

考虑知识扩散的大型产品装配生产调度模型及算法

陆志强, 马新一

(同济大学 机械与能源工程学院, 上海 201804)

摘要: 人力资源的技能结构可以显著影响生产线的效率, 而线内师徒培训引起的知识扩散现象是改善人力资源结构的手段之一。引入社交网络描述知识扩散现象, 提出在已知目标人力资源结构的条件下考虑知识扩散下资源技能效率提升的多技能资源受限项目调度问题, 建立以最小化达成培训目标所需项目个数和平均每个项目所需成本为目标的混合整数规划模型。针对该模型, 设计了一种改进的基于分解的多目标进化算法, 并嵌入动态规划资源分配算子和任务聚类贪婪调度计划生成机制。通过数值实验证明算法优越性。

关键词: 项目调度; 人力资源培训; 知识扩散; 社交网络

中图分类号: TP18; F273

文献标志码: A

Modeling and Algorithm for Assembly Production Scheduling Problem for Large-Scale Products Considering Knowledge Diffusion

LU Zhiqiang, MA Xinyi

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

Abstract: The skill structure of human resources can significantly affect the efficiency of production lines, and the knowledge diffusion resulting from in-line apprenticeship training is one way to improve this structure. To describe the knowledge diffusion phenomenon, a social network is introduced, and a multi-skilled resource-constrained project scheduling problem is proposed considering the improvement of resource and skill efficiency under knowledge diffusion, assuming a known target human resource structure. A mixed integer programming model is developed to minimize both the number of projects required to achieve training objectives and the average cost required per project. An improved multi-objective evolutionary algorithm based on decomposition is designed to address this model,

incorporating a dynamic programming resource allocation operator and a task clustering greedy schedule generation scheme. The superiority of the algorithm is demonstrated through numerical experiments.

Keywords: project scheduling; human resources training; knowledge diffusion; social network

由于客户需求变化以及减少库存等需求, 现代制造业逐渐从大批量生产向定制化生产转变, 项目需求反复变化对制造企业人力资源管理带来新的挑战。在项目生产稳定阶段, 团队成员的技能水平能够充分满足项目需求。然而, 在项目启动初期往往存在现有人力资源结构与项目需求不匹配的现象。在这一时期团队将经历较长的试生产和培训过程以进行人力资源结构优化, 更好地满足项目稳定生产需求。本文旨在缩短项目试生产过程, 并降低相关培训过程的成本。

学习效应^[1]指在生产过程中工人会随时间逐渐提高其技能水平, 在项目试生产过程中, 员工通常需要学习复杂的操作流程。线内学习过程有边做边练、师徒培训等。线内师徒培训可以用于培养操作员正确、高效地操作生产设备和工具, 以生产出符合要求的产品。在生产线上, 对人力资源学习效应的研究有助于提高生产效率, 降低成本。近年来, 调度和学习效应集成问题逐渐成为热点。Chen 等^[2]研究了多项目调度和多技能员工分配问题, 考虑工人学习效应和在工人缺口时雇佣外部工人的情况。胡金昌等^[3]研究了基于学习效应的一人多机模式下的调度问题。Jemmali 等^[4]在并行机调度问题中考虑 DeJong 学习效应, 并以最小化完工时间为目标。Heuser 等^[5]在单机调度问题中考虑工人学习效应和遗忘效应。Zhang 等^[6]在无关并行机调度问题中考虑工人学习效应, 并

收稿日期: 2023-12-12

基金项目: 国家自然科学基金(61473211)

第一作者: 陆志强, 教授, 工学博士, 主要研究方向为物流与供应链的建模和优化以及生产工程。

E-mail: zhiqianglu@tongji.edu.cn



论文
拓展
介绍

根据技能等级对工人分组,提出可以嵌入多种启发式规则的组进化算法求解该问题。陆志强等^[7]在以飞机生产为背景,在多技能资源投入问题中考虑工人学习效应,并以快速、低成本地提高现有资源的效率为目标。现有文献研究主要集中于资源通过自身实践引起的效率提高,而对团队中由师徒培训的知识扩散现象引起的学习效应研究较少。李斌等^[8]定义团队交互学习效应,建立以最小化完工时长为目标的作业可中断的项目调度问题,但没有考虑人力资源多技能属性。Hosseinian等^[9]首次利用社交网络描述团队中的知识扩散,但在研究中将知识扩散引起的效率提高和调度分离,没有考虑在参与不同作业时资源团队结构异质性对知识扩散的影响。

知识扩散是指将知识从一个人或一个团体传播给其他人或团体的过程,在生产线上,知识扩散是人力资源提高技能水平的重要途径。目前对知识扩散的描述主要从扩散方式展开,李柏洲等^[10]考虑不同个体知识异质性,提出知识生态位概念,知识生态位重叠会影响知识扩散的效果。余谦等^[11]利用多维临近创新网络描述知识扩散,将临近性分为地理临近、技术临近和社会临近,并分别建立知识扩散模型。社交网络模型利用网络结构研究社会个体之间社交互动对整个社会以及一些社会现象的影响。目前社交网络已经广泛应用于企业知识管理领域^[12]。Zheng等^[13]将个体分为易感者S、感染者I和犹豫不决者H,并在传染病模型和社交网络基础上建立SIH知识扩散模型。Song等^[14]利用社交网络描述隐性知识扩散,并利用蒙特卡洛仿真证明该方法的有效性。现有文献对知识扩散的研究主要集中于企业合作等宏观层面,对微观层面的生产线中师徒培训的知识扩散机制研究较少。

基于当前研究局限性,本文考虑资源多技能属性,进一步完善Cremonini^[15]提出的知识扩散模型,规定只有在使用相同技能共同参与同一作业的资源之间才会发生知识扩散,且知识扩散呈现为高效资源与低效资源之间的一对一培训,从而更加具体地描述实际作业中师徒培训学习过程。并构建在目标人力资源结构已知情况下以最短化培训时长与最小化平均项目成本为目标的混合整数规划模型。

1 问题描述与数学模型

1.1 问题描述与基本假设

如图1所示,企业根据生产线内外部因素确定

目标人力资源效率结构。生产线中 m 名工人重复装配 N 件某型号大型产品,每件产品可视为一个项目,项目由 n 项具有时序约束的作业组成,工期均为 T 。生产初期人力资源结构已知,生产线中人力资源掌握产品生产所需的所有技能 $S=\{1, 2, \dots, s, \dots, k\}$,但不同技能效率不同。每项作业需要至少一种技能,作业对所需技能有标准效率要求和资源数量要求,实际作业时长受参与作业的资源效率影响。在参与同一项作业同一种技能的活动时资源会通过线内师徒培训发生从高技能水平资源向低技能水平资源的知识扩散,提高低技能水平资源的技能熟练度,资源技能效率增长采用社交网络模型描述。在项目依次进行的同时,资源技能效率通过培训得到提升,达成目标人力资源结构后对培训效果进行评估。

问题决策涵盖资源分配决策和调度计划决策2个主要方面。资源分配决策包括将资源分配给不同作业以及将参与同一作业的资源分配给不同技能2个方面。由于高效率资源能够更迅速地培训其他资源,并缩短项目完成时间,但用工成本较高,因此,考虑这2个相互矛盾的因素,选取每个项目的平均成本和达到目标资源结构经过的项目数量作为目标函数。为更清晰地描述问题,给出以下假设:

(1) 参与作业每个所需技能的资源中至少需要1个效率水平不小于标准效率要求的资源参与才能保证作业顺利开展。

(2) 实际作业时长由作业所需所有技能中耗时最长的技能决定,考虑到作业中各技能耦合效应,参与作业的所有资源在整项作业完成后方可释放。

(3) 人力资源培养目标的确定受到多种内外部因素影响^[16],不在问题研究范围内,因此假设人力资源培训目标事先已知且合理。

(4) 所有项目依次生产,新项目仅可在当前项目完成后开展。

(5) 项目允许拖期,但拖期需要承担拖期成本。

1.2 基于社交网络的知识扩散模型

一对一师徒培训是生产线中常见的知识扩散模式^[17],本文知识扩散模型以一对一学徒培训为背景,做出以下假设:

