

考虑学习/遗忘特性的软件项目调度多目标优化方法

郭一楠^{1,2}, 季俊华^{1†}, 吉建娇¹, 巩敦卫¹

(1. 中国矿业大学 信息与控制工程学院, 江苏 徐州 221008;

2. 中山大学 机器智能与先进计算教育部重点实验室, 广州 510006)

摘要: 人员作为软件项目调度过程中的核心资源,其学习遗忘特性是无法忽视的. 借鉴已有学习和遗忘模型,构建学习/遗忘效应与人员技能水平之间的动态关联模型,进而给出考虑人员学习/遗忘效应的软件项目调度多目标优化模型. 针对该模型,采用新型调度方案编码方式和不可行解修复方法,给出基于改进 NSGA-II 的软件项目调度多目标优化方法. 面向具有不同项目规模的算例仿真实验表明,考虑人员的学习能力有利于改善调度方案性能,而遗忘效应则会使调度方案的项目总工期和成本增加. 因此,在软件项目调度问题中,考虑人员的学习和遗忘效应是十分必要的.

关键词: 学习效应; 遗忘效应; NSGA-II; 多目标; 软件项目调度

中图分类号: TP273

文献标志码: A

Multi-objective software project scheduling optimization method with the learning and forgetting effect

GUO Yi-nan^{1,2}, JI Jun-hua^{1†}, JI Jian-jiao¹, GONG Dun-wei¹

(1. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China; 2. Key Laboratory of Machine Intelligence and Advanced Computing of Ministry of Education, Sun Yat-sen University, Guangzhou 510006, China)

Abstract: As a core resource in software project scheduling, employees' learning and forgetting characteristic can not be ignored. Based on the existing learning and forgetting models, a model describing the dynamic relationship between the learning/forgetting effect and the employees' skill levels is constructed. Then a multi-objective optimization model for software project scheduling considering above improved learning-forgetting model is given. The multi-objective software project scheduling optimization method based on the improved NSGA-II is introduced to solve this problem by using the new encoding mode and gene-based repair method for the infeasible offsprings. Experimental results under different project scales show that, considering the employees' learning ability is beneficial to improve the performances of the optimal scheduling scheme, whereas the employees' forgetting effect makes the duration and cost of the optimal scheduling scheme worse. Thus it is very necessary to give full consideration to the employees' learning and forgetting effects in the software project scheduling problems.

Keywords: learning effect; forgetting effect; NSGA-II; multi-objective; software project scheduling

0 引言

20 世纪 60 年代,为解决“软件开发与维护费用居高不下、生产率低下、版本升级时间长、软件质量不能满足用户需要”等软件开发危机,越来越多的组织开始对软件项目调度问题的影响因素进行研究^[1]. 其中,软件开发人员的专业素质毫无疑问是影响软件开发质量的重要因素^[2]. 在软件研制过程中,重复性

工作或者技术培训可以提高研发人员开发水平^[3],项目管理者要充分考虑到项目成本的影响. 但是,在已有的软件项目调度文献中,通常认为研发人员的研发技能在整个研发过程中是不变的^[4-5],这是不合理的. 已存在软件公司的开发项目实例^[6]证明,软件项目开发过程中存在学习-遗忘-再学习现象^[7]. 综上所述,在软件项目调度问题中,充分考虑人的学习能力

收稿日期: 2016-12-05; 修回日期: 2017-06-07.

基金项目: 国家自然科学基金项目(61573361); 国家 973 计划项目(2014CB046300); 中山大学机器智能与先进计算教育部重点实验室开放研究基金项目(MSC20170A); 中国矿业大学创新团队项目(2015QN003).

作者简介: 郭一楠(1975-),女,教授,博士生导师,从事智能优化与控制理论及应用等研究; 季俊华(1993-),男,硕士生,从事软件项目调度的研究.

†通讯作者. E-mail: jijunhua.cumt@foxmail.com

和遗忘效应对其工作效率的影响是非常必要的。

生产过程中人员的学习效应最早由 Wright^[8] 提出,通过研究人员工作效率和产品产量之间的关联关系,提出了 WLC(wright's learning curve)学习模型. Smunt 等^[9] 利用学习模型将车间调度问题中的数据进行归纳,更加准确地对中短期生产计划作出决策. 在以外包成本为目标的项目调度问题中, Wu 等^[10] 应用学习模型,预测员工工作能力与工作量的变化关系. 在艾宾浩斯提出的遗忘模型基础上, John 等^[11] 提出了 VRVF(variable regression to variable forgetting)遗忘模型. 它以曲线形式描述了人员的停工时间与遗忘效应的关系. 对学习模型和遗忘模型的合理利用,可以帮助项目管理者制定更符合人实际认知变化过程的调度方案. 但是,在已有软件项目调度问题的研究中,极少考虑人员的学习和遗忘效应.

针对上述问题,本文的核心是研究人的学习/遗忘效应对软件项目研发总工期和总费用的影响. 基于已有 WLC 学习模型和 VRVF 遗忘模型,建立软件研制任务和人员工作效率的关联模型,给出基于动态人员工作效率的软件项目调度问题描述. 针对该多任务、多人员、多技能软件项目调度问题,设计一种调度方案的新型个体编码方式和不可行解的修复策略,并基于此给出一种基于 NSGA-II 算法的软件项目调度优化策略. 面向具有不同项目规模的算例仿真实验表明,考虑人员的学习能力有利于改善调度方案性能,而遗忘效应则会使调度方案的项目总工期和成本增加. 因此,在软件项目调度问题中,考虑人员的学习和遗忘效应是十分必要的.

