

资源受限多项目调度问题的两阶段算法

陈俊杰^{1†}, 同淑荣¹, 叶正梗², 张静文¹, 王 曜³, 聂亚菲¹, 张雨芊¹

(1. 西北工业大学 管理学院, 西安 710072; 2. 西北工业大学 机电学院, 西安 710072;
3. 中山大学 管理学院, 广州 510275)

摘 要: 在资源受限项目调度问题中, 将可更新资源进一步拓展为具有胜任力差异的人力资源, 建立考虑胜任力差异的人力资源受限多目标项目调度问题模型. 该模型是对传统多模式资源约束项目调度问题更接近研发项目群实际的扩展. 针对模型提出两阶段优化算法, 第1阶段是项目时序约束优化阶段, 采用蚁群算法(ACO)进行任务列表的优化求解, 通过对信息素增量规则的改进、串联进度生成机制(SSGS)及资源冲突消解策略的使用, 使蚁群算法的求解效率和质量得以提高; 第2阶段是资源约束优化阶段, 以第1阶段求得的优化任务列表为输入, 逐项对人力资源约束进行核查与调整, 最终生成项目调度的优化方案. 数值实验表明, 考虑胜任力差异的数学优化模型更符合研发项目群管理实践, 同时两阶段算法在求解质量方面具有良好性能.

关键词: 胜任力; 研发项目群; 项目调度; 蚁群算法; 串行调度生成机制; 冲突消解

中图分类号: TP273

文献标志码: A

Two-stage algorithm for resource-constrained multi-project scheduling problem

CHEN Jun-jie^{1†}, TONG Shu-rong¹, YE Zheng-geng², ZHANG Jing-wen¹, WANG Yao³, NIE Ya-fei¹, ZHANG Yu-qian¹

(1. School of Management, Northwestern Polytechnical University, Xi'an 710072, China; 2. School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China; 3. School of Management, Sun yat-Sen University, Guangzhou 510275, China)

Abstract: In resource-constrained project scheduling problem, renewable resource can be expanded into human resource with competency differences, and a flexible resource-constrained project scheduling problem with competency difference is proposed, which is a practical extension close to research and development (R&D) program from the traditional multi-mode resource-constrained project scheduling problem. In order to address the problem, a two-stage algorithm is proposed. In the first stage of precedence constraints satisfactory optimization, a revised ant colony optimization (ACO) algorithm is presented to obtain the feasible activity sequence. In order to accelerate the convergence efficiency and quality, a revised pheromone increment updating operator of ACO with the combination of the serial schedule generation scheme (SSGS) and the resource conflict resolution strategy are used. At the second stage of resource-constraints satisfactory optimization, the obtained optimum precedence activity sequence is taken as the input, and the resource capacity is examined and adjusted one by one until the optimal scheduling solution is obtained. Experimental results show that the optimization model considering the difference of competency is more suitable for the R&D program management practice, and two-stage algorithm can solve the model effectively.

Keywords: competence; R&D program; project scheduling; ant colony optimization; serial schedule generation scheme; conflict resolution

0 引 言

当高技术企业飞速发展时, 大量相似的项目不断涌现, 企业需要制定战略计划来满足长期发展的需要, 一种有效的多项目管理方法(即项目群管理)应运

而生. 高技术企业多个研发项目并行开展, 使得企业资源配置管理面临更大的挑战. 不合理的资源结构、不均衡的资源配置以及组织缺乏系统协调的能力会导致时而资源过剩、时而资源短缺的现象. Elonen

收稿日期: 2018-11-08; 修回日期: 2019-03-02.

基金项目: 国家自然科学基金项目(71572148); 航空科学基金项目(2015ZG53080); 中国博士后科学基金项目(2015M580875).

责任编辑: 刘士新.

[†]通讯作者. E-mail: junjiechen75@nwpu.edu.cn.

等^[1]从项目群管理的视角,调查了企业存在的问题,得出了影响项目群管理的各因素重要性排序,其中位列第2的是“资源短缺及资源不合理配置”。因此,如何协调与优化多个项目之间的资源配置,是企业迫切需要解决的问题。企业技术改造类研发项目群致力于通过研究新的工艺技术实现降低产品成本、提升产能、提高质量水平以及缩短研发项目周期的目标。此类研发项目群是一个充满风险的动态创新过程,它成功的一项关键因素是需要投入大量复杂脑力劳动的研发人员,因此研究如何优化配置稀缺的研发人员,成为高技术企业研发项目群亟需解决的核心问题。

企业研发项目群人员优化配置问题属于资源约束型项目调度问题,它是当前国际上项目管理领域研究的聚焦点,以往研究项目调度问题很少区分不可更新资源与可更新资源的不同,也很少从人力资源具有柔性属性的角度研究项目调度问题。事实上,人力资源约束下的项目调度问题属于柔性资源受限项目调度问题^[2](flexible resource-constrained project scheduling problem, FRCPSP),即鉴于人力资源是具备多种能力属性的柔性资源,研究考虑柔性资源配置方案的资源受限项目调度问题。