(1) 作业中资源当前使用技能效率超过作业在该技能所需标准效率时具备培训资格,可以对其他参与该技能的效率低于该资源的资源进行培训。

(2) 具备培训资格的资源在作业时对受培训资源进行一对一培训,且所有受培训资源获得的培训时长平均分配。

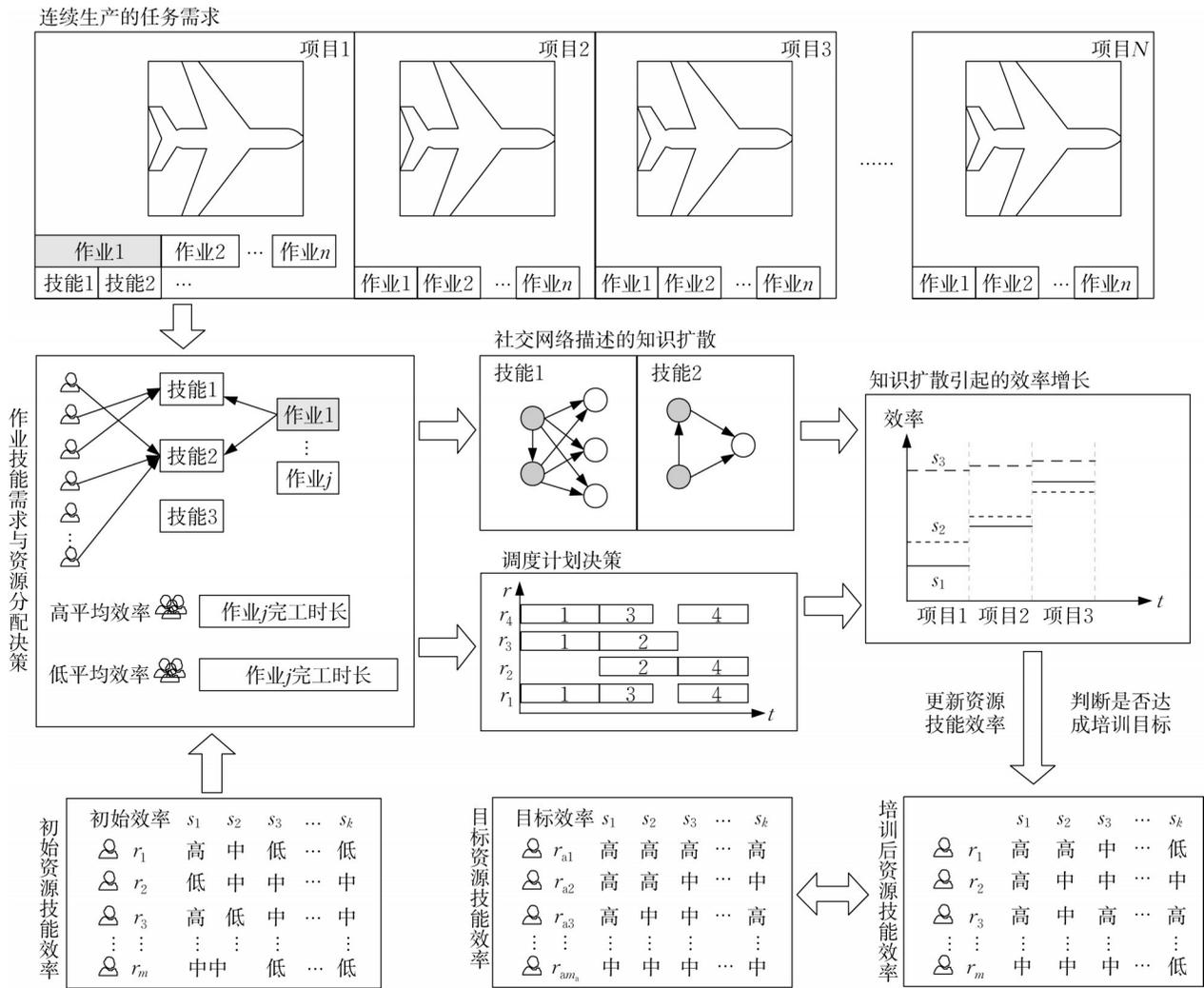


图1 问题描述

Fig. 1 Description of problem

(3) 由于技能增长速度较慢,假设在同一项目中资源效率不变,在项目结束后更新效率。

采用社交网络模型描述资源间知识扩散和技能效率增长,如图2所示。假设当前项目*i*中作业*j*需要*s* = 1和*s* = 2这2种技能,对技能*s*所需资源数量为*l_{js}*,在对作业分配资源后则可以得到2张社交网络。网络中每个节点表示一个资源,资源*r*的效率*η_{(i-1)rs}*超过作业标准技能效率要求*e_{js}*则具有培训资格,如灰色节点所示,效率低于*e_{js}*的资源不具备培训资格,如白色节点所示。节点间有向弧表示两资源之间的知识扩散方向,可引起受培训资源技能效率提高。培训形式为一对一培训,根据参与作业资源的效率计算作业时长与各资源受培训时长,采用Cremonini等^[15]提出的知识扩散模型进行效率更新。

Cremonini等提出的模型定义如式(1)所示。

$$\Delta\eta_{ijrs} = \sum_{q=1}^m \frac{\eta_{(i-1)qs} - \eta_{(i-1)rs}}{\gamma + \rho e^{-\frac{h_{(i-1)rq}}{\theta}}}, \forall i, j, r, s \quad (1)$$

式中:*η_{(i-1)qs}* - *η_{(i-1)rs}*为项目*i*结束前两资源*q*和*r*在技能*s*上的效率差距;*h_{(i-1)rq}*为在项目*i*结束前两资源*q*和*r*历史交互培训总时长,用以描述双方信任程度;*γ*、*ρ*和*θ*为学习率参数,分别控制效率增长比例上限和下限以及双方信任程度的影响。当一个资源在作业中接受多个不同资源培训时,其效率增量*Δη_{ijrs}*为所有资源对其培训效率增长总和。在该知识扩散模型基础上考虑本次培训交互时长影响,效率增长量与本次交互时长成正比,如式(2)所示:

$$\Delta\eta_{ijrs} = \sum_{q=1}^m \frac{(\eta_{(i-1)qs} - \eta_{(i-1)rs}) \cdot \Delta h_{ijrq}}{\gamma + \rho e^{-\frac{h_{(i-1)rq}}{\theta}}}, \forall i, j, r, s \quad (2)$$

式中:*Δh_{ijrq}*为两资源在参与项目*i*作业*j*的交互培训时长。两资源在作业中交互时长*Δh_{ijrq}*与作业完工

步骤1: 根据作业所需的技能标准效率要求, 每项技能所需的资源数分配资源并绘社交网络图

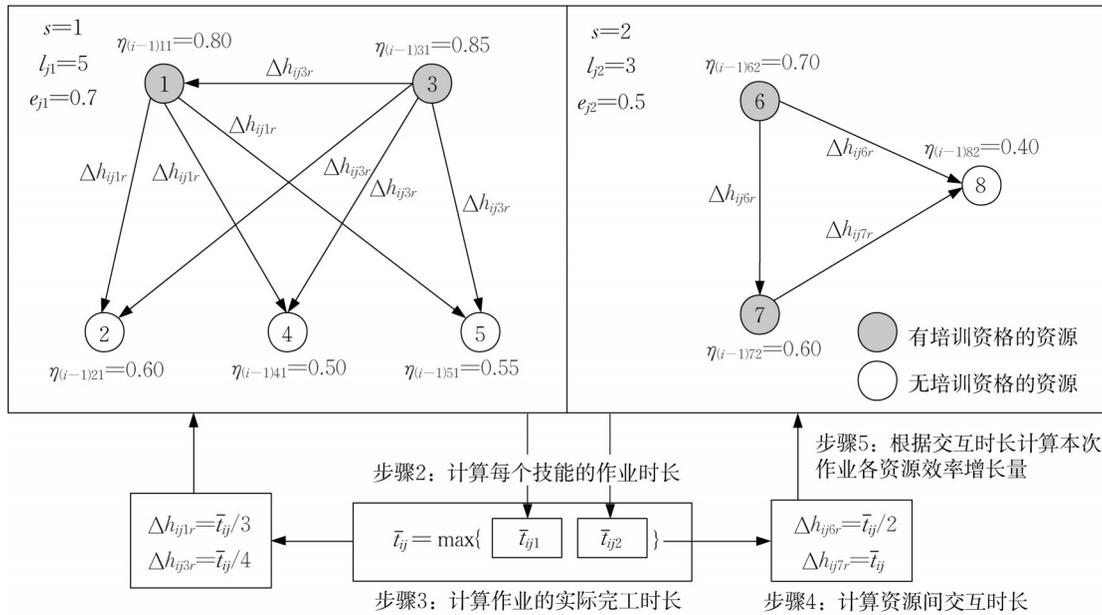


图 2 基于社交网络的效率增长模型示意

Fig. 2 Schematic diagram of efficiency growth model based on social network

时长相关。作业中各技能实际作业时长 \bar{t}_{ij} 由分配给该技能的资源平均效率与标准效率 e_{js} 的比值决定, 如式(3)所示:

$$\bar{t}_{ijs} = \left[t_j \cdot \frac{e_{js}}{\left(\sum_{r=1}^m \lambda_{ijrs} \cdot \eta_{(i-1)rs} \right) / \left(\sum_{r=1}^m \lambda_{ijrs} \right)} \right], \forall i, j, s \quad (3)$$

式中: t_j 为作业标准工时; λ_{ijrs} 为资源 r 是否参与项目 i 作业 j 技能 s 的变量, λ_{ijrs} 为 0 或 1。作业实际作业时长 \bar{t}_{ij} 由作业所需各项技能中耗时最长的技能决定, 如式(4)所示:

$$\bar{t}_{ij} = \max_s (\bar{t}_{ijs}) \quad (4)$$

资源效率增长采用一对一师徒培训模式, 具有培训资格的资源将作业实际完工时长均等地用于培训其他资源, 交互培训时长 Δh_{ijrq} 定义如式(5)所示:

$$\Delta h_{ijrq} = \kappa_{ijrqs} \cdot \frac{\bar{t}_{ij}}{\sum_{r=1}^m \kappa_{ijrqs}} \quad (5)$$