1 软件开发过程中人员的学习/遗忘效应

传统 WLC 学习模型和 VRVF 遗忘模型,以人员完成单位产量所需要的时间作为学习/遗忘效应的度量,而软件开发是以代码行数作为度量依据. 假设 e_i^k 表示人员 e_i 采用技能 sk_k 在单位时间内可以完成的代码量(千行/周),也称为人员技能值. 根据该人员是否参与任务执行,采用如下改进的学习/遗忘模型确定其技能值变化:

$$e_i^k = \begin{cases} e_i^k |_{tw_{ij}^k} \cdot x^{\alpha_i^k}, & tw_{ij}^k \in Tw_i^k; \\ e_i^k |_{ts_{ij}^k} \cdot y^{-\beta_i^k}, & ts_{ij}^k \in Ts_i^k, \end{cases} \quad (1)$$

其中: tw_{ij}^k 和 ts_{ij}^k 分别为人员 e_i 使用技能 sk_k 完成任务 a_j 的工作时间和未使用该技能的时间; Tw_i^k 和 Ts_i^k 分别为员工 e_i 使用或未使用技能 sk_k 的时间段集合,有 $Tw_i^k \cap Ts_i^k = \emptyset$ 和 $Tw_i^k \cup Ts_i^k = T_d$; $e_i^k |_{tw_{ij}^k}$ 和 $e_i^k |_{ts_{ij}^k}$ 分别为对于任务 a_j 而言,该人员在学习或遗

忘效应下的初始技能值.

人员执行某项任务的质量通过执行时间和误码率度量,误码率 φ 指每千行代码中的错误代码率,有

$$x = e_i^k |_{tw_{ij}^k} \cdot tw_{ij}^k (1 - \varphi), \quad (2)$$

$$y = e_i^k |_{ts_{ij}^k} \cdot ts_{ij}^k (1 - \varphi). \quad (3)$$

本文假设同一人员的不同技能之间具有彼此独立的学习/遗忘效应;相对于同一技能,不同人员的学习/遗忘能力也不同. 无论何种技能,随着人员使用该技能的工作量增加,在学习作用下,其技能值将升高;反之,如果人员在执行任务时未使用该技能,则其技能值将在遗忘效应作用下降低,如图 1 所示. 受到人执行能力的生理限制,在学习或遗忘效应作用后的技能存在学习上限和遗忘下限,分别记为 \bar{e}_i^k 和 \underline{e}_i^k .

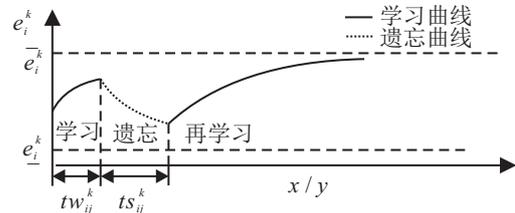


图 1 人员的学习/遗忘特性

2 考虑人员学习/遗忘效应的软件项目调度模型

软件项目调度的目的是协调研发人员各项技能与项目任务之间的匹配关系,保证整个软件开发项目在规定时间内高质量完成.

假设整个软件开发项目包含 $J + 2$ 个任务,任务集合记为 $V = \{a_1, \dots, a_j, \dots, a_{J+2}\}$. 其中,任务 a_1 和任务 a_{J+2} 是虚拟任务,代表项目的开始和结束. 多个任务之间存在时序约束,即对于任务 a_j ,存在一个紧前任务集合 P_{a_j} ,只有当 P_{a_j} 中所有的任务全部完成后, a_j 才可以开始执行,且执行该任务的人员不能在中途退出.

每个任务通常对一个或多个技能存在需求,因此,任务 a_j 的实际工期 d_j 通常为执行该任务的所有员工工作量的最大值,记为 $d_j = \max \lceil a_j^k / e_i^k \rceil$,其中 $\lceil \cdot \rceil$ 表示向上取整. 在任务紧前关系约束下,每个任务的开始时间 st_j 为其紧前任务集 P_{a_j} 中所有任务结束时间的最大值,即 $st_j = \max \{st_{P_{a_j}} + d_{P_{a_j}}\}$.

每个任务的完成依赖于一种或多种技能,对各种技能的需求量往往不同,且不能用其他技能来代替. 虚拟任务的技能需求量为 0,假设整个软件开发项目共需要 K 种技能,则任务 a_j 所需技能集合为 $Sk_j = \{sk_j^1, sk_j^2, \dots, sk_j^k, \dots, sk_j^K\}$. 项目的 I 个可用人员集合为 $R, R = \{e_1, \dots, e_i, \dots, e_I\}$. 根据人员

使用技能值完成任务和参与项目情况,定义软件项目调度问题中的决策变量如下:

$$x_{ijk} = \begin{cases} 1, & \text{员工 } e_i \text{ 在任务 } a_j \text{ 中使用技能 } sk_k; \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

$$y_{it} = \begin{cases} 1, & \text{员工 } e_i \text{ 在 } t \text{ 时刻正在参与项目;} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$$z_i = \begin{cases} 1, & \text{员工 } e_i \text{ 参与了项目;} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