鉴于柔性资源受限项目调度问题在学术界及企业研发项目群运用中都具有重要作用,国内外许多学者开始关注并研究该问题。王一帆等^[3]对多技能人力资源约束的项目调度问题提出了两阶段算法;任逸飞等^[4]以最小化项目总工期为目标,考虑作业执行时间因资源具备的技能水平而变,利用双层决策及局部优化策略的混合算法求解;Wang等^[5]提出了用知识引导的多目标果蝇优化算法解决多技能项目调度问题;邓富民等^[6]考虑手术台、执刀医师等资源约束,构建了多个目标函数的模糊调度模型,设计了改进的非支配排序遗传算法;Chen等^[7]研究了基于事件调度的软件项目调度和人员配置的蚁群算法;Zheng等^[8]提出了一种基于教学的优化算法,以工期最小化为目标的资源约束型项目调度问题。上述文献大多研究了最小化项目工期问题^[3-4,8]和最小化项目成本问题^[7],少数文献研究了多目标调度优化问题^[5-6]。

FRCPSP的提出者Tritschler等^[9]提出了一个新术语——柔性资源配置,仅研究了柔性资源的配置数量与活动工期的关系。传统的多模式资源约束项目调度问题(multi-mode resource-constrained project scheduling problem, MRCPSP)也仅研究了柔性资源的配置数量对活动工期的影响,未充分考虑人力资源

具有多种能力的柔性属性。因此FRCPSP和MRCPSP都未充分研究柔性资源的配置方案对活动工期或成本的影响。

以往研究认为,研发人员的工作绩效主要依赖于员工的能力和技能,所以学者们主要研究人力资源工作效率或能力差异对活动实际执行时间的影响。Heimerl等^[10]基于人力资源的工作效率与活动工期之间的关系,建立了多项目背景下的数学模型并求解,但模型中未解决每个员工确切工作量的估计;Chen等^[11]研究了人员工作效率与活动时间的关系,并利用工时定额量化活动工作量。文献[3]假设资源完全具有或者完全不具有某种技能,文献[4-5,8]通过技能级别描述资源的能力差异,对资源的能力差异进行建模。然而,近些年研究发现,研发人员的技能对能否提升个人绩效没有直接预测效果,研发人员的胜任力才是决定绩效差异的一个决定性因素^[12],即胜任力水平高的研发人员作业用时短、质量高、研发耗费的物料成本低。

基于前人研究成果,本文考虑到企业研发项目群中人力资源胜任力对活动工期和研发活动消耗的物料成本均有显著影响这一事实,提出衡量人员胜任力的参数,建立问题的双目标数学优化模型。此研究问题是对传统MRCPSP的拓展,称作特殊的多模式资源约束项目调度问题(special multi-mode resource-constrained project scheduling problem, S-MRCPSP)。它是人力资源配置与项目调度相结合的新问题,其问题解空间和求解难度都更大。根据问题特征,结合蚁群算法、串行调度生成机制和资源冲突消解策略,设计两阶段优化算法。数值实验表明,求解考虑胜任力差异的项目调度问题的两阶段算法具有良好性能。

1 模型构建

1.1 问题界定

本文主要研究为实现以最省时最经济的方式完成多个并行项目,如何确定项目进度计划及如何确定满足多目标的人力资源问题。具体问题特征界定如下:企业内部有并行研发项目群,人力资源库中人员有待分配任务,各项目中任务的紧前紧后关系已知,各任务的物料限量及各任务的分配人数已知,各任务的工时定额已知,人员的物料消耗率已知。问题是如何确定多项目活动的执行模式和开始时间,以实现多项目总工期和总费用最小化目标。

为了问题的简化,做出如下假设:1) 执行任务的人员组合都是同时开始同时结束,不允许任务未完成

人员中途撤离;2) 所有任务消耗物料的种类(除人力资源外)相同;3) 人员在不同任务中对同一种物料的消耗率不同;4) 多项目总费用只考虑研发消耗的物料成本;5) 各人员执行任务的工期、成本都具有明显差异;6) 各人员执行任务的质量无明显差异,均能达到企业质量要求。

1.2 问题模型的构建

构建此问题的混合整数规划模型,其中的符号定义如下所示:记 a_{jk} 为项目 j 的活动 k ; P_{jk} 为项目 j 的活动 k 的紧前活动集合; NP 为项目的数量; N_j 为项目 j 的活动总数; NR 为员工的数量; T 为多项目总工期的上限; T_{jk} 为项目 j 的活动 k 的工时定额; N_{jk} 为设定的分配给项目 j 中活动 k 的员工数; N_i 为设定的各时段可分配的员工数上限; K_{ijk} 为员工 i 完成项目 j 的活动 k 的工时系数^[13]; Z 为物料的种类数; p_z 为物料 z 的单价; L_{jkz} 为项目 j 的活动 k 的物料 z 的限量; Φ_{ijkz} 为项目 j 的活动 k 中员工 i 对物料 z 的消耗率; Φ_{jkmz} 为项目 j 的活动 k 在模式 m 的员工组合下物料 z 消耗率总和; d_{jkm} 为项目 j 的活动 k 采用模式 m 时的活动工期; $H(j, k)$ 为项目 j 的活动 k 的可行执行模式集合; S_{ijkm} 为如果项目 j 的活动 k 的执行模式 m 中包含员工 i ,则取值为1,否则取值为0; $y_{jkm t}$ 为0 ~ 1的决策变量,若项目 j 的活动 k 采用模式 m 且开始于 t 时刻,则取值为1,否则取值为0。

