project risk response actions[J]. Control and Decision, 2011, 26(4): 530-534.
- [8] Zhang Y, Fan Z P. An optimization method for selecting project risk response strategies[J]. International Journal of Project Management, 2014, 32(3): 412-422.
- [9] Keil M, Li L, Mathiassen L, et al. The influence of checklists and roles on software practitioner risk perception and decision-making[J]. Journal of Systems and Software, 2008, 81(6): 908-919.
- [10] Brito M, May J. Safety critical software process improvement by multi-objective optimization algorithms[C]// Lecture Notes in Computer Science, 2007, 4470: 96-108.
- [11] Li J P, Li M L, Wu D S, et al. An integrated risk measurement and optimization model for trustworthy software process management[J]. Information Sciences, 2012, 191: 47-60.
- [12] 宋浩. 面向可信的软件开发过程风险管理研究 [D]. 北京: 中国科学院大学, 2011.
Song Hao. Study on software development risk management for software trustworthiness[D]. Beijing: University of Chinese Academy of Sciences, 2011.
- [13] Song H, Cai C, Li M L, et al. Software risks correlation analysis using meta-analysis[C]// Communications in Computer and Information Science, 2009, 35: 559-565.
- [14] Buyukozkan G, Ruan D. Choquet integral based aggregation approach to software development risk assessment[J]. Information Sciences, 2010, 180(3): 441-451.

- [15] Fang C, Marle F. A simulation-based risk network model for decision support in project risk management[J]. *Decision Support Systems*, 2012, 52(3): 635–644.
- [16] 苏越良, 张卫国. 基于全局的复杂产品开发项目风险协调控制方法 [J]. *系统工程理论与实践*, 2008, 28(5): 70–76.
Su Yueliang, Zhang Weiguo. Study on a risk coordination control method based on overall situation for complex product development project[J]. *Systems Engineering — Theory & Practice*, 2008, 28(5): 70–76.
- [17] Borgonovo E, Smith C L. A study of interactions in the risk assessment of complex engineering systems: An application to space PSA[J]. *Operations Research*, 2011, 59(6): 1461–1476.
- [18] 罗长青, 朱慧明, 欧阳资生. 跳跃 - 扩散条件下信用风险相关性度量的变结构 Copula 模型 [J]. *中国管理科学*, 2014, 22(3): 1–12.
Luo Changqing, Zhu Huiming, Ouyang Zisheng. Variable structure Copula models of credit risk correlation under the condition of jump-diffusion process[J]. *Chinese Journal of Management Science*, 2014, 22(3): 1–12.
- [19] Aven T, Renn O. On risk defined as an event where the outcome is uncertain[J]. *Journal of Risk Research*, 2009, 12(1): 1–11.
- [20] Van Dorp J R. Statistical dependence through common risk factors: With applications in uncertainty analysis[J]. *European Journal of Operational Research*, 2005, 161(1): 240–255.
- [21] 李建平, 丰吉闯, 宋浩, 等. 风险相关性下的信用风险、市场风险和操作风险集成度量 [J]. *中国管理科学*, 2010, 18(1): 18–25.
Li Jianping, Feng Jichuang, Song Hao, et al. Integrating credit, market and operational risk based on risk correlation[J]. *Chinese Journal of Management Science*, 2010, 18(1): 18–25.
- [22] Sims J R, Wang Z Y. Fuzzy measures and fuzzy integrals: A survey[J]. *International Journal of General Systems*, 1990, 17(2–3): 157–189.
- [23] Grabisch M, Kojadinovic I, Meyer P. A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package[J]. *European Journal of Operational Research*, 2008, 186(2): 766–785.
- [24] Grabisch M. K-order additive discrete fuzzy measures and their representation[J]. *Fuzzy Sets and Systems*, 1997, 92(2): 167–189.
- [25] 武建章, 张强. 基于 2-可加模糊测度的多准则决策方法 [J]. *系统工程理论与实践*, 2010, 30(7): 1229–1237.
Wu Jianzhang, Zhang Qiang. Multicriteria decision making method based on 2-order additive fuzzy measures[J]. *Systems Engineering — Theory & Practice*, 2010, 30(7): 1229–1237.
- [26] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究 [J]. *软件学报*, 2009, 20(2): 271–289.
Gong Maoguo, Jiao Licheng, Yang Dongdong, et al. Research on evolutionary multi-objective optimization algorithms[J]. *Journal of Software*, 2009, 20(2): 271–289.
- [27] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 256–279.
- [28] Shao X Y, Qi M L, Gao M G. A risk analysis model of flight operations based on region partition[J]. *Kybernetes*, 2012, 41(10): 1497–1508.
- [29] Wallace L, Keil M, Rai A. Understanding software project risk: A cluster analysis[J]. *Information & Management*, 2004, 42(1): 115–125.
- [30] Schmidt R, Lyytinen K, Keil M, et al. Identifying software project risks: An international Delphi study[J]. *Journal of Management Information Systems*, 2001, 17(4): 5–36.
- [31] 王云, 冯毅雄, 谭建荣, 等. 基于多目标粒子群算法的柔性作业车间调度优化方法 [J]. *农业机械学报*, 2011, 42(2): 190–196.
Wang Yun, Feng Yixiong, Tan Jianrong, et al. Optimization method of flexible job-shop scheduling based on multi objective particle swarm optimization algorithm[J]. *Transactions of the Chinese Society for Agricultural Machinery*, 2011, 42(2): 190–196.