假设 h_i 和 s_i 为员工的工作时长和单位时间的薪资水平. 调度方案的优劣通过项目研制总工期 T_d 和总成本 T_c 加以度量,调度模型的目标函数记为

$$T_d = st_{J+2}, \quad (7)$$

$$T_c = \sum_{i=1}^I h_i \cdot s_i. \quad (8)$$

考虑到项目执行过程中,人员使用技能执行任务的单一性,定义调度模型的约束条件如下:

1) 员工在执行某一任务时,只能使用一种技能

$$\bar{y}_i = \frac{1}{T} \sum_{t=1}^{T_d} y_{it}; \quad (9)$$

2) 任务对某一技能的需求只能由一个员工完成

$$\sum_{i=1}^I x_{ijk} \leq 1; \quad (10)$$

3) 一个员工同一时刻只能参与一项任务

$$\sum_{j=1}^{J+2} (x_{ijk} \cdot y_{it}) \leq 1. \quad (11)$$

3 基于改进NSGA-II的软件项目调度多目标优化方法

NSGA-II是由Deb等^[12]提出的一种基于Pareto最优概念的多目标遗传算法.应用该算法解决软件项目调度问题时,每个个体代表一种调度方案.由一定数量的个体构成一个种群,通过个体评价、非支配排序,以及交叉、变异、选择等进化操作,经过一定迭代次数后得到最优调度方案集.软件项目调度本质上是一个离散的多目标组合优化问题,还需要考虑人员这一类动态资源,且任务和资源种类具有多种复杂的技能需求.因此,解决该类问题的多目标优化策略的核心是:1)合理有效的编码设计;2)不可行解的修复方法.

3.1 调度方案的新型个体编码方式

合理的编码方式是算法有效执行的前提,也是改善算法执行效果的保障.软件项目调度方案通常包

括两部分:任务调度方案 C_1 和人员分配方案 C_2 . 本文对这两部分进行独立编码.对于 C_1 ,采用基于优先权的编码方式^[13]表示调度任务.首先,根据任务的紧前关系,确定各个任务的优先权.根据任务的优先权从高到低选择任务进行人员分配. C_2 采用矩阵Sch形式表示,矩阵中的每个元素为员工编号.例如:对于一个7任务调度问题,根据其任务优先权,确定任务调度编码 C_1 ,如图2(a)所示.任务所需技能总数为4种,一共有8位人员参与该软件项目执行,则一个可行的人员分配方案Sch如图2(b)所示.图中,Sch(5,2)=5表示第5项任务中,对第2种技能的需求由第5位员工来完成.矩阵中0元素表示没有技能需求.

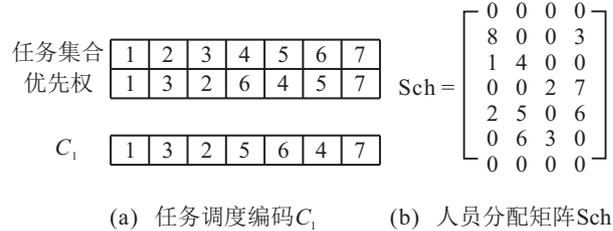


图2 调度方案的构成

为便于进化操作实施,重组人员分配矩阵Sch.由于第1行和最后1行对应的是虚拟任务,不需要人员指派,于是从第2行~第 $J+1$ 行逐行排列,形成展开后的人员分配编码 C_2 .进而与任务调度 C_1 合并,形成新的个体编码 C ,对应一个完整的软件项目调度方案,如图3所示.

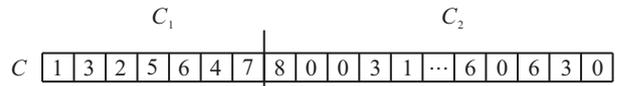


图3 个体编码 C

3.2 基于基因位的不可行个体修复策略

本文进化算子采用经典的单点交叉和按位变异^[14].考虑调度方案编码形式,分别针对父代个体中的 C_1 和 C_2 进行交叉和变异操作.然而,进化操作可能产生不能满足任务紧前关系或模型约束条件的不可行个体.分析表明,调度方案的不可行状况仅与任务调度方案编码 C_1 有关.不可行情况包含:1) C_1 中存在重复调度的子任务,或者缺失的子任务;2) C_1 中存在子任务不符合紧前关系.

假设两个父代个体的任务调度部分记为 $\text{parent}_{c_1}(1)$ 和 $\text{parent}_{c_1}(2)$,经过单点交叉操作后得到子代个体,记为 $\text{offspring}_{c_1}(1)$ 和 $\text{offspring}_{c_1}(2)$.对于子代个体,只有交叉点之后的部分进行交叉互换,可能出现不可行编码.因此,针对子代个体交叉点之后的不可行修复步骤如下,修复过程如图4所示.

Step 1: 判断 $\text{offspring}_{c_1}(1)$ 和 $\text{offspring}_{c_1}(2)$ 中是

否存在重复或缺失的任务,如果存在不可行子任务,则执行 Step2, 否则执行 Step3.