1) $T, \Phi_{jkmz}, d_{jkm}, H(j, k)$ 参数值的计算。

① 多项目总工期上限。

$$T = \sum_{j=1}^{NP} \sum_{k=1}^{N_j} T_{jk}. \quad (1)$$

② 活动采用模式 m 时的物料消耗率。

$$\Phi_{jkmz} = \sum_{i=1}^{NR} \Phi_{ijkz} S_{ijkm}, \quad \forall j = 1, 2, \dots, NP, \\ k = 1, 2, \dots, N_j, \quad z = 1, 2, \dots, Z, \quad m \in H(j, k). \quad (2)$$

③ 活动采用模式 m 时的活动工期。

$$\sum_{i=1}^{NR} d_{jkm} K_{ijk} S_{ijkm} \geq N_{jk} T_{jk}, \quad \forall j = 1, 2, \dots, NP, \\ k = 1, 2, \dots, N_j, \quad m \in H(j, k). \quad (3)$$

$$d_{jkm} \Phi_{jkmz} \leq L_{jkz}, \quad \forall j = 1, 2, \dots, NP,$$

$$k = 1, 2, \dots, N_j, \quad z = 1, 2, \dots, Z, \quad m \in H(j, k). \quad (4)$$

式(3)表示各任务人员组合完成的总工作量不能小于活动工作量,需要说明的是,求得的实际工时数通常是非整数,为了确保完成活动,一般向上取整。式

(4)表示人员消耗的物料不能超过活动的限制量,一般向下取整。 d_{jkm} 取值为同时满足式(3)和(4)的最小整数。

④ 活动可行执行模式集合的构成。

下式表示规定分配给项目 j 活动 k 的员工数:

$$\sum_{i=1}^{NR} \sum_{m \in H(j, k)} \sum_{t'=\max(0, t-d_{jkm}+1)}^t y_{jkm t'} S_{ijkm} = N_{jk}, \\ \forall j = 1, 2, \dots, NP, \quad k = 1, 2, \dots, N_j. \quad (5)$$

$H(j, k)$ 由同时满足式(3)~(5)的执行模式组成。

2) 目标函数。

$$f_1 = \text{minimize } D, \quad \sum_{t=0}^T \sum_{m \in H(j, k)} (t + d_{jkm}) y_{jkm t} \leq D, \\ \forall j = 1, 2, \dots, NP, \quad k = 1, 2, \dots, N_j. \quad (6)$$

$$f_2 = \min \sum_{z=1}^Z \sum_{t=0}^T \sum_{j=1}^{NP} \sum_{k=1}^{N_j} \sum_{m \in H(j, k)} d_{jkm} y_{jkm t} \Phi_{jkmz} p_z. \quad (7)$$

3) 约束条件。

$$\sum_{j=1}^{NP} \sum_{k=1}^{N_j} \sum_{m \in H(j, k)} \sum_{t'=\max(0, t-d_{jkm}+1)}^t y_{jkm t'} S_{ijkm} \leq 1, \\ \forall i = 1, 2, \dots, NR, \quad t = 0, 2, \dots, T. \quad (8)$$

$$\sum_{t=0}^T \sum_{m \in H(j, k)} y_{jkm t} = 1, \\ \forall j = 1, 2, \dots, NP, \quad k = 1, 2, \dots, N_j. \quad (9)$$

$$\sum_{i=1}^{NR} \sum_{j=1}^{NP} \sum_{k=1}^{N_j} \sum_{m \in H(j, k)} \sum_{t'=\max(0, t-d_{jkm}+1)}^t y_{jkm t'} S_{ijkm} \leq N_i, \quad \forall t = 0, 1, \dots, T. \quad (10)$$

$$\sum_{t=0}^T \sum_{m \in H(j, k)} t y_{jkm t} \geq \sum_{t=0}^T \sum_{m \in H(j, k)} y_{jkm t} (t + d_{jkm}), \\ \forall a_{jk} \in P_{jv}, \quad j = 1, 2, \dots, NP. \quad (11)$$

$$y_{jkm t} \in \{0, 1\}, \quad \forall j = 1, 2, \dots, NP, \quad k = 1, 2, \dots, N_j, \\ t = 0, 1, \dots, T, \quad m \in H(j, k). \quad (12)$$

式(6)表示首要目标是多项目总工期极小化,多项目总工期是指多个并行项目中最早启动的活动的开始时间到最晚结束的活动的完成时间;式(7)表示次要目标是多项目总成本极小化,多项目总成本是指多个项目耗费的研发物料成本之和;式(8)表示在同一个时间段一项活动至多分配一个人员;式(9)表示

每项活动仅能选择一种模式并且仅在某一个时刻开始;式(10)规定同一时段可分配的员工数量上限;式(11)表示活动之间要满足时序关系,即某项活动的结束时间要小于其所有紧后活动的开始时间;式(12)限定了模型的决策变量取值范围。

2 算法设计

2.1 问题复杂性解析

本文研究的特殊的多模式资源受限项目调度问题(S-MRCPSP)的活动执行时间可变,相当于活动有“多个执行模式”,因此读者很可能将本文问题理解为传统的MRCPSP. S-MRCPSP与传统MRCPSP的主要区别在文献[14]中已进行了详细解析。

S-MRCPSP引入带有胜任力水平差异的人力资源,使得传统MRCPSP的决策因素和约束数量增多,从而导致问题模型和可行解分布情况更加复杂、可行解数量急剧增加. 因此需要设计有效算法优化求解。