式中: κ_{ijrqs} 为该作业中资源 q 是否对资源 r 进行培训的变量, κ_{ijrqs} 为 0 或 1。

1.3 符号定义

模型中涉及的变量与参数分别如表 1 和表 2 所示。

表 1 模型所涉及变量

Tab. 1 Variables involved in the model

变量符号	定义
x_{ijd}	0-1 变量, 项目 i 的作业 j 在 d 时刻正在执行为 1, 否则为 0
y_{ijrsd}	0-1 变量, 资源 r 在 d 时刻使用技能 s 操作项目 i 的作业 j 为 1, 否则为 0
$z_{rr'}$	0-1 变量, 资源 r 的培训方向为资源 r' 则为 1, 否则为 0
λ_{ijrs}	中间变量, 0-1 变量, 资源 r 使用技能 s 参与项目 i 的作业 j 为 1, 否则为 0
μ_i	中间变量, 0-1 变量, 在项目 i 后达到目标人力资源结构为 1, 否则为 0
κ_{ijrqs}	中间变量, 0-1 变量, 在项目 i 作业 j 中资源 q 对资源 r 的技能 s 进行培训则为 1, 否则为 0

1.4 数学模型

目标函数为

$$\min F = (f_1, f_2) \quad (6)$$

$$f_1 = \sum_{i=1}^{f_2} \left(\sum_{j=1}^n \sum_{r=1}^m \sum_{s=1}^k c_{ir} \cdot (\lambda_{ijrs} \cdot \bar{t}_{ij}) + c \cdot \Delta T_i \right) / f_2 \quad (7)$$

$$f_2 = \min_{i \in I} \{ i \cdot \mu_i + (1 - \mu_i) \cdot M \} \quad (8)$$

表2 模型所涉及参数
Tab. 2 Parameters involved in the model

参数符号	定义	参数符号	定义
I	项目集合 $I = \{1, 2, \dots, i, \dots\}$	$\Delta\eta_{ijrs}$	资源 r 在项目 i 作业 j 中技能 s 的效率增长值
T	项目工期	η_{irs}	i 项目后资源 r 技能 s 的效率
T_i	项目 i 实际工期	$\hat{\eta}_{r,s}$	资源 r_a 对技能 s 的目标效率
c	单位时间拖期成本	\bar{t}_{ij}	项目 i 作业 j 的实际作业时长
ΔT_i	项目 i 拖期时间	e_{js}	作业 j 使用技能 s 部分的标准效率要求
J	作业集合 $J = \{1, 2, 3, \dots, j, \dots, n\}$	l_{js}	使用技能 s 参与作业 j 的资源数量
P_j	作业 j 的紧前作业集合	γ	学习率参数, 控制单位时间技能增长上限
R	现有资源集合 $R = \{1, 2, 3, \dots, r, \dots, m\}$	ρ	学习率参数, 控制单位时间技能增长下限
R_a	目标资源集合 $R_a = \{1, 2, 3, \dots, r_a, \dots, m_a\}$	θ	学习率参数, 控制交互作用的影响
S	技能集合 $S = \{1, 2, \dots, s, \dots, k\}$	c_{ir}	项目 i 中使用资源 r 的(单位时间)成本
Δh_{ijrq}	资源 r 与资源 q 在项目 i 作业 j 中的交互时长	M	一个极大的常数
h_{irq}	项目 i 完成后资源 r 与资源 q 的总交互时长		

约束条件为

$$\lambda_{ijrs} = \begin{cases} 1, & \text{若 } \sum_{d=1}^{T_i} y_{ijrsd} \geq 1, \forall i, j, r, s \\ 0, & \text{其他} \end{cases} \quad (9)$$

$$\mu_i = \begin{cases} 1, & \text{若 } \eta_{irs} \cdot z_{rr_a} \geq \hat{\eta}_{r,s}, \forall r, r_a, s \\ 0, & \text{其他} \end{cases} \quad (10)$$

$$\kappa_{ijrqs} = \begin{cases} 1, & \text{若 } (\eta_{(i-1)qs} > \eta_{(i-1)rs}) \wedge (\eta_{(i-1)qs} > e_{js}) \wedge \\ & (\lambda_{ijqs} = 1) \wedge (\lambda_{ijrs} = 1) \\ 0, & \text{其他} \end{cases}, \quad (11)$$

$$\Delta T_i = \begin{cases} T_i - T, & \text{若 } T_i - T > 0, \forall i \\ 0, & \text{其他} \end{cases} \quad (12)$$

$$\sum_{d=1}^{T_i} x_{ijd} = \bar{t}_{ij}, \forall i, j \quad (13)$$

$$\bar{t}_{ip} x_{ijd} \leq \sum_{\tau=1}^{d-1} x_{ip\tau}, p \in P_j, \forall i, j, d \quad (14)$$

$$\bar{t}_{ij} \cdot x_{ijd} - \bar{t}_{ij} \cdot x_{ij(d+1)} + \sum_{\tau=d+2}^{T_i} x_{ij\tau} \leq \bar{t}_{ij}, \forall i, j, d \quad (15)$$

$$l_{js} = \sum_{r=1}^m \lambda_{ijrs}, \forall i, j, s \quad (16)$$

$$\max_{r \in R} (\lambda_{ijrs} \cdot \eta_{(i-1)rs}) > e_{js}, \forall i, j, s \quad (17)$$

$$\bar{t}_{ij} \times y_{ijrsd} = \sum_{d=1}^{T_i} y_{ijrsd}, \forall i, j, r, s, d \quad (18)$$

$$\Delta h_{ijrq} = \Delta h_{ijqr}, \forall i, j, r, q \quad (19)$$

$$h_{irq} = h_{(i-1)r_q} + \sum_{j=1}^n \Delta h_{ijrq}, \forall i, r, q \quad (20)$$

$$\Delta \eta_{ijrs} = \sum_{q=1}^m \left(\frac{(\eta_{(i-1)qs} - \eta_{(i-1)rs}) \cdot \Delta h_{ijrq}}{\gamma + \rho e^{-\frac{h_{(i-1)r_q}}{\theta}}} \right), \forall i, j, r, s \quad (21)$$

$$\eta_{irs} = \eta_{(i-1)rs} + \sum_{j=1}^n \Delta \eta_{ijrs}, \forall i, r, s \quad (22)$$

$$\sum_{j=1}^n \sum_{s=1}^k y_{ijrsd} \leq 1, \forall i, r, d \quad (23)$$

$$\sum_{r_a=1}^{m_a} z_{rr_a} \leq 1, \forall r \quad (24)$$

$$\sum_{r=1}^m z_{rr_a} = 1, \forall r_a \quad (25)$$

$$x_{ijd}, y_{ijrsd}, z_{rr_a} \in \{0, 1\}, \forall i, j, r, s, d, r_a \quad (26)$$

式(6)表示最小化达成资源培训目标所需要项目个数和平均每个项目的成本;式(7)表示现有资源达成培训目标时所需要的项目平均成本;式(8)表示现有资源达成培训目标所需要经过的项目个数;式(9)至式(11)表示决策变量与中间变量的函数关系;式(12)表示项目拖期时间计算;式(13)表示作业必须全部完成;式(14)表示作业时序约束;式(15)表示作业不允许中断;式(16)表示作业所需技能的资源数量约束;式(17)表示在使用某项技能参与作业的所有资源中,至少要有1个资源的效率大于标准效率要求;式(18)表示资源为非抢占式;式(19)表示交互时长增长是双向的;式(20)表示一个项目完成后

资源交互总时长更新;式(21)表示作业中资源效率增量计算;式(22)表示一个项目完成后资源技能效率更新;式(23)表示某时刻一个资源只能为一项作业的技能提供需求;式(24)表示每个资源最多有一个培训目标;式(25)表示每个培训目标只能应用于一个资源;式(26)表示决策变量取值范围。

2 改进的 MOEA/D-NG 算法设计

多技能资源受限项目调度问题 (Multi-Skill Resource-Constrained Project Scheduling Problem, MS-RCPSP) 已被证明为 NP-Hard 问题^[18], 本问题考

虑资源多技能属性以及资源间交互培训产生的知识扩散, 解空间广泛, 且现有商用软件无法求得多目标优化问题精确帕累托前沿。MOEA/D (Multi-objective Evolutionary Algorithm Based on Decomposition) 是一种多目标进化算法, 本文在标准 MOEA/D 算法基础上提出基于分解的考虑小生境的多目标进化算法 (Niche-Guided Multi-objective Evolutionary Algorithm Based on Decomposition, MOEA/D-NG), 算法上层利用动态规划对参与作业的资源 and 技能进行分配, 下层根据各作业的资源使用情况聚类进行优先级分配并利用资源插空机制生成调度计划。算法流程如图 3 所示。