Step 2: 查找 $offspring_{c_1}(1)$ 和 $offspring_{c_1}(2)$ 中交叉点 cp_1 后子任务与交叉点 cp_1 前子任务冲突的基因位置, 互换 $offspring_{c_1}(1)$ 和 $offspring_{c_1}(2)$ 中该基因位置上的子任务.

Step 3: 判断 $offspring_{c_1}(1)$ 和 $offspring_{c_1}(2)$ 是否符合任务间的紧前关系, 若符合, 则执行 Step5, 否则执行 Step4;

Step 4: 判断子代个体中交叉点 cp_1 后每个子任务的紧前任务是否已经完成, 如果存在未完成的紧前任务, 则将该子任务移至其紧前任务之后.

Step 5: 修复完成, 结束.

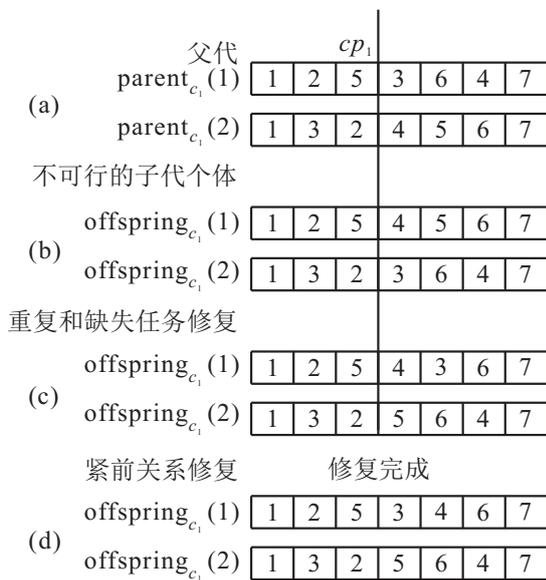


图4 不可行子代个体的修复过程

根据上述改进后 NSGA-II 算法的操作步骤, 结合文献 [15] 已有算法的复杂度分析可知, 本调度优化算法的时间复杂度为 $O(MN^2)$. 其中, M 和 N 分别代表目标数量和种群规模.

4 实验分析与比较

为深入分析软件项目调度优化过程中人员的学习/遗忘效应对项目调度结果和性能的影响, 设计 3 方面实验加以分析: 1) 考虑人员的学习作用和遗忘效应前后, 软件项目调度方案的性能差异; 2) 进化过程中, 在人员的学习/遗忘效应影响下, 其技能值的动态变化过程; 3) 不同的学习因子和遗忘因子对调度优化结果的影响. 实验中人员所具有的技能、薪资水平等属性如表 1 所示. 本文采用改进的 NSGA-II 算法实现项目调度方案寻优, 算法参数选取如下: 种群规模 200, 交叉概率 0.9, 变异概率 0.1, 基因变异概率 0.15, 最大迭代次数 200.

表1 人员属性

人员基本属性								
人员	技能水平				薪资水平	学习因子	遗忘因子	误码率/%
	sk_1	sk_2	sk_3	sk_4				
e_1	2	2	2	0	240	0.29	0.243	6.64
e_2	2	0	3	2	240	0.184	0.199	7.91
e_3	3	0	0	1	240	0.236	0.228	8.24
e_4	1	1	0	2	240	0.264	0.19	4.75
e_5	0	3	1	0	240	0.324	0.22	3.66
e_6	0	5	5	2	360	0.446	0.164	4.83
e_7	0	5	0	4	360	0.358	0.177	1.54
e_8	2	0	5	2	360	0.335	0.193	2.16
e_9	0	3	5	0	360	0.325	0.151	0.35

人员技能上下限								
员工号	技能水平下限				技能水平上限			
	sk_1	sk_2	sk_3	sk_4	sk_1	sk_2	sk_3	sk_4
1	1	1	1	0	4	4	4	0
2	1	0	1.5	1	4	0	4	4
3	1.5	0	0	1	4	0	0	4
4	1	1	0	1	4	4	0	4
5	0	1.5	1	0	0	4	4	0
6	0	2.5	2.5	1	0	3	6	6
7	0	2.5	0	2	0	3	0	6
8	1	0	2.5	1	6	0	6	6
9	0	1.5	2.5	0	0	6	6	0

4.1 人员学习/遗忘效应对调度优化性能的影响

采用 Alba 等 [16] 提出的算例生成器生成 30 个不同规模的算例, 分析人员学习/遗忘效应对项目调度性能的影响. 根据算例的任务数量, 将其均分为 3 组, 每组任务数分别为 10、20、30. 每组中 10 个算例具有不同的任务紧前关系和工作量. 由于算例数量过多, 不再对算例中的具体任务需求和紧前关系列表说明. 每个算例分别在 3 种人员模式下运行 10 次进行实验分析, 记为 S^0 : 不考虑人员的学习能力及遗忘效应; S^L : 仅考虑人员的学习能力; $S^{L\&F}$: 兼顾人员的学习作用和遗忘效应. 对每次运行获得的最优调度方案计算其平均总工期 \bar{T}_d 和平均总成本 \bar{T}_c , 并记录算法的平均运行时间 \bar{t}_r , 如表 2 所示.