2.2 两阶段优化算法的构建

该S-MRCPSP的解由活动的开始时间和活动的人员配置两部分组成,考虑到问题的模型包含活动的时序关系约束和资源能力约束,活动的时序关系约束是指活动之间紧前或紧后的逻辑关系约束;资源能力约束是指可更新及不可更新资源的约束条件限制. 本文的求解思路是分解成两个阶段依次解决,第1阶段是第2阶段求解的基础。

第1阶段为时序关系优化阶段,此阶段的输出是满足时序关系约束的任务链表. 在生成优化任务链表的过程中,可采用资源冲突消解策略应对并联活动争夺资源的问题. 第2阶段为人力资源能力优化阶段,在得到满足时序关系任务链表的基础上,从现有的资源出发,逐项核查并调整任务链表中各活动的资源约束,从而得到满足时序约束和资源约束的调度方案. 下面将对这两个阶段的设计流程进行详细说明。

2.2.1 时序关系优化阶段

本阶段根据标准定额下的活动工时定额、人员限量,采用蚁群算法,以各作业距网络终点的最大距离为参数构造启发函数,以项目总工期最短为目标,对项目时序约束进行优化. 使用蚂蚁经过总路径的持续时间构造信息素增量,可克服蚁群算法容易陷入局部最优的缺点. 为了提高蚁群算法的求解质量及效率,采用串行调度生成机制. 在生成优化任务链表的过程中,可采用资源冲突消解策略解决并联活动的资源冲突,直至所有资源的冲突都被消除,最终得到

标准定额条件下的最优调度方案。

1) 蚁群算法设计.

① 蚁群移动规则的建立.

在搜索的过程中,蚂蚁根据各条路径上的信息量及路径的启发信息计算状态转移概率. 蚁群移动规则为

$$p_{jj'}^i(t) = \begin{cases} \frac{[\tau_{jj'}(t)]^\alpha \times [\eta_{jj'}(t)]^\beta}{\sum_{j \in \zeta_i} [\tau_{jj'}(t)]^\alpha \times [\eta_{jj'}(t)]^\beta}, & j \in \zeta_i; \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

其中: $p_{jj'}^i(t)$ 表示在 t 时刻蚂蚁 i 由活动 j 转移到活动 j' 的状态转移概率; $\zeta_i = \{0, 1, \dots, n-1\} - \text{tabu}_i$ 表示当前时刻蚂蚁 i 的可排活动集合, n 表示节点总数,蚂蚁 i 的禁忌表 tabu_i 用来记录蚂蚁 i 已经走过的节点集合; α 表示信息启发式因子,反映了蚂蚁在移动过程中积累的信息在蚂蚁移动时所起的作用; β 表示期望启发式因子,反映了蚂蚁在移动过程中启发信息在蚂蚁选择路径中的受重视程度; $\tau_{jj'}(t)$ 表示 t 时刻残留在路径 (j, j') 上的信息素浓度, $\tau_{jj'}(0)$ 表示初始时刻残留在路径 (j, j') 上的信息素浓度; $\eta_{jj'}(t)$ 表示蚂蚁由节点 j 转移到节点 j' 的启发函数。

$$\eta_{jj'}(t) = \text{distance_max}(j'), \quad (14)$$

其中 $\text{distance_max}(j')$ 表示待选择活动 j' 距离网络终点的最大工期。

② 信息素更新规则的建立.

为了避免残留信息素过多引起残留信息淹没启发信息,在每只蚂蚁遍历所有节点之后需要更新残留信息. 信息素更新规则为

$$\tau_{jj'}(t+1) = (1-\rho) \times \tau_{jj'}(t) + \Delta\tau_{jj'}(t), \quad \rho \in (0, 1),$$

$$\Delta\tau_{jj'}(t) = \sum_{i=1}^m \Delta\tau_{jj'}^i(t). \quad (15)$$

其中: ρ 表示挥发因子, $1-\rho$ 表示路径 (j, j') 上的信息素经过一段时间后的残留因子; $\Delta\tau_{jj'}^i(t)$ 表示第 i 只蚂蚁在本次循环中留在路径 (j, j') 上的信息量, $\Delta\tau_{jj'}^i(0) = 0$ 表示初始时刻的信息素增量; $\Delta\tau_{jj'}(t)$ 表示本轮循环中蚁群在路径 (j, j') 上的信息素增量。

本文采用Ant-Cycle模型计算蚂蚁 i 释放的信息素增量

$$\Delta\tau_{jj'}^i(t) = \begin{cases} \frac{Q}{\text{time}(i)}, & \text{蚂蚁 } i \text{ 从活动 } j \text{ 访问活动 } j'; \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

其中: Q 为正常数, 表示信息素强度; $\text{time}(i)$ 表示蚂蚁 i 在本次循环中所走路径的总工期。

2) 串联进度生成机制。

进度生成机制 (schedule generation scheme, SGS)^[13] 对于每个基于优先级的启发式算法都不可或缺。基于任何优先规则并采用 SSGS 生成的单模式资源受限项目调度问题 (single-mode resource-constrained project scheduling problem, SRCPSP) 的调度计划均为积极调度计划, 它既缩小了搜索范围, 又不会错过最优解^[15]。当各个活动的执行模式被指定, 并且指定的模式满足不可更新资源约束时, MRCPSP 就退化为 SRCPSP, 因此, MRCPSP 也满足 SRCPSP 的性质和定理^[15]。鉴于此, 本文采用 SSGS。

