

控制与决策

Control and Decision

基于动态资源权重的多技能项目调度启发式算法

胡振涛, 崔南方, 张艳, 胡雪君

引用本文:

胡振涛, 崔南方, 张艳, 等. 基于动态资源权重的多技能项目调度启发式算法[J]. *控制与决策*, 2021, 36(10): 2553–2561.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.0070>

您可能感兴趣的其他文章

Articles you may be interested in

面向建材装备集团制造的分布式多项目资源调度

Distributed multi-project resource scheduling oriented to manufacturing of building materials equipment group

控制与决策. 2021, 36(9): 2133–2142 <https://doi.org/10.13195/j.kzyjc.2019.1802>

基于参数自适应蚁群算法的高速列车行车调度优化

Optimization of high-speed train operation scheduling based on parameter adaptive improved ant colony algorithm

控制与决策. 2021, 36(7): 1581–1591 <https://doi.org/10.13195/j.kzyjc.2020.0992>

超启发式交叉熵算法求解模糊分布式流水线绿色调度问题

Hyper-heuristic cross-entropy algorithm for green distributed permutation flow-shop scheduling problem with fuzzy processing time

控制与决策. 2021, 36(6): 1387–1396 <https://doi.org/10.13195/j.kzyjc.2019.1681>

基于地标特征和元学习方法推荐最适用优化算法

Recommending best suitable metaheuristic based on landmarking feature and meta-learning approach

控制与决策. 2021, 36(5): 1223–1231 <https://doi.org/10.13195/j.kzyjc.2019.0993>

基于改进NSGA-II算法求解多目标资源受限项目调度问题

An improved NSGA-II algorithm for multi-objective resource-constrained project scheduling problem

控制与决策. 2021, 36(3): 669–676 <https://doi.org/10.13195/j.kzyjc.2019.0906>

基于动态资源权重的多技能项目调度启发式算法

胡振涛¹, 崔南方^{1†}, 张艳¹, 胡雪君²

(1. 华中科技大学 管理学院, 武汉 430074; 2. 湖南大学 工商管理学院, 长沙 410082)

摘要: 多技能资源受限项目调度问题中, 一个资源可同时具备多项技能, 相较于传统的单技能项目调度, 其资源分配对调度计划的工期影响程度更大, 因此在对多技能项目进行排程时更加重视资源的分配. 基于此, 从资源视角提出一种启发式算法求解工期最短的调度计划. 算法以并行调度为主体, 并设计一种动态资源权重计算方法, 在每一决策点, 首先采用二分图最大匹配法确定当前可排活动集, 而后将动态资源权重值作为调度过程中资源分配的依据, 其核心思想是将资源灵活度高、对后续活动影响大的资源留置, 以待下一决策点调用. 最后, 为验证算法有效性, 对不同参数设置下的算例进行实验, 结果表明, 相较于随机资源分配算法和静态资源权重算法, 新算法具有明显优势.

关键词: 项目调度; 多技能资源; 并行调度; 二分图; 资源权重; 启发式算法

中图分类号: F224 文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0070

开放科学(资源服务)标识码(OSID):



引用格式: 胡振涛, 崔南方, 张艳, 等. 基于动态资源权重的多技能项目调度启发式算法[J]. 控制与决策, 2021, 36(10): 2553-2561.

Dynamic resource priority-based heuristics for multi-skill resource constrained project scheduling problem

HU Zhen-tao¹, CUI Nan-fang^{1†}, ZHANG Yan¹, HU Xue-jun²

(1. School of Management, Huazhong University of Science and Technology, Wuhan 430074, China; 2. Business School, Hunan University, Changsha 410082, China)

Abstract: The resources in the multi-skilled project scheduling problem(MSPSP) are assumed to be multi-skilled. Compared with the single-skilled project scheduling problem, the resource allocation has more influence on the makespan in the MSPSP. Therefore, more attention should be paid to resource allocation when scheduling a multi-skilled project. In this paper, we propose a heuristic algorithm to solve the MSPSP from the perspective of resource allocation. The algorithm uses a maximum matching method of bipartite graphs to select schedulable activities and develops a dynamic resource priority to allocate resources based on the parallel scheduling scheme. The core idea of the algorithm is to retain resources which are more flexible or more important to the subsequent activities for the next decision time. In order to verify the effectiveness of the proposed heuristics, a series of computational tests are conducted. The results show that the proposed algorithm has a significant advantage over the algorithms based on random or static resource priority.