根据实验统计值可知, 考虑学习效应和遗忘效应会增加算法的运行时间, 这是因为算法在每次计算个体适应值的过程中需要不断更新人员的技能值, 从而导致计算量小幅度增加. 另外, 随着项目中任务数量的增长, 参与项目完成的人员数量随之增多, 人员的折合工作量变大, 从而使最优调度方案的总成本和总

表2 不同项目规模下3种模式的调度方案性能

任务数	算例	S_0			S_L			$S^{L\&F}$		
		\bar{T}_d	\bar{T}_c	\bar{t}_r	\bar{T}_d	\bar{T}_c	\bar{t}_r	\bar{T}_d	\bar{T}_c	\bar{t}_r
10	1	11.9	12691.2	143.7	11.0	10540.3	151.9	19.2	16679.6	166.9
	2	11.7	11871.9	146.6	11.0	9600.8	150.6	18.2	15294.2	156.2
	3	12.5	11637.4	145.1	10.2	9593.9	149.8	18.2	17013.7	157.9
	4	12.5	13112.1	147.2	11.7	11263.9	153.6	16.8	18729.0	161.1
	5	11.2	11131.7	147.9	10.8	10812.0	148.7	20.1	18552.7	157.9
	6	12.0	12884.5	149.0	11.0	10315.0	153.8	18.0	18182.8	148.2
	7	11.8	12230.6	141.2	10.8	9580.6	151.5	16.9	19271.2	150.7
	8	13.0	11933.6	143.5	12.5	10503.0	154.0	20.1	18250.3	161.0
	9	13.7	11682.1	148.3	13.0	10948.4	157.9	17.8	19392.5	160.6
	10	10.7	9928.3	144.7	10.0	9649.9	149.3	17.2	15072.7	151.5
20	1	18.9	25903.6	246.6	17.5	25810.1	263.2	28.2	37991.3	280.7
	2	18.5	29537.1	246.5	17.9	26725.1	254.2	28.8	40159.8	269.6
	3	19.7	26929.6	252.7	18.1	22777.6	266.4	27.5	43037.4	284.4
	4	22.2	27957.2	247.5	19.9	25119.7	273.5	29.0	40912.3	276.4
	5	22.3	29732.6	244.7	19.4	26689.9	264.8	28.9	39444.4	274.1
	6	22.5	30705.2	248.2	21.0	26640.1	260.9	28.3	42328.1	274.6
	7	22.0	26391.7	247.7	21.4	25005.5	259.3	29.9	41170.2	275.6
	8	21.8	29148.3	243.3	20.2	22998.3	267.7	28.8	46682.2	278.3
	9	20.1	29947.3	240.1	18.9	24946.2	266.5	32.4	41801.9	280.7
	10	23.5	28491.3	248.4	21.6	24986.9	271.0	32.7	43268.0	272.4
30	1	36.8	47860.8	361.4	29.5	41826.2	374.0	46.3	68980.1	401.7
	2	33.7	47984.0	358.6	29.9	43390.7	373.9	43.7	67840.5	390.3
	3	35.4	48942.1	356.3	30.5	40827.9	372.3	49.0	66960.7	394.2
	4	33.9	47614.5	364.9	30.9	45916.2	371.3	46.0	63200.9	388.4
	5	34.5	48354.6	361.3	29.3	44068.7	371.1	41.7	62047.2	392.7
	6	35.0	42238.3	360.3	29.5	37371.4	382.7	43.2	66563.1	398.6
	7	37.8	50420.1	359.4	34.0	44225.2	369.3	48.4	70214.7	397.5
	8	28.2	43982.5	352.2	25.5	38834.9	377.1	44.8	58897.4	392.3
	9	34.7	48060.2	357.6	30.7	42927.8	376.7	45.0	69991.6	397.7
	10	38.9	53329.8	354.3	34.5	44803.5	378.8	49.3	63354.4	397.5

工期都呈上升趋势. 对比相同项目规模下相同算例的调度性能可知, 仅考虑人员的学习作用, 即 S^L 模式下, 项目执行的总工期和成本都小于 S^0 模式下所得的调度方案性能. 这是因为, 在学习作用影响下, 人员的技能水平会随着任务完成量的增加而升高, 从而使其完成相同工作量任务的时间缩短, 相应的总薪酬降低. 在综合考虑学习/遗忘效应, 即 $S^{L\&F}$ 模式下, \bar{T}_d 和 \bar{T}_c 均劣于 S^L 模式下的调度方案性能. 分析表明, 考虑遗忘效应后, 人员的技能水平会在未使用该技能的任务执行时间内发生衰退, 从而导致完成同质任务的总工期延长, 相应的薪酬成本增加. 综上所述, 人员的学习能力和遗忘效应会对调度方案性能产生显著的影响. 人员的学习能力有利于调度方案性能的提升, 而遗忘效应则会使其性能变差, 二者都是

人的固有特性, 均不可忽略其影响.