3) 资源冲突消解策略。

在资源受限项目调度问题中资源冲突是一个棘手的关键性问题, 此问题在多项项目调度中更加突出。资源冲突的解决方法通常有延迟非关键活动方法、降低非关键活动强度方法和对活动进行可间断执行的3种处理方法^[16]。针对并联项目中出现活动争夺资源的情况, 本阶段采用如下策略对资源冲突进行消解。如果同一时段人员使用量超过人员使用量的上限, 则基于本文假设, 人员在任务完成之前无法撤离, 因而需等待人员释放后再开始待排活动的排产。

4) 构建时序优化算法。

将串联调度生成机制和资源冲突消解策略融合在蚁群算法中, 构建寻找最优排列方案的蚁群算法 (searching optimal scheduling scheme by ant colony algorithm, SOSS_ACA), 具体步骤如下。

step 1: 参数初始化。活动开始时间 $t = 0$, 最大迭代次数 $\text{iter_max} = 200$, 蚂蚁数 $m = 50$, 初始信息素浓度 $\tau = \{1\}_{n \times n}$, 信息素总量 $Q = 1$, 信息素挥发因子 $\rho = 0.1$, 信息启发式因子 $\alpha = 1$, 期望启发式因子 $\beta = 5$ 。

step 2: 迭代次数 iter , 第 iter 代蚂蚁开始搜索。查找初始可排活动 (即初始位置), 给蚂蚁 i 随机分配初始位置 $\text{loc} = 1$ 。

step 3: 从 loc 开始搜索, 查找当前有无可排活动。如果无可排活动, 则根据正在执行活动的最早结束时间更新当前活动下一可排开始时间 t , 并重新执行 step 3。如果有可排活动, 则计算可排活动的转移概率, 选择概率最大的活动, 对它做进一步判断, 即安排它时人力资源占用量是否超限: 若超限, 则根据正在执行活动的最早结束时间, 更新下一可排开始时间 t ,

并重新执行 step 3; 若不超过限, 则确定此活动开始执行并记录活动结束时刻。

step 4: 判断蚂蚁 i 是否完成全程搜索。若否, 则 $\text{loc} = \text{loc} + 1$, 转到 step 3; 若是, 则转 step 5。

step 5: 判断第 iter 代中 m 只蚂蚁是否都完成了搜索。若否, 则 $i = i + 1$, 返回执行 step 2 ~ step 4; 若是, 则寻找第 iter 代中 m 只蚂蚁工期最短的路径, 更新路径信息素 τ 。

step 6: 判断是否 $\text{iter} \geq \text{iter_max}$ 。若否, 则 $\text{iter} = \text{iter} + 1$, 返回执行 step 2 ~ step 5; 若是, 则找出逐代最优路径, 输出结果。

2.2.2 人力资源能力优化阶段

本阶段是针对项目活动中的人力资源约束进行优化, 它以第1阶段得到的优化活动链表为依据, 采用蚁群算法, 以活动的可行解的实际工期为参数构造启发函数, 采用串联进度生成机制及资源冲突消解策略, 针对两个优化目标, 采用多目标决策的分层序列法寻优, 即以最短项目总工期为首要目标, 以最低项目总成本为次要目标, 对人员的配置进行优化, 得到考虑员工工时系数的最优人员调度方案。

1) 蚁群算法设计。

① 蚁群移动规则的建立。

在搜索过程中, 蚂蚁状态转移概率采用式 (13) 进行计算, 但启发函数 $\eta_{x_r, y_s}(t)$ 和可排集合 ζ_i 不同。从活动 j 的可行解 r 到活动 j' 的可行解 s 的转移概率计算公式为

$$p_{j_r, j'_s}^i(t) = \begin{cases} \frac{[\tau_{j_r, j'_s}(t)]^\alpha \times [\eta_{j_r, j'_s}(t)]^\beta}{\sum_{j \in \zeta_i} [\tau_{j_r, j'_s}(t)]^\alpha \times [\eta_{j_r, j'_s}(t)]^\beta}, & j \in \zeta_i; \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

其中: $\tau_{j_r, j'_s}(t)$ 表示 t 时刻从活动 j 的可行解 r 到活动 j' 的可行解 s 的路径上信息素浓度; ζ_i 表示当前时刻蚂蚁 i 待排活动 j' 的可行解集合; $\eta_{j_r, j'_s}(t)$ 表示从活动 j 的可行解 r 到活动 j' 的可行解 s 的路径的启发函数, 且有

$$\eta_{j_r, j'_s}(t) = \frac{1}{\text{time_real}(j'_s)}, \quad (18)$$

这里 $\text{time_real}(j'_s)$ 表示活动 j' 的可行解 s 的实际工期。

② 信息素更新规则的建立。

信息素更新规则为

$$\tau_{j_r, j'_s}(t+1) = (1 - \rho) \times \tau_{j_r, j'_s}(t) + \Delta \tau_{j_r, j'_s}(t),$$

$$\rho \in (0, 1), \Delta\tau_{j_r j'_s}(t) = \sum_{i=1}^m \Delta\tau_{j_r j'_s}^i(t). \quad (19)$$

其中: $\Delta\tau_{j_r j'_s}(t)$ 表示在 t 时刻 m 只蚂蚁从活动 j 的可行解 r 到活动 j' 的可行解 s 的路径上累积释放的信息素增量, $\Delta\tau_{j_r j'_s}^i(t)$ 表示在 t 时刻蚂蚁 i 从活动 j 的可行解 r 到活动 j' 的可行解 s 的路径上释放的信息素. 关于蚂蚁释放信息的问题, 同前采用 ant cycle system 模型计算每只蚂蚁释放的信息素量

$$\Delta\tau_{j_r j'_s}^i(t) = \begin{cases} \frac{Q}{\text{time}(i)}, & \text{蚂蚁 } i \text{ 从活动 } j \text{ 的可行解 } r \\ & \text{访问了活动 } j' \text{ 的可行解 } s; \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

其中: Q 表示信息素强度, $\text{time}(i)$ 表示蚂蚁 i 在本次循环中走过路径的总工期.