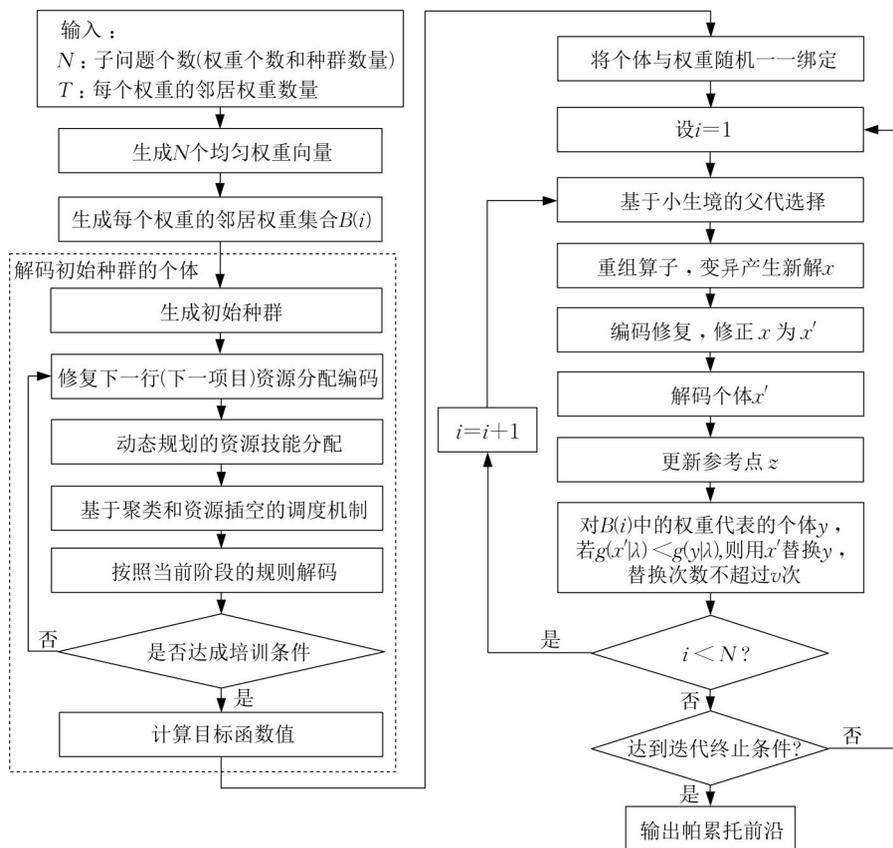


图3 MOEA/D-NG 算法流程
Fig.3 Flow of MOEA/D-NG algorithm

MOEA/D-NG 使用双子串实数编码, 如图 4 所示。第 1 子串编码为资源目标层编码, 编码长度为资源培养目标的个数, 编码数字为资源编号, 表示资源培训方向。第 2 子串的资源分配层编码为一个矩阵, 矩阵每行表示一个项目中资源和作业的分配关系, 每行编码长度为当前项目中所有作业所需资源数之和, 编码数字为资源编号, 矩阵行数受限于项目数量上限。对于违反作业标准技能效率要求的资源分配层编码, 则将其中一个分配给该作业的资源随

机替换成另一个满足效率要求的资源。两子串染色体的交叉复制均采用 POX (precedence operation crossover) 交叉算子^[19]。目标层的变异算子为互换两培训目标对应的资源或选择其他无培训目标的资源替换当前具有培训目标的资源, 资源分配层的编码变异算子为用其他未参与该作业的资源替换参与该作业的资源。

MOEA/D 算法目标函数如式(27)所示:



图4 MOEA/D-NG 编码示意
Fig. 4 MOEA/D-NG encoding

$$\min g = (x | \lambda, z^*) = \max_{1 \leq i < m} \{ \lambda_i | f_i(x) - z_i^* | \}, \forall x \in \Omega \quad (27)$$

式中: z^* 为理想点; λ_i 为权重向量在第 i 个目标维度的权重。MOEA/D 算法规定了每个权重的领域权重, 较优个体的替换均在领域内进行, 更新如式(28)所示, 新个体 y' 将替换邻域内所有适应度低于 y' 的个体。

$$x_j = \begin{cases} y', & \text{若 } g(y' | \lambda_j, z) \leq g(x_j | \lambda_j, z) \\ x_j, & \text{其他} \end{cases}, \forall j \in B(i) \quad (28)$$

MOEA/D-NG 相比 MOEA/D 算法改进之一为限制领域内较优个体的替换次数, 设在邻域内替换次数上限为 v , 防止少量较优解的出现让算法快速收敛陷入局部最优。MOEA/D-NG 相比标准 MOEA/D 改进之二是采用基于小生境的父代选择, 小生境被定义为相似个体组成的领域。标准 MOEA/D 算法的 2 个父代均从领域 $B(i)$ 中随机选择, 而 MOEA/D-NG 将首先计算邻域 $B(i)$ 中个体的相似度 $\varphi(i)$, 如式(29)所示:

$$\varphi(i) = \sum_{j=1}^T \left(\omega_o \frac{O_{ij}}{m_o} + \omega_r \frac{D_{ij}}{m_d} \right) / T, \forall j \in B(i) \quad (29)$$

式中: ω_o 和 ω_r 分别为资源目标层编码和资源分配层编码相似度的权重; m_o 和 m_d 分别为资源培训目标数和资源分配层中每个项目的所有作业所需的资源总数; O_{ij} 和 D_{ij} 分别为种群中个体 i 和个体 j 中资源编号和培训目标对应关系相同的资源数量和各项目各作业中资源分配情况相同的资源数量。MOEA/D-NG 第 1 个父代个体 i 从种群中依次选择, 第 2 个父代个体选取范围 P_2 则根据 $\varphi(i)$ 确定, 如式(30)所示:

$$P_2 = \begin{cases} B(i), & \text{若 } \varphi(i) < \xi \\ \{1, 2, \dots, N\} / B(i), & \text{其他} \end{cases} \quad (30)$$

式中: ξ 为邻域相似度阈值, 当相似度大于 ξ 时, 父代选择范围将扩大到种群中除邻域 $B(i)$ 以外的所有个体, 减少两父代个体相似的情况。

2.1 基于动态规划的资源分配算法

资源分配编码确定每个作业的可用资源池, 资源和技能合理配对可以提高培训效率。采用逆向动态规划算法对参与作业的资源进行技能配对。动态规划(Dynamic Programming, DP)资源分配算法以作业所需的技能 s 为阶段, 状态变量 s_s 为第 s 阶段剩余的未分配资源, 决策变量 x_s 为分配给 s 技能的资源组合, 状态转移方程如式(31)所示, 表示 s_s 阶段剩余资源在分配 x_s 资源给第 s 个技能后剩余资源为下一阶段的状态。阶段指标函数 $v_s(s_s, x_s)$ 表示 x_s 决策下资源在技能 s 下的效用值 U 。最优值函数 $f_s(s_s)$ 表示将 s_s 个资源分配给第 s 至该作业所需的最后一种技能 k 的最大效用值。逆向动态规划递推公式如式(32):

$$s_{s+1} = s_s - x_s \quad (31)$$

$$\begin{cases} f_{k+1}(s_{k+1}) = 0 \\ f_s(s_s) = \max(v_s(s_s, x_s) + f_{s+1}(s_{s+1})), \\ s = k + 1, k, \dots, 1 \end{cases} \quad (32)$$

效用值 U 定义为加权效率增长量, 计算如式(33)所示:

$$U = \omega_1 \Delta \eta_1 + \omega_2 \Delta \eta_2 \quad (33)$$

式中: ω_1 和 ω_2 分别为有培训目标的效率增长量权重和无培训目标的效率增长量权重; $\Delta \eta_1$ 和 $\Delta \eta_2$ 分别为与培训目标相符的效率增长量与无培训目标的效率增长量。有无培训目标由资源目标编码判断, 若资源在当前所分配技能的效率尚未达到培训目标, 则归为有培训目标的效率增长。通过 DP 得到最大效用值后可反推出作业中资源和技能的配对关系。

2.2 聚类插空调度计划生成机制

Myszkowski 等^[20] 在 MS-RCPSP 求解中提出 Task-Greedy SGS (TG SGS) 进行优先级分配和调度, 本文在 TG SGS 基础上, 嵌入基于 U-K-means 的优先级分配算法和基于插空的调度计划生成机制, 提出聚类插空调度计划生成机制 (Task-Cluster-Calking SGS, TCC SGS), TCC SGS 算法流程如下所示:

步骤 1: 将项目中 n 项作业与 m 个资源的分配情况用 $n \times m$ 独热编码表示。

步骤 2: 以 $n \times m$ 独热编码为输入, 使用 U-K-means 算法将资源使用情况相似的作业聚类, 输出 $n \times c$ 的隶属度矩阵 b , 其中 c 为簇的个数。其中 U-K-means 算法具体描述见 2.2.1 节。

步骤 3: 计算每个簇内作业的平均最晚开始时间

(latest start time, LST), 并将各个簇根据 LST 按升序排序, 同一个簇内的作业按索引排序, 得到所有作业优先级。

步骤4: 根据有无紧后任务将所有作业划分为有紧后任务作业和无紧后任务作业, 同时将2类中的作业按照步骤3获得的优先级进行排序。

步骤5: 基于资源插空算法对有紧后任务作业按优先级、时序约束和资源约束进行调度。其中资源插空算法具体描述见2.2.2节。

步骤6: 基于资源插空算法对无紧后任务作业按优先级、时序约束和资源约束进行调度。

2.2.1 基于聚类(U-K-means)的优先级分配算法

TG SGS关键思想在于有紧后任务作业相比无紧后任务作业更可能处在关键路径上, 该方法将作业分成有紧后任务作业和无紧后任务作业2类, 并优先调度有紧后任务作业。但该算法对类内作业按索引赋优先级, 未使用作业的资源使用数据, 容易因部分资源任务堆积导致项目完工时间增加。本文在TG SGS基础上, 在2类作业的内部根据各作业资源利用情况相似性, 利用聚类方式对2类作业中的作业进一步分配优先级, 减少资源竞争。