4.2 人员学习/遗忘效应对进化过程的影响分析

考虑具有30个任务(总任务数量为32, 包含首尾两个空任务)的生成算例, 任务需要4种技能, 如表3所示. 基于表3所示的人员属性, 分别在 S^0 、 S^L 、 $S^{L\&F}$ 三种模式下, 分析人员学习/遗忘效应对进化过程的影响, 获得的最优调度方案如图5所示. 图5中, 每个点分别代表一种调度方案. 每个调度方案的项目总工期和总成本如表4所示. 可见, 在不同人员模式下, 最优调度方案的性能差异较大. 特别是在 $S^{L\&F}$ 模式下, 最优调度方案的项目总成本最高, 工期最长. 因此, 在解决实际项目调度问题时, 如果忽略人员的学习/遗忘效应, 则会导致项目成本与工期预估不准确, 甚至导致项目研制失败.

表3 任务属性表

任务	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
紧前任务	无	1	1	1	3	5	5	3	5	2	2,9
任 务	sk_1	0	0	0	0	2	5	0	0	0	2
量	sk_2	0	5	8	10	6	0	0	0	1	0
	sk_3	0	7	5	0	10	0	4	7	10	0
	sk_4	0	7	4	8	2	10	0	0	0	8

任务	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}
紧前任务	3,10	7	6	2	7,15	9,15	10	4,13,16	6,18	14,18	8,13,17
任 务	sk_1	2	0	9	0	0	8	10	0	0	0
量	sk_2	0	7	0	7	5	10	2	6	0	2
	sk_3	0	5	0	0	0	0	0	0	4	6
	sk_4	0	0	0	1	0	7	0	6	8	0

任务	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}	a_{30}	a_{31}	a_{32}
紧前任务	11,16,22	13,20	15,20,21	9,12,24	14,26	6,17	16,17,27	7,25,28	19,23,27,	29,30,31
任 务	sk_1	5	7	0	0	9	0	0	7	9
量	sk_2	3	1	6	0	0	8	0	10	0
	sk_3	8	8	7	0	1	9	3	9	5
	sk_4	0	0	6	10	0	1	0	1	5

表4 不同人员模式下最优调度方案性能

模型	目标	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}
S^0	T_d	30	32	34	35	36	38	40	43
	T_c	50 520	49 200	47 880	47 760	47 280	46 440	45 600	44 160

模型	目标	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	—	—	—
S^L	T_d	26	28	31	32	33	—	—	—
	T_c	42 720	42 100	41 160	40 560	40 020	—	—	—

模型	目标	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}	—
$S^{L\&F}$	T_d	39	41	42	44	45	46	47	—
	T_c	66 120	65 820	65 410	65 160	64 920	64 280	63 720	—

表5 方案 p_{23} 和方案 p_{35} 项目完成后的人员技能值

员工	p_{23}				p_{35}			
	sk_1	sk_2	sk_3	sk_4	sk_1	sk_2	sk_3	sk_4
e_1	2.4	3.13	4	0	2.45	3.45	2.27	0
e_2	2.21	0	3.89	2.97	<u>1.45</u>	0	<u>1.96</u>	<u>1.26</u>
e_3	4	0	0	1	3.46	0	0	1.51
e_4	1	1.59	0	4	1.14	1.64	0	<u>1.62</u>
e_5	0	4	2.71	0	0	4	1	0
e_6	0	5	6	4.56	0	6	6	2
e_7	0	6	0	5.94	0	5.64	0	6
e_8	6	0	6	6	4.61	0	<u>2.57</u>	1
e_9	0	6	6	0	0	3.67	<u>4.41</u>	0

受到学习/遗忘效应的影响,人员的技能水平会随着任务的执行而发生动态变化.分别在 S^L 和 $S^{L\&F}$ 模式下,选取一个最优调度方案 p_{23} 和 p_{35} ,其技能值变化如表5所示.表5中,加粗项表示人员的该项技能值提升,下划线项表示人员的技能值下降.由于 p_{23} 调度方案仅考虑了人员的学习能力,人员的技能水平随着任务的执行逐步上升,甚至接近其上限极值.与之不同, p_{35} 调度方案兼顾了人员的学习和遗忘效应,尽管部分人员的技能水平得到了增长,但是增

长幅度明显小于 p_{23} ,而且部分人员的技能水平存在下降,从而导致项目的工期和成本大幅上升.

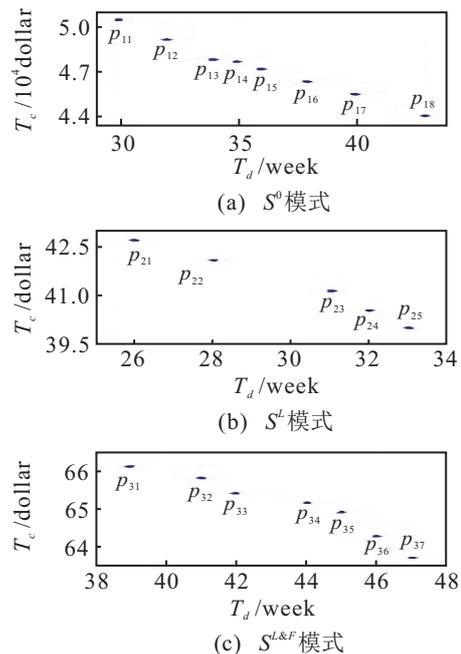


图5 不同人员模式下的最优调度方案解集

进一步对比如图6所示的 p_{16} 、 p_{23} 和 p_{35} 调度方案的甘特图可知:由于 p_{16} 采用技能水平固定的 S^0 人员模式,同一人员采用相同技能执行任务时,任务完成时间不变。 p_{23} 的项目执行前期,与 p_{16} 对同一任务的执行时间相近,但是在项目执行后期,随着人员技能水平的提升, p_{23} 执行同一任务的工期明显少于前者。这是因为,人员的技能水平会随着执行任务工作量的积累而提升,人员的学习效应对项目后期任务的