2) 人力资源冲突消解.

针对不同项目中活动出现的资源争夺的情况, 本阶段采用如下策略对资源冲突进行消解. 如果同一时段人员使用量超过人员使用量的上限, 则需等待人员释放后再开始待排活动的排产; 或者如果出现待排活动的可行方案中人员全部占用的情况, 则需等待人员释放后再开始待排活动的排产.

3) 构建人力资源优化算法.

将串联调度生成机制和资源冲突消解策略融合在蚁群算法中, 构建了以最短工期为第1目标、最低成本为第2目标对资源进行优化的蚁群算法 (time optimization with resource constraints by ant colony algorithm, TORC_ACA), 具体步骤如下.

step 1: 参数初始化. 活动开始时间 $t = 0$; 最大迭代次数 $\text{iter_max} = 200$; 初始信息素浓度 $\tau = \{1\}_{n \times n}$; 信息素总量 $Q = 1$; 信息素挥发因子 $\rho = 0.1$; 信息启发式因子 $\alpha = 1$; 期望启发式因子 $\beta = 5$; 蚂蚁数 $m = 2 \times \max\{M\}$, M 表示所有活动可行解的数量集合, $\max\{M\}$ 表示集合 M 中元素的最大值.

step 2: 迭代次数为 iter , 第 iter 代蚂蚁开始搜索. 读取第1阶段排产得到 $\text{loc} = 1$ 位置的活动代码, 随机分配蚂蚁 i 在 $\text{loc} = 1$ 时的初始位置 (可行解) 并前进到 $\text{loc} = 2$.

step 3: 读取第1阶段排产得到的 loc 位置活动代码, 搜索 loc 活动全部可行人员组合, 并判断当前活动可行解中人员是否均在占用. 如果是, 则根据正在执行活动的最早结束时间, 更新当前活动下一可排开

始时间 t , 并重新执行 step 3. 如果否, 则计算人员未占用的活动可行解转移概率, 选择概率最大的活动可行解, 对它做进一步判断, 即安排它时人力资源占用量是否超限: 若超限, 则根据正在执行活动的最早结束时间, 更新当前活动下一可排开始时间 t , 并重新执行 step 3; 若不超过限, 则排产活动并记录最大概率可行解及活动结束时刻.

step 4: 判断蚂蚁 i 是否完成全程搜索. 若否, 则 $\text{loc} = \text{loc} + 1$, 转到 step 3; 若是, 则转 step 5.

step 5: 判断第 iter 代中 m 只蚂蚁是否都完成了搜索. 若否, 则 $i = i + 1$, 返回执行 step 2 ~ step 4; 若是, 则寻找第 iter 代中 m 只蚂蚁工期最短的路径, 再寻找工期最短路径中的费用最低路径, 更新路径信息素 τ .

step 6: 判断是否 $\text{iter} \geq \text{iter_max}$. 若否, 则 $\text{iter} = \text{iter} + 1$, 返回执行 step 2 ~ step 5; 若是, 则找出逐代最优路径, 输出结果.

3 算法测试及实例分析

3.1 算例构建

为了验证本文设计的两阶段优化算法求解 S-MRCPSP 的有效性, 选择测试问题算例库 PSPLIB 进行数值实验, 因没有现成的测试模型和算法的算例, 需要根据问题模型对 PROGEN 生成器添加一些参数, 生成适应本文模型的实验算例. 本文设置 PROGEN 中 expl.bas 文件的参数, 产生 6 组共 30 个算例, 分别包含实活动数为 $J = 13, 15, 17, 19, 21, 23$, 每组包含 5 个算例. 实验算例中, 活动顺序关系的生成与 PROGEN 一致. 研究 S-MRCPSP 模型还需要再设定一些测试算例库中没有的参数, 这些参数的生成标准见表 1.

表 1 参数的生成标准

参数	标准
p_z	物料 z 的单价服从 3 ~ 10 的离散均匀分布
T_{jk}	工时定额服从 5 ~ 15 的离散均匀分布
L_{jkz}	项目中各活动的物料限量服从 140 ~ 500 的离散均匀分布
N_{jk}	设定的分配给各活动的员工数服从 1 ~ 3 的离散均匀分布
K_{ijk}	员工完成中各活动的工时系数服从 0.4 ~ 1.8 的离散均匀分布
Φ_{ijkz}	计算公式 Φ_{ijkz} 为各活动的标准人物料 z 消耗率 K_{ijk}
N_i	设定的各时段分配的员工数上限值取员工总数的 80 %
NR	员工的总数服从 10 ~ 15 的离散均匀分布

3.2 算例测试结果

测试算例采用的硬件平台为 Windows 7 操作系统, Intel(R)Core i7 处理器, 3.6 GHz 主频, 4 G 内存, Matlab 7.0 仿真软件.