K-means是一种被广泛使用的聚类算法, 但该算法需要手动设置聚类数量和聚类初始点, Sinaga等^[21]为K-means算法构建了一个无监督学习模式,

提出无监督K-means算法(Unsupervised K-means clustering algorithm, U-K-means), 该算法可以在没有参数选择的情况下免于初始化, 并能够同时找到最优数量的聚类。U-K-means目标函数如式(34)所示:

$$\min J_{U-K-means}(\mathbf{b}, A, \alpha) = \sum_{i=1}^n \sum_{j=1}^c b_{ij} \|x_i - a_j\|^2 - \beta_c n \sum_{j=1}^c \alpha_j \ln \alpha_j - \gamma_c \sum_{i=1}^n \sum_{j=1}^c b_{ij} \ln \alpha_j \quad (34)$$

式中: \mathbf{b} 为数据隶属度; A 为聚类中心; α 为数据点隶属于该簇的概率; β_c 和 γ_c 分别为比例因子和学习率, 随着迭代过程更新。该目标函数由三部分组成。第1部分是数据点到聚类中心的距离平方和, 第2部分和第3部分基于信息熵理论提出, 用于衡量聚类中心分布情况。通过最小化目标函数, U-K-means算法可以自动确定最优聚类中心和聚类数量。U-K-means算法通过迭代不断更新聚类中心 A 和数据点的隶属度 \mathbf{b} , 以最小化目标函数。如图5所示, 以作业数 $n=8$ 、资源数 $m=8$ 为例, 其中J1和J10为虚拟作业, 将资源分配情况以 8×8 独热编码输入, 经过U-K-means计算后输出 8×3 隶属度矩阵, 表示8个作业被分成3个簇。分到同一个簇的作业资源用同种颜色阴影表示, 簇内作业的资源使用情况相似。

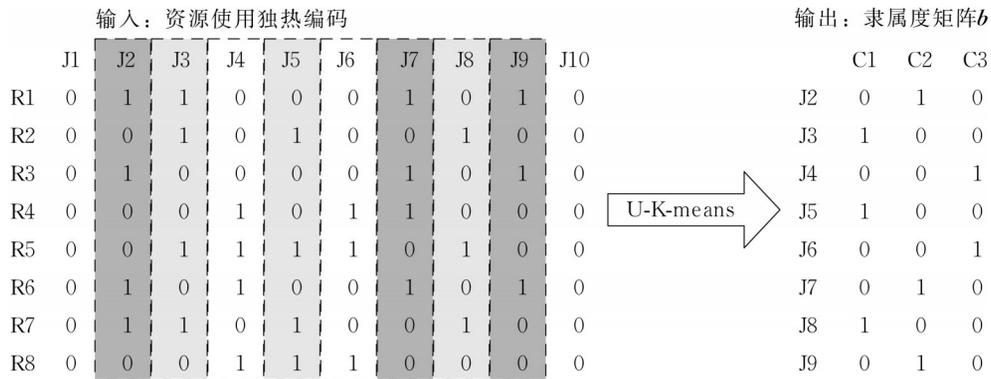


图5 U-K-means示意

Fig. 5 Schematic diagram of U-K-means

簇内作业存在资源竞争, 簇与簇之间资源竞争则相对较小。为减少资源竞争对完工时长的影响, 需对各簇赋优先级, 使较为紧急的簇内作业尽早开始。采用平均LST对各簇赋优先级, 同一簇内作业按索引排序。

2.2.2 资源插空机制

优先级排序并不充分考虑资源冲突情况, 按照优先级顺序调度作业可能导致资源闲置。引入资源

插空机制, 让可执行作业跳出优先级限制, 灵活利用空闲资源提前执行, 减少资源浪费。资源插空的调度计划生成机制伪代码如算法1所示:

```

1  输入: 待调度作业优先级排序 pry_job、时序约束、
n×m的资源使用集合 job_res、作业时长 job_dur
2  初始化 时间指针 t=0, 当前正在执行的作业集合
set_doing, 正在作业的资源集合 set_working
3  while pry_job 非空 do
4      for current_job in pry_job do

```

```

5         if current_job 紧前作业均完成 & 根据
set_working所需资源均为空闲 do
6             set_working.add(job_res(current_
job))
7             set_doing.add(set_working,
current_job)
8             current_job的开始时间 = t
9             current_job的结束时间 = t + job_dur
(current_job)
10        end if
11    end for
12    更新时间指针 t = min(set_doing 中作业的结束
时间)
13    set_doing.remove(在 t 时刻完成的作业)
14    set_working.remove(在 t 时刻完成的作业所占
用的资源)
15    pry_job.remove(在 t 时刻完成的作业)
16 end while
17 输出: 所有作业开始时刻和完成时刻

```

第4至11行根据 t 时刻的资源空闲情况判断是否允许其他作业提前开始,若提前开始则对当前正在执行的作业集合和正在作业的资源集合进行更新,并计算插入作业的开始和结束时间。第12至15行表示当无法再插入新的作业后,跳出循环,更新时间指针到正在作业集合中最先完工的作业的结束时刻,释放该作业占用的资源,并从待调度作业和正在执行作业集合中移除该作业。到所有作业均完成调

度后停止循环,输出所有作业开始和结束时刻。

对图5所示算例,TCC SGS和TG SGS得到的甘特图如图6 a和b所示,利用执行作业6时的空闲资源提前开始执行优先级较低的作业2,在执行作业5时利用空闲资源提前开始作业7,总完工时长为25,而TG SGS按照索引依次开始各作业,在进行作业2时没有利用闲置资源提前开始作业6,造成时间浪费,总作业时长增加到27。

3 数值实验

目前尚无考虑知识扩散的MS-RCPSP标准测试集,因此采用对PSPLIB中现有算例进行改造的方式。改造方式如下:①作业时长,时序约束来自PSPLIB。②技能数量为4,每项技能所需资源数量不少于2人,每个技能所需的标准效率服从 $N(0.6, 0.08^2)$ 的正态分布。③考虑企业中存在通用工人与专业工人的实际情况,根据资源效率的平均值与极大值两者的算术平均从高到低将资源划分成4级,初始资源各等级数量占比分别为10%、20%、30%、40%,目标资源结构各等级占比分别为20%、40%、40%、0。④考虑作业数量 $n = \{10, 14, 20\}$ 、资源数量 $m = \{10, 15, 20\}$ 共9种问题规模进行测试,每种问题规模均设计5个算例,所有数值实验均使用改造后的数据集。

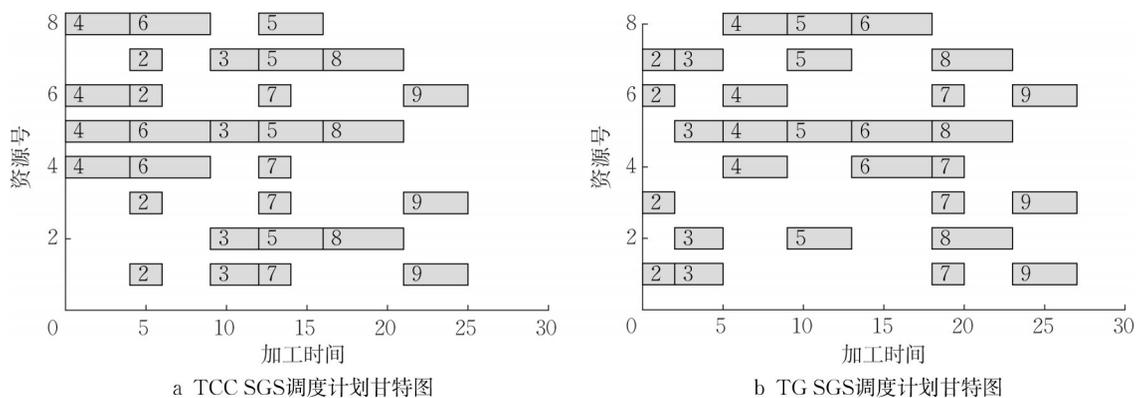


图6 TCC SGS与TG SGS调度计划甘特图对比

Fig. 6 Comparison of Gantt charts for TCC SGS and TG SGS scheduling plans

由于本问题难以获得真实 Pareto 前沿,因此选定多样性指标纯粹距离(Pure Distance, PD)^[22]、收敛性指标高维空间体积(Hyperarea, H)^[23]和综合性指标超体积指标(Hypervolume, HV)^[24]3项指标,以全面评估算法的性能,GAP定义为当前算法的性能指标与表现最好的算法之间的差距比例。

实验平台使用 Windows 11 64 位操作系统, CPU 为 Intel Core i7-12700H 2.30 GHz,内存 16.0 GB。开发环境为 MATLAB R2021b。