执行时间有更加显著的影响,从而使项目的总工期缩短。观察 p_{35} 的甘特图发现,项目执行的中后期存在一些耗时较长的任务。通过观察其人员分配方案,发现在项目执行过程中,有些任务存在等待状态。此时,虽然这些任务的紧前任务均已完成,但是由于具有该任务所需技能的人员均在工作中,导致任务停滞。人员的执行任务时间增加,一定程度上是受到了遗忘效应的影响,使较少使用的技能具有较低的掌握水平。

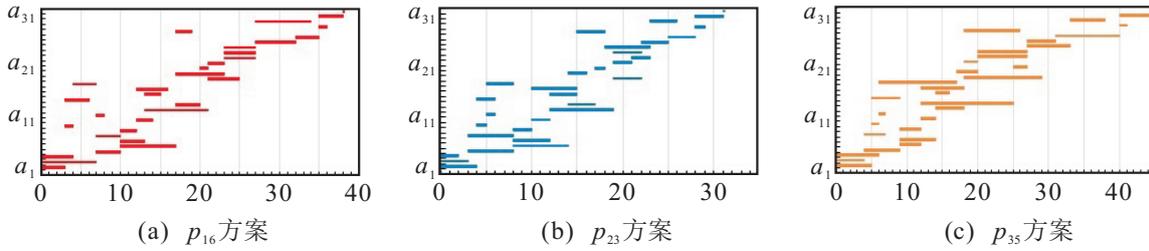


图6 甘特图

表6 学习因子 α 和遗忘因子 β 的取值

员工		e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
normal	α	0.29	0.184	0.236	0.264	0.324	0.446	0.358	0.335	0.325
	β	0.243	0.199	0.228	0.19	0.22	0.164	0.177	0.193	0.151
low α	α	0.145	0.092	0.118	0.132	0.162	0.223	0.179	0.168	0.163
	β	0.243	0.199	0.228	0.19	0.22	0.164	0.177	0.193	0.151
high α	α	0.58	0.368	0.472	0.528	0.648	0.892	0.716	0.67	0.65
	β	0.243	0.199	0.228	0.19	0.22	0.164	0.177	0.193	0.151
low β	α	0.29	0.184	0.236	0.264	0.324	0.446	0.358	0.335	0.325
	β	0.122	0.1	0.144	0.095	0.11	0.082	0.088	0.096	0.076
high β	α	0.29	0.184	0.236	0.264	0.324	0.446	0.358	0.335	0.325
	β	0.486	0.398	0.455	0.381	0.44	0.329	0.353	0.385	0.302