表 2 算例测试结果 1

实活动数	传统MRCPSP			S-MRCPSP(两阶段算法)			S-MRCPSP (GA)		
	平均总工期	平均总成本	平均CPU时间/s	平均总工期	平均总成本	平均CPU时间/s	平均总工期	平均总成本	平均CPU时间/s
$J = 13$	37	36 730	5.6	26	19 212	258	26	19 439	131
$J = 15$	49	58 984	7.2	34	30 807	657	36	33 100	185
$J = 17$	50	60 631	8.6	34	30 806	530	35	32 295	201
$J = 19$	60	52 661	12.6	40	26 269	565	42	28 042	232
$J = 21$	51	61 870	13.4	37	32 060	489	41	36 163	278
$J = 23$	59	83 762	14.8	41	43 782	1 085	44	45 501	324

由表2可见,S-MRCPSP与传统MRCPSP相比较,项目总工期平均缩短约30%,项目总成本平均降低约48%,由此说明针对企业研发项目群,若考虑人员的胜任力水平差异,则能得到更优化的项目调度计划。

表2将本文提出的两阶段算法和文献[17]中GA进行测试和对比,两阶段算法将项目总工期平均缩短约5%,项目总成本平均降低约6%,对于本文研究的S-MRCPSP,当实活动 $J = 13$ 时,可行解最少达到 $2.78e+13$ 个,搜索如此大解空间相当耗时,从CPU运行时间角度比较,两阶段算法稍逊于遗传算法,但对于企业决策者而言可以接受。

3.3 实例分析

为节省篇幅,采用PROGEN生成器产生一个网络实例^[14],并对求解结果进行实例分析。将传统MRCPSP与本文研究的S-MRCPSP最优调度计划进行对比。不考虑员工胜任力水平差异求得的最优调度计划,所有项目任务均选用标准人执行,这属于传统的多模式调度研究,其最优调度结果为:多项目总工期为46,总成本为19 360。在考虑了员工胜任力水平差异求得的最优调度计划中(即本文运用两阶段算法求解的S-MRCPSP),多项目总工期缩短至40,总成本降至16 604。通过对比,本文研究的S-MRCPSP将总工期缩短了13%,总成本节省了15.6%。

表 3 传统MRCPSP与S-MRCPSP最优调度计划比较

活动	传统MRCPSP				S-MRCPSP(两阶段算法)			
	人员组合	工期	费用	最早开始- 最早结束时间	人员组合	工期	费用	最早开始- 最早结束时间
a_{11}	2个标准人	7	560	0-7	2,9	4	580	0-4
a_{12}	2个标准人	9	720	0-9	1,6	6	960	0-6
a_{13}	3个标准人	12	960	9-21	1,6,7	11	2 761	6-17
a_{14}	2个标准人	10	800	21-31	2,11	7	1 060	17-24
b_{21}	2个标准人	19	1 520	0-19	8,10	17	2 601	0-17
b_{22}	2个标准人	12	960	7-19	4,11	12	2 004	4-16
b_{23}	2个标准人	12	960	21-33	9,12	10	1 730	17-27
b_{24}	3个标准人	14	1 120	19-33	1,3,10	15	3 540	17-32
b_{25}	2个标准人	13	1 040	33-46	2,5	8	1 368	32-40

表3数据结果表明:本文提出的S-MRCPSP能够进一步缩短传统MRCPSP的项目总工期和总成本。

3.4 实例算法对比

文献[14]采用动态规划算法求得了该实例的精确解,文献[17]采用遗传算法求解了该实例。

表 4 求解S-MRCPSP算法对比

对比参数	遗传算法	两阶段算法	精确算法
项目总工期/天	41	40	40
项目总工期偏差/%	2.5	0	—
项目总成本/元	17 284	16 604	16 335
项目总成本偏差/%	5.8	1.6	—
CPU时间/s	113	135	—

从表4中可看出,两阶段算法求得的项目总工期和总成本均优于遗传算法的解,而且两阶段算法求得

的项目总工期等于精确解的最短工期,求得的项目总成本与精确解最低成本的偏差仅为1.6%。但就CPU运算时间而言,遗传算法优于两阶段算法。

4 结 论

本文将人力资源配置与项目调度问题相结合,对传统MRCPSP进行拓展,提出了特殊的“多模式”项目人力资源调度问题和两阶段优化算法。6组仿真算例和1个项目实例结果表明:考虑了人员胜任力水平差异的S-MRCPSP比不考虑胜任力差异的传统MRCPSP的多项目总工期更短、总成本也更低。S-MRCPSP模型更符合研发项目群管理实践需要。同时,对于求解中小规模的S-MRCPSP问题,本文提出的两阶段优化算法比遗传算法具有更好的求解质

量. 但当问题规模较大时, 无法避免模型的NP-hard特征, 因此后续研究将尝试采用其他新型群体智能优化算法^[18], 以实现求解本文模型的更好性能.