3.1 MOEA/D-NG 算法性能评价

将本文提出的 MOEA/D-NG 算法(A1)与标准 MOEA/D 算法(A2)以及广泛应用的 MOEA/D-DE

算法(A3)进行比较,3种算法均使用DP资源分配算法与TCC SGS,对比结果如表3所示,表中 $n \times m$ 表示算例规模。MOEA/D-NG算法在不同规模的算例中均有较优的表现,其多样性指标PD相较MOEA/D提高1.1%~21.7%,较MOEA/D-DE提

高2.2%~15.8%。收敛性指标H较MOEA/D降低5.5%~31.4%,较MOEA/D-DE降低1.3%~15.4%。综合性指标HV较MOEA/D提高9.7%~39.3%,较MOEA/D-DE提高4.8%~13.3%。

图7为3种算法下各算例CPU平均运行时间

表3 算法性能比较

Tab. 3 Comparison of algorithm performance

$n \times m$	算法编号	PD		H		HV	
		AVG	GAP/%	AVG	GAP/%	AVG	GAP/%
10×10	A1	485.1	-0.7	123 274.8	0	38 870.8	0
	A2	488.6	0	143 829.4	16.7	32 447.0	-16.5
	A3	466.8	-4.5	132 769.4	7.7	35 356.2	-9.0
10×15	A1	920.7	0	358 160.2	0	50 928.2	0
	A2	801.8	-12.9	470 634.4	31.4	41 104.0	-19.3
	A3	822.4	-10.7	413 225.3	15.4	44 175.8	-13.3
10×20	A1	1295.3	0	408 882.4	0.0	92164.6	0.0
	A2	1014.2	-21.7	476 408.8	16.5	79409.4	-13.8
	A3	1107.4	-14.5	464 552.3	13.6	87284.6	-5.3
14×10	A1	458.0	0	139 338.2	0	55 782.6	0
	A2	432.2	-5.6	156 159.3	12.1	50 365.4	-9.7
	A3	426.6	-6.9	145 670.6	4.5	53 100.2	-4.8
14×15	A1	618.1	0	342 402.0	0	81 411.3	0
	A2	611.6	-1.1	361 161.2	5.5	49 424.4	-39.3
	A3	600.9	-2.8	346 766.6	1.3	71 370.5	-12.3
14×20	A1	836.8	0	525 743.4	0.0	62 188.2	0
	A2	777.3	-7.1	593 693.4	12.9	53 016.2	-14.7
	A3	818.4	-2.2	565 573.4	7.6	58 042.6	-6.7
20×10	A1	450.9	0	137 732.6	0	36 311.8	0
	A2	395.2	-12.4	17 9847.8	30.6	30 037.2	-17.3
	A3	410.2	-9.0	149 197.4	8.3	33 737.8	-7.1
20×15	A1	859.5	0	434 772.4	0	118 492.8	0
	A2	779.5	-9.3	558 082.8	28.4	100 918.4	-14.8
	A3	733.7	-14.6	460 908.8	6.0	108 180.3	-8.7
20×20	A1	851.2	0	511 553.2	0	60 348.2	0
	A2	693.8	-18.5	545 736.8	6.7	47 531.8	-21.2
	A3	716.4	-15.8	554 977.8	8.5	54 848.2	-9.1

t_{CPU} ,由于不同规模算例运行时间差异较大,为方便在不同规模中比较3种算法运行时间的差异,对 t_{CPU} 做底数为10的对数处理,若不取对数,小规模算例的柱形很短,难以对比3种算法的运行时间差异。由于本文提出的MOEA/D-NG需要计算父代与邻域内个体的相似度值,因此在不同规模算例中,MOEA/D-NG的CPU时间较MOEA/D长约8.1%,较MOEA/D-DE长约16.1%。

3.2 资源分配算法性能比较与评价

将本文提出的基于DP的资源分配算法与其他广泛应用的算法进行对比,其中D2表示基于排序的资源分配算法,即根据每种资源分配情况的效用值高低进行排序,李斌等[8]采用该算法,D3表示基于搜索的资源分配算法,即利用随机搜索的方式迭代选择资源分配情况,陆志强等[7]采用该算法。将这几种

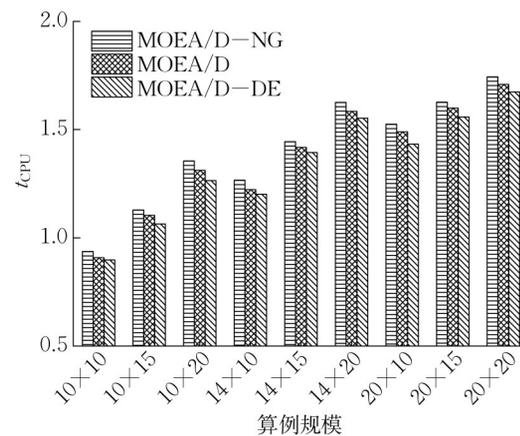


图7 不同作业规模下3种算法的CPU运行时间

Fig. 7 CPU running time of the three algorithms across different example sizes

资源分配算法进行比较,比较结果如表4所示。

在 $n=10$ 的小规模算例中,由于解空间有限性,基于搜索的算法D3表现最为出色,DP资源分配算法的综合性指标HV与D3差距在4%以内,性能接近;基于排序的算法D2性能最差。而在 $n=14$ 和 $n=20$ 规模算例中,DP算法综合性能最佳,解空间的扩大导致基于搜索的算法D3性能下降,收敛性指标H相较于DP算法增加21.9%~69.3%,综合性指标HV相较DP算法下降2.2%~27.7%。在 $n=20$ 规模算例中,基于搜索的算法D3在相同次数迭代下难以收敛,表现差于另外2种算法。为进一步衡量3种算法下资源培训的效率,定义有效培训占比 $\pi=(\eta_T-\eta_0)/(\eta_F-\eta_0)$,其中 η_T 表示目标资源效率值, η_0 表示资源初始效率值, η_F 表示培训完成后资源实际效率值。选取3种算法在每个规模下其中一个算例的5次实验得到的5组非支配解,绘制有效培训占比箱型图如图8所示。

DP算法具有良好的性能表现。其中在 $n=14$ 和 $n=20$ 的算例中,DP算法的 π 均值高于排序算法和基于搜索的算法。在 $n=10$ 的小规模算例中,虽

然DP的 π 均值低于基于搜索的算法,但二者之间差距较小。通过箱形图的四分位范围分析可知,排序算法与DP算法的数据分布紧凑,而基于搜索的算法则呈现较大的数据分散程度,表明DP算法和排序算法在鲁棒性方面表现较为出色。

3.3 优先级分配和调度算法性能比较与评价

为研究本文所提出的优先级分配和调度算法性能,将TCC SGS记为S1,将Myszkowski等^[20]提出的TG SGS记为S2,将没有使用U-K-Means聚类但使用了资源插空规则的TG SGS(TGC SGS)记为S3。通过测试,得到如表5所示的对比结果。

本文提出的TCC SGS在收敛性方面具有优越性,且随着算例规模的增大收敛性优势更加明显。在 $n=10$ 规模的算例中,TCC SGS的H比TGC SGS降低0.2%~1.3%,而在 $n=14$ 和 $n=20$ 规模的算例中,其H平均降幅则分别达到2.3%和6.3%。在综合性指标HV方面,在3个不同规模的算例中,TCC SGS比TGC SGS分别提高1.7%、2.9%和5.0%。TG SGS收敛性最差。由于作业完工时长与拖期成本直接相关,可以通过平均每个项

表4 资源分配算法实验结果

Tab. 4 Experimental results of resource allocation algorithm

$n \times m$	算法编号	PD		H		HV	
		AVG	GAP/%	AVG	GAP/%	AVG	GAP/%
10×10	D1	485.1	-2.3	123 274.8	2.5	38 870.8	-3.6
	D2	274.6	-44.7	173 836.4	44.5	30 361.4	-24.7
	D3	496.3	0	120 301.2	0	40 324.4	0
10×15	D1	920.7	0	358 160.2	0	50 928.2	-0.5
	D2	508.8	-44.7	485 318.8	35.5	35 611.2	-30.4
	D3	867.5	-5.8	359 532.4	0.4	51 202.2	0
10×20	D1	1295.3	0	408 882.4	3.3	92 164.6	0
	D2	1022.5	-21.1	614 352.2	55.2	81 941.6	-11.1
	D3	1102.4	-14.9	395 832.5	0	91 546.5	-0.7
14×10	D1	458.0	0	139 338.2	0	55 782.6	0
	D2	406.4	-11.3	170 537.2	22.4	41 620.2	-25.4
	D3	432.4	-5.6	169 862.2	21.9	40 356.3	-27.7
14×15	D1	618.1	0	342 402.0	0	81 411.3	0
	D2	565.4	-8.5	597 089.8	74.4	75 668.4	-7.1
	D3	598.3	-3.2	576 323.6	68.3	79 653.2	-2.2
14×20	D1	836.8	0	525 743.4	0	62 188.2	0
	D2	827.6	-1.1	636 842.2	21.1	53 127.7	-14.6
	D3	836.7	0.0	645 632.7	22.8	51 237.8	-17.6
20×10	D1	450.9	-4.6	237 732.6	0	36 311.8	0
	D2	472.7	0	338 412.5	42.4	32 451.6	-10.6
	D3	402.3	-14.9	359 682.5	51.3	29 649.8	-18.3
20×15	D1	859.5	-13.3	434 772.4	0	118 492.8	0
	D2	991.0	0	705 263.4	62.2	113 580.8	-4.1
	D3	865.7	-12.6	735 893.4	69.3	108 932.6	-8.1
20×20	D1	851.2	0	511 553.2	0	60 348.2	0
	D2	768.8	-9.7	733 876.4	43.5	51 430.6	-14.8
	D3	782.4	-8.1	785 337.2	53.5	50 237.7	-16.8