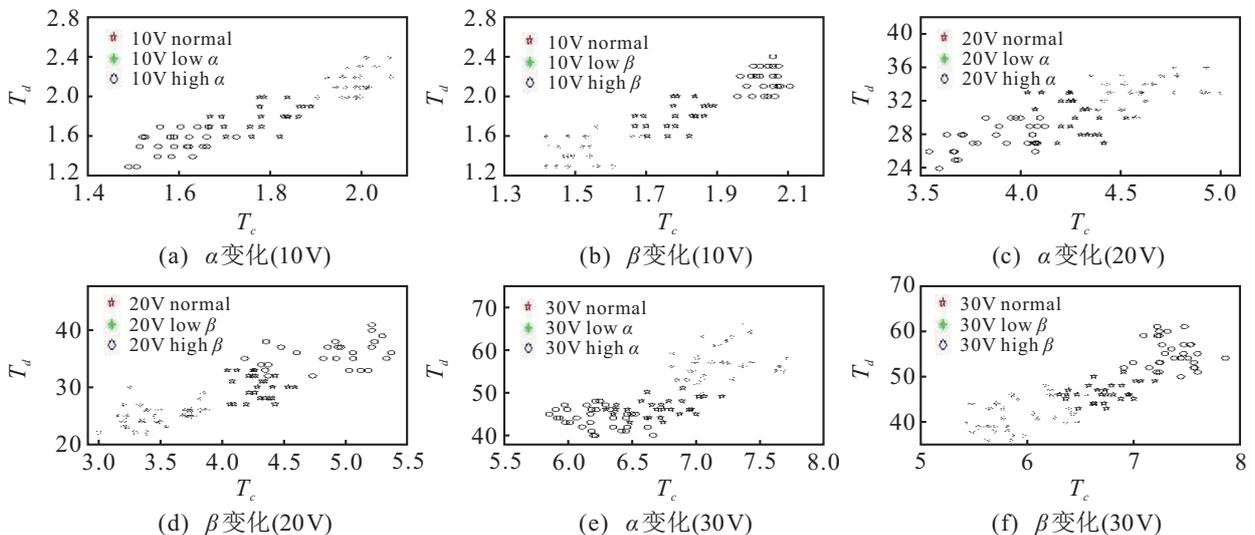


图7 不同学习/遗忘因子下的项目调度性能比较

4.3 学习因子和遗忘因子对调度性能的影响

针对上述30个算例,采用5组不同的人员学习因子 α 和遗忘因子 β 分析其对项目调度结果的影响.5组学习/遗忘因子分别记为normal, low α , high α , low β 和high β ,具体取值如表6所示.图7为相应的调度方案性能,10V、20V、30V分别表示算例的项目规模为10、20和30.可知,无论项目规模大小,项目总工期 T_d 和总成本 T_c 均与学习因子 α 成反比,与遗忘因子 β 成正比.这是因为,学习因子 α 越大,人员的学习能力越强;在执行相同工作量的任务后,人员的技能水平提升越多,单位时间内可以完成的任务量越显著增加,从而使项目总工期和成本减少.同理,遗忘因子 β 越大,人员在未使用该技能时,技能值下降得越快,单位时间内可以完成的任务量越少,导致项目总工期和成本上升.

5 结论

已有较多证据证明,人员固有的学习能力和遗忘效应会显著影响项目调度优化性能,但是在软件项目调度方面的相关研究较少.本文借鉴已有学习和遗忘模型,构建了基于学习和遗忘效应的人员技能动态模型,进而给出了软件项目调度优化模型.针对该模型,采用新型调度方案编码方式,给出了基于改进NSGA-II的软件项目调度多目标优化方法.面向不同项目规模算例的仿真实验表明,考虑人员的学习能力有利于改善调度方案性能,而遗忘效应则会使调度方案的项目总工期和成本增加.尽管本文考虑了学习/遗忘效应对人员技能值的影响,但是调度方案中没有充分考虑在动态技能水平影响下,如何平衡整体人力资源,这是下一步的研究工作.

参考文献(References)

- [1] Rekha J H, Parvathi R. Survey on software project risks and big data analytics[J]. Procedia Computer Science, 2015, 50: 295-300.
- [2] Sarigiannidis L, Chatzoglou P D. Quality vs risk: An investigation of their relationship in software development projects[J]. Int J of Project Management, 2014, 32(6): 1073-1082.
- [3] 郭研.资源受限的软件项目群调度问题研究[D].南京:南京航空航天大学经济与管理学院,2013.
(Guo Y. Research on the resource-constrained software programme scheduling problem[D]. Nanjing: College of Economics and Management, Nanjing University of Aeronautics and Astronautics, 2013.)
- [4] Xiao J, Ao X T, Tang Y. Solving software project scheduling problems with ant colony optimization[J]. Computers & Operations Research, 2013, 40(1): 33-46.
- [5] 句彩霞.考虑知识员工软技能的软件多项目调度问题研究[D].西安:西安电子科技大学经济与管理学院,2014.
(Ju C X. A study of software multi-projects scheduling problem considering soft skills of knowledge workers[D]. Xi'an: College of Economics and Management, Xidian University, 2014.)
- [6] Qin S, Liu S, Kuang H. Piecewise linear model for multiskilled workforce scheduling problems considering learning effect and project quality[J]. Mathematical Problems in Engineering, 2016, 4: 1-11.
- [7] 范增辉,方伟.改进遗传算法在软件项目调度中的研究[J].计算机应用与软件,2016,33(12):34-39.
(Fan Z H, Fang W. Research on improved genetic algorithm for software project scheduling[J]. Computer Applications and Software, 2016, 33(12): 34-39.)
- [8] Wright. Factors affecting the cost of airplanes[J]. J of the Aeronautical Sciences, 1936, 3(4): 122-128.
- [9] Smunt T L, Watts C A. Improving operations planning with learning curves: overcoming the pitfalls of 'messy' shop floor data[J]. J of Operations Management, 2003, 21(1): 93-107.
- [10] Wu M C, Sun S H. A project scheduling and staff assignment model considering learning effect[J]. Int J of Advanced Manufacturing Technology, 2006, 28(11): 1190-1195.
- [11] John G Carlson, Alan J Rowe. How much does forgetting cost?[J]. Industrial Engineering, 1976, 8(9): 40-47.
- [12] Deb K, Agrawal S, Pratap A, et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II[C]. Parallel Problem Solving From Nature Vi Conf. Springer, 2000: 849-858.
- [13] Jia Q, Seo Y. An improved particle swarm optimization for the resource-constrained project scheduling problem[J]. Int J of Advanced Manufacturing Technology, 2013, 67(9): 2627-2638.
- [14] 罗辞勇,陈民铀,张聪誉.采用循环拥挤排序策略的改进NSGA-II算法[J].控制与决策,2010,25(2):227-231.
(Luo C Y, Chen M Y, Zhang C Y. Improved NSGA-II algorithm with circular crowded sorting[J]. Control and Decision, 2010, 25(2): 227-231.)
- [15] 陈志旺,陈林,白铨,等.求解约束多目标区间优化的交互多属性决策NSGA-II算法[J].控制与决策,2015,30(5):865-870.
(Chen Z W, Chen L, Bai X, et al. Interactive multi-attribute decision-making NSGA-II for constrained multi-objective optimization with interval numbers[J]. Control and Decision, 2015, 30(5): 865-870.)
- [16] Alba E, Chicano J F. Software project management with GAs[J]. Information Sciences, 2007, 177(11): 2380-2401.

(责任编辑:郑晓蕾)