参考文献(References)

- [1] Elonen S, Artto K A. Problems in managing internal development projects in multi-project environments[J]. International Journal of Project Management, 2003, 21(6): 395-402.
- [2] Naber A, Kolisch R. MIP models for resource-constrained project scheduling with flexible resource profiles[J]. European Journal of Operational Research, 2014, 239(2): 335-348.
- [3] 王一帆, 刘士新, 陈迪. 求解多技能人力资源约束的项目调度问题的两阶段算法[J]. 东北大学学报: 自然科学版, 2014, 35(2): 184-189.
(Wang Y F, Liu S X, Chen D. A two-stage algorithm for project scheduling problems with multi-skilled workforce constraint[J]. Journal of Northeastern University: Natural Science, 2014, 35(2): 184-189.)
- [4] 任逸飞, 陆志强, 刘欣仪, 等. 考虑技能水平的多技能资源约束项目调度[J]. 浙江大学学报: 工学版, 2017, 51(5): 1000-1006.
(Ren Y F, Lu Z Q, Liu X Y, et al. Project scheduling problem with hierarchical levels of skills[J]. Journal of Zhejiang University: Engineering Science, 2017, 51(5): 1000-1006.)
- [5] Wang L, Zheng X L. A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem[J]. Swarm & Evolutionary Computation, 2018, 38(2): 54-63.
- [6] 邓富民, 梁学栋, 刘爱军, 等. 多资源约束下改进NSGA-II算法的手术调度[J]. 系统工程理论与实践, 2012, 32(6): 1337-1345.
(Deng F M, Liang X D, Liu A J, et al. Surgical operation scheduling with multi-resource constrained based on the improved NSGA-II algorithm[J]. Systems Engineering — Theory & Practice, 2012, 32(6): 1337-1345.)
- [7] Chen W N, Zhang J. Ant colony optimization for software project scheduling and staffing with an event-based scheduler[J]. IEEE Transactions on Software Engineering, 2012, 39(1): 1-17.
- [8] Zheng H Y, Wang L, Zheng X L. Teaching-learning-based optimization algorithm for multi-skill resource constrained project scheduling problem[J]. Soft Computing, 2017, 21(6): 1537-1548.
- [9] Tritschler M, Naber A, Kolisch R. A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles[J]. European Journal of Operational Research, 2017, 262(1): 262-273.
- [10] Heimerl C, Kolisch R. Scheduling and staffing multiple projects with a multi-skilled workforce[J]. OR Spectrum, 2010, 32(2): 343-368.
- [11] Chen J J, Zhu J L, Zhang D N. Multi-project scheduling problem with human resources based on dynamic programming and staff time coefficient[C]. International Conference on Management Science & Engineering. Harbin: IEEE, 2014: 1012-1018.
- [12] 李婧婷. 研发项目群人员能力及其对员工绩效的作用机理研究[D]. 哈尔滨: 哈尔滨工业大学管理学院, 2015.
(Li J T. The capacities of R&D programme staff and its mechanism of employee performance[D]. Harbin: School of Management, Harbin Institute of Technology, 2015.)
- [13] 张静文. 项目调度优化模型与方法的拓展[M]. 西安: 西安交通大学出版社, 2014: 71-76.
(Zhang J W. Extensions for models and methods of project scheduling problems[M]. Xi'an: Xi'an Jiaotong University Press, 2014: 71-76.)
- [14] 陈俊杰, 同淑荣, 聂亚菲, 等. 考虑胜任力水平的研发项目群人力资源调度[J]. 计算机工程与应用, 2019, 55(3): 209-218.
(Chen J J, Tong S R, Nie Y F, et al. R&D program scheduling and staff assignment with hierarchical levels of competency[J]. Computer Engineering and Applications, 2019, 55(3): 209-218.)
- [15] 刘士新. 项目优化调度理论与方法[M]. 北京: 机械工业出版社, 2007: 47-48.
(Liu S X. Project scheduling theory and method[M]. Beijing: China Machine Press, 2007: 47-48.)
- [16] 王军强. 一种求解资源受限多项目调度问题的分解算法[J]. 计算机集成制造系统, 2013, 19(1): 83-96.
(Wang J Q. Decomposition algorithm for resource-constrained multi-project scheduling problem[J]. Computer Integrated Manufacturing Systems, 2013, 19(1): 83-96.)
- [17] 王曜. 软件企业人力资源优化调度及算例研究[D]. 西安: 西北工业大学管理学院, 2018.
(Wang Y. Software enterprises human resources optimization scheduling and test instances research[D]. Xi'an: School of Management, Northwestern Polytechnical University, 2018.)
- [18] 林诗洁, 董晨, 陈明志, 等. 新型群智能优化算法综述[J]. 计算机工程与应用, 2018, 54(12): 1-9.
(Lin S J, Dong C, Chen M Z, et al. Summary of new group intelligent optimization algorithms[J]. Computer Engineering and Applications, 2018, 54(12): 1-9.)

作者简介

陈俊杰(1975—), 女, 讲师, 博士生, 从事工业工程、项目调度的研究, E-mail: junjiechen75@nwpu.edu.cn;
同淑荣(1963—), 女, 教授, 博士生导师, 从事产品设计管理、质量管理等研究, E-mail: stong@nwpu.edu.cn;
叶正梗(1989—), 男, 助理工程师, 博士生, 从事复杂系统建模的研究, E-mail: yezhengeng@mail.nwpu.edu.cn;
张静文(1976—), 女, 教授, 博士生导师, 从事项目调度等研究, E-mail: zhangjingwen@nwpu.edu.cn;
王曜(1995—), 男, 硕士生, 从事项目调度和数据分析的研究, E-mail: wangy787@mail2.sysu.edu.cn;
聂亚菲(1993—), 女, 博士生, 从事工业工程的研究, E-mail: nyf1993@mail.nwpu.edu.cn;
张雨芊(1996—), 女, 硕士生, 从事项目调度的研究, E-mail: yuqianzhang@mail.nwpu.edu.cn.

(责任编辑: 闫妍)