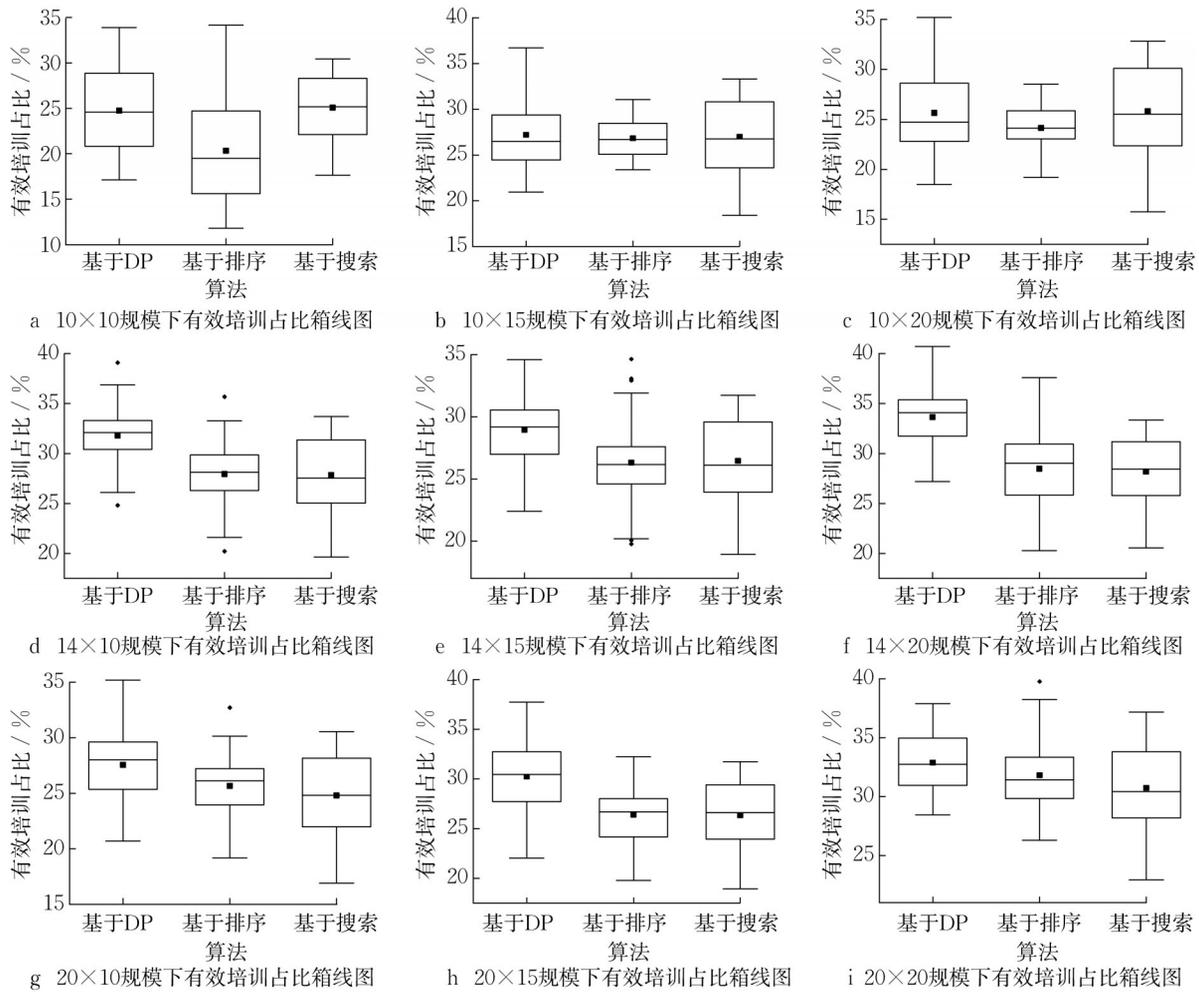


图 8 各算法有效培训占比箱形图

Fig. 8 Box diagram of effective training proportion of each algorithm

表 5 优先级分配和调度算法实验结果

Tab. 5 Experimental results of priority allocation and scheduling algorithm

$n \times m$	算法编号	PD		H		HV		$n \times m$	算法编号	PD		H		HV	
		AVG	GAP/%	AVG	GAP/%	AVG	GAP/%			AVG	GAP/%	AVG	GAP/%	AVG	GAP/%
10×10	S1	485.1	0	123 274.8	0	38 870.8	0	14×20	S1	836.8	-5.0	525 743.4	0	62 188.2	0
	S2	482.7	-0.5	124 733.2	1.2	38 583.4	-0.7		S2	880.4	0	558 999.2	6.3	58 066.3	-6.6
	S3	475.6	-2.0	123 592.4	0.3	38 768.5	-0.3		S3	856.6	-2.7	540 782.3	2.9	59 867.6	-3.7
10×15	S1	920.7	0	358 160.2	0	50 928.2	0	20×10	S1	450.9	-0.3	137 732.6	0	38 311.8	0
	S2	858.3	-6.8	363 272.2	1.4	48 432.6	-4.9		S2	420.8	-7.0	152 095.8	10.4	35 654.8	-6.9
	S3	900.4	-2.2	362 880.5	1.3	49 556.6	-2.7		S3	452.3	0	147 602.3	7.2	36 254.7	-5.4
10×20	S1	1295.3	-4.2	408 882.4	0	92 164.6	0	20×15	S1	859.5	-4.3	434 772.4	0	12 0492.8	0
	S2	1155.9	-14.5	420 194.8	2.8	87 444.2	-5.1		S2	898.2	0	454 155.3	4.5	11 4651.2	-4.8
	S3	1352.6	0	410 211.1	0.3	90 356.7	-2.0		S3	834.5	-7.1	452 302.5	4.0	11 5234.2	-4.4
14×10	S1	458.0	-2.2	139 338.2	0	55 782.6	0	20×20	S1	851.2	-1.4	511 553.2	0	60 348.2	0
	S2	468.4	0	145 983.2	4.8	53 408.8	-4.3		S2	821.6	-4.8	565 766.8	10.6	55 654.8	-7.8
	S3	449.1	-4.1	142 576.8	2.3	54 576.8	-2.2		S3	863.4	0	550 236.7	7.6	57 266.1	-5.1
14×15	S1	618.1	-6.6	342 402.0	0	81 411.3	0								
	S2	653.7	-1.2	350 852.4	2.5	78 920.2	-3.1								
	S3	661.7	0	348 954.8	1.9	79 216.3	-2.7								

目拖期成本在总成本占比研究调度算法对成本的优化效果。选取3种算法在每种规模其中一个算例的

5次实验得到的5组非支配解,绘制平均项目拖期成本占比箱型图,如图9所示。

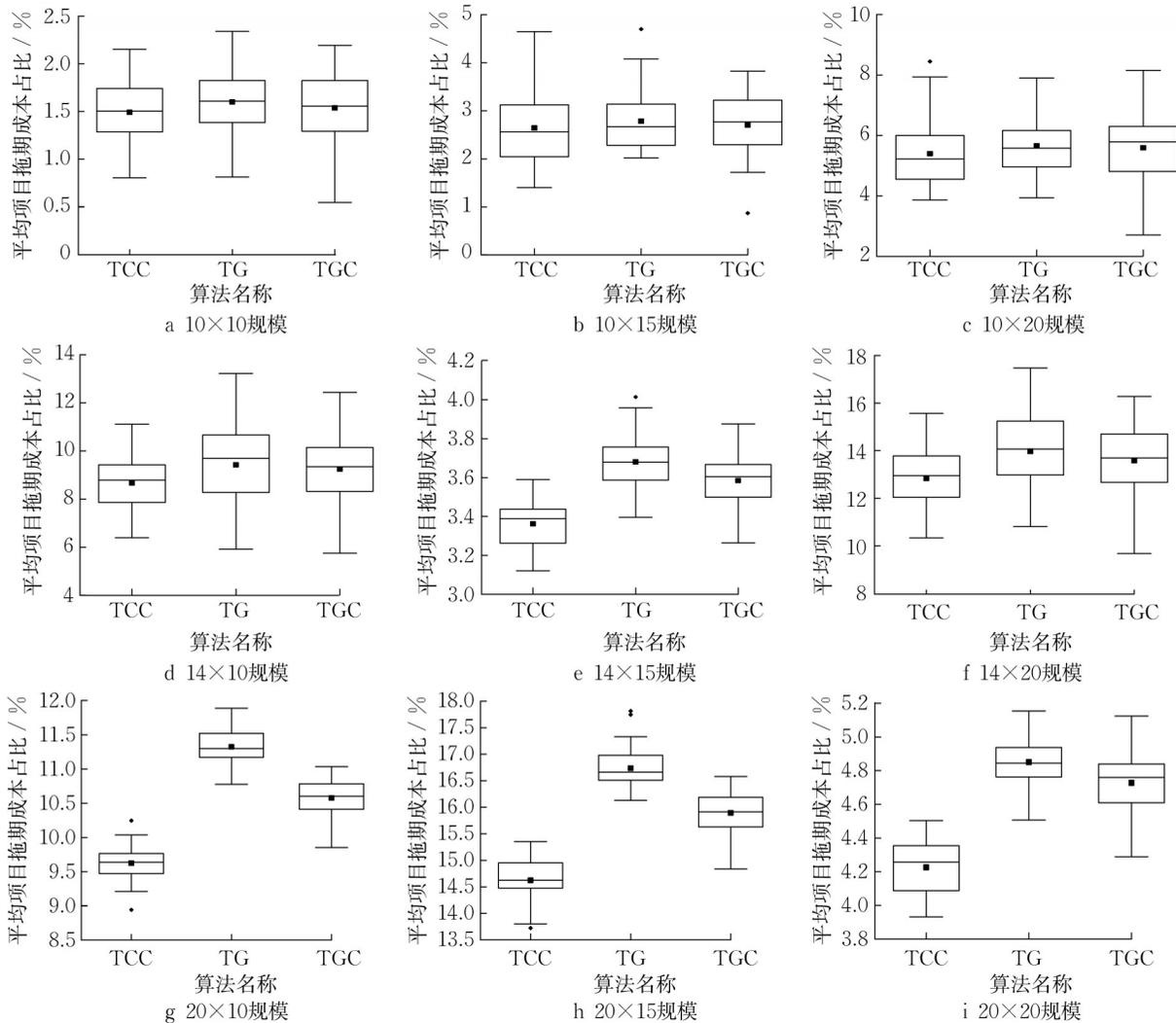


图9 各算法平均项目拖期成本占比箱形图

Fig. 9 Box plot showing the average project delay cost proportion for each algorithm

在所有9个算例中,TCC SGS均具有最好的性能表现,其均值和中位数在3个算法中均为最小,并且随着算例规模的增大,其优势更加显著。在 $n=10$ 规模算例中,使用TCC SGS得到的结果平均项目拖期成本比使用TGC SGS和TG SGS分别降低2.9%和4.5%;在 $n=14$ 规模算例中,降幅升至5.9%和8.3%;在 $n=20$ 规模算例中,降幅为9.1%和14.5%,表明TCC SGS能够有效缩短作业时长。

4 结语

考虑生产线中的知识扩散现象,以最短化培训时长与最小化平均项目成本为目标,建立考虑团队

学习效应的多技能资源受限项目调度模型。采用社交网络模型描述知识扩散,提出基于动态规划的资源技能分配算法与基于聚类的优先级分配和基于资源插空的调度计划生成机制,并将其嵌入基于小生境父代选择的MOEA/D-NG算法求解该问题。利用数值实验结果验证算法有效性。

本文仅考虑师徒一对一培训这一种知识扩散形式,后续还可以考虑离线培训、集体培训等其他形式。此外,还可以在一对一培训中,进一步根据被培训资源的效率值高低分配培训时长,提高知识扩散效率。

作者贡献声明:

陆志强:提出研究选题,设计研究思路和论文框架。

马新一:设计研究思路,完成实验设计并实施,分析数据,撰写论文。

参考文献:

- [1] BISKUP D. Single-machine scheduling with learning considerations [J]. *European Journal of Operational Research*, 1999, 115(1): 173.
- [2] CHEN J C, CHEN Y Y, CHEN T L, *et al.* Multi-project scheduling with multi-skilled workforce assignment considering uncertainty and learning effect for large-scale equipment manufacturer [J]. *Computers & Industrial Engineering*, 2022, 169: 108240.
- [3] 胡金昌,刘紫薇,马文凯,等.考虑学习效应的单人作业车间多目标调度算法[J].*计算机集成制造系统*,2021,27(5):1361.
HU Jinchang, LIU Ziwei, MA Wenkai, *et al.* Multi-objective scheduling algorithm for one worker job shop considering learning effect [J]. *Computer Integrated Manufacturing Systems*, 2021, 27(5): 1361.
- [4] JEMMALI M, HIDRI L. Bounding schemes for the parallel machine scheduling problem with DeJong's learning effect [J]. *Journal of Parallel and Distributed Computing*, 2021, 156: 101.
- [5] HEUSER P, TAUER B. Single-machine scheduling with product category-based learning and forgetting effects [J]. *Omega*, 2023, 115: 102786.
- [6] ZHANG L, DENG Q, LIN R, *et al.* A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect [J]. *Expert Systems with Applications*, 2021, 175: 114843.
- [7] 陆志强,陈丞.考虑柔性资源技能进化的多项目组合调度问题[J].*湖南大学学报(自然科学版)*,2021,48(10):10.
LU Zhiqiang, CHEN Cheng. Multi-project portfolio scheduling problem considering skill evolution of flexible resource [J], *Journal of Hunan University (Natural Sciences)*, 2021, 48(10):10.
- [8] 李斌,谢乃明.考虑团队交互学习效应的单工作组可中断任务调度模型[J].*计算机集成制造系统*,2022,28(1):161.
LI Bin, XIE Naiming. Single workgroup interruptible task scheduling model considering team interactive learning effect [J]. *Computer Integrated Manufacturing Systems*, 2022, 28(1): 161.
- [9] HOSSEINIAN A H, BARADARAN V, BASHIRI M. Modeling of the time-dependent multi-skilled RCPSP considering learning effect: An evolutionary solution approach [J]. *Journal of Modelling in Management*, 2019, 14(2): 521.
- [10] 李柏洲,曾经伟,董恒敏,等.实践社群知识扩散模型及仿真研究[J].*运筹与管理*,2021,30(1):130.
LI Bozhou, ZENG Jingwei, DONG Hengmin, *et al.* A model for knowledge diffusion in communities of practice and its simulation [J]. *Operations Research and Management Science*, 2021, 30(1): 130.
- [11] 余谦,朱锐芳.多维邻近创新网络中知识扩散模型与仿真研究[J].*情报科学*,2020,38(5):65.
YU Qian, ZHU Ruifang. Knowledge Diffusion Model and Simulation in Multi-dimensional Proximity Innovation Network [J]. *Information Science*, 2020, 38(5): 65.
- [12] LI N, HUANG Q, Ge X, *et al.* A review of the research progress of social network structure [J]. *Complexity*, 2021, 2021: 1.
- [13] ZHENG W, PAN H, SUN C. A friendship-based altruistic incentive knowledge diffusion model in social networks [J]. *Information Sciences*, 2019, 491: 138.
- [14] SONG L, MA Y. Evaluating tacit knowledge diffusion with algebra matrix algorithm based social networks [J]. *Applied Mathematics and Computation*, 2022, 428: 127125.
- [15] CREMONINI M. Introducing serendipity in a social network model of knowledge diffusion [J]. *Chaos, Solitons & Fractals*, 2016, 90: 64.
- [16] 温跃杰,李焯坤,张泓翊,等.基于模型的人力资源管理系统工程:以某企业为例[J].*系统工程学报*,2023,38(2):225.
WEN Yuejie, LI Zhaokun, ZHANG Hongyi. Model-based human resource systems engineering: an enterprise case example [J]. *Journal of Systems Engineering*, 2023, 38(2): 225.
- [17] SHARMA G G, STOL K J. Exploring onboarding success, organizational fit, and turnover intention of software professionals [J]. *Journal of Systems and Software*, 2020, 159: 110442.
- [18] CORREIA I, LOURENCO L L, Saldanha-da-Gama F. Project scheduling with flexible resources: formulation and inequalities [J]. *OR Spectrum*, 2012, 34: 635.
- [19] XU E, LI Y, LIU Y, *et al.* Energy saving scheduling strategy for job shop under TOU and tiered electricity price [J]. *Alexandria Engineering Journal*, 2022, 61(1): 459.
- [20] MYSZKOWSKI P B, LASZCZYK M. Diversity based selection for many-objective evolutionary optimisation problems with constraints [J]. *Information Sciences*, 2021, 546: 665.
- [21] SINAGA K P, YANG M S. Unsupervised K-means clustering algorithm [J]. *IEEE Access*, 2020, 8: 80716.
- [22] WANG H, JIN Y, YAO X. Diversity assessment in many-objective optimization [J]. *Cybernetics IEEE Transactions on*, 2017, 47(6):1510.
- [23] ZITZLER E, THIELE L. Multiobjective optimization using evolutionary algorithms—a comparative case study [C]// *International conference on parallel problem solving from nature*. Berlin: Springer Berlin Heidelberg, 1998: 292-301.
- [24] KNOWLES J, CORNE D. Properties of an adaptive archiving algorithm for storing nondominated vectors [J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 100.