Keywords: project scheduling; multi-skill resource; parallel scheduling scheme; bipartite graph; resource priority; heuristic algorithm

0 引言

资源受限项目调度问题(resource constrained project scheduling problem, RCPSP)以最小化项目工期^[1-2]、最大化净现值^[3-4]及资源均衡^[5-6]为目标, 在确定或不确定环境下构建一个满足资源和工序约束的调度计划. 由基础RCPSP可以延伸出若干项目调度问题, 如多模式的RCPSP^[7]、活动可中断的RCPSP^[8]、

柔性资源约束的RCPSP^[9]等, 其中多技能项目调度问题(multi-skilled project scheduling problem, MSPSP)成为近些年研究的热点.

MSPSP与传统RCPSP的主要区别在于, 项目资源可以具备多项技能, 能以不同的技能形式参与到不同的活动中去, 因此资源与活动会存在多种对应关系. 在进行多技能项目调度时, 不仅要决定活动的开

收稿日期: 2020-01-15; 修回日期: 2020-05-14.

基金项目: 国家自然科学基金项目(71971094); 国家自然科学基金项目(71701067); 湖南省自然科学基金项目(2019JJ50039).

†通讯作者. E-mail: nfcui@mail.hust.edu.cn.

始时间及资源在活动间的分配,还要决定资源以何种技能参与其中,解的维度增加,问题变得更加复杂。

以服装生产为例,物料要依次流经以下机器:捆条机、绷缝机、锁边机、打钮机、熨烫机等,每种机器都配有若干操作人员,若将服装厂的一个订单视为一个项目,则各个机器的运作便是项目中的各个活动,这些操作工人可看作是项目中的可更新资源。为了尽快完成订单,如何规划各机器的开工时间,以及对这些工人如何调配,便成了一个RCPSP问题。从传统的RCPSP视角,每个工人只具有操作某一种机器的技能,因此在满足工人数量约束的前提下,一旦规划好机器的开工时间,对应机器的空闲工人即可上岗,人力资源分配方案也随之确定。而在现实中,由于这些机器的操作并不复杂,多数工人都能掌握不止一种机器的操作方法,将工人这种具备多技能的特征纳入RCPSP,就形成了MSPSP问题。

不同于传统RCPSP的是,在对MSPSP求解时,由于这些多技能工人可以在不同的机器间“转换”,即使已经确定了机器的开工时间,工人的分配方案依旧是不确定的。一旦指派某个工人到某个机器岗位之后,该工人所具备的其他技能便被“抢占”,其他机器的可用工人数量也随之减少1,各机器(活动)的可用操作工人(资源)像这样交互作用会最终影响项目的工期。

目前关于MSPSP的研究多以求解最短项目工期为目标,少部分文献考虑了工期以外的目标函数,如最大化项目的期望收益^[10]、最小化资源投入成本^[11]、最小化人力资源数量^[12]等。在约束条件方面,Peteghem等^[13]考虑了多模式的MSPSP,Javanmard等^[14]研究了活动可中断情况下的MSPSP,Bellenguez等^[15]则假设资源对不同技能的熟练程度不同,其转换效率也不同。

由于MSPSP是NP-hard的,有学者通过建立混合整数规划模型并对模型的不同约束进行松弛,求解了工期的上界^[7],更多的学者则设计了元启发式算法予以求解,如蚁群算法^[16]、混沌粒子群算法^[17]、杂草入侵算法^[18]等。这些算法在实验环境下可以用较短的时间求得满意解,但是在现实项目中往往活动数量更多,工序关系更复杂,且项目环境多变,项目管理者需要更为简洁易行且能快速求解的算法。

基于规则的启发式算法是一种通过设计优先规则对问题进行迭代求解的算法,具有过程简单易懂、求解速度快等特点,在现实项目中应用广泛。针对MSPSP,Almeida等^[19]提出了基于资源规则、活动

规则的启发式算法,通过与CPLEX求解结果对比,该算法具有求解速度快、适应性强等优点。Almeida算法提出了一种资源权重计算方法^[19],使得各资源的权重值在项目调度之初就已确定,整个调度过程中不再发生变化,是一种静态权重。而现实中的项目网络往往极为复杂,随着项目进行,大量已经完成的活动会造成资源权重中包含冗余和不合理部分,因此,在调度计划的后期这种静态的资源权重可能会起不到对资源的优选作用,甚至可能会误导资源的合理分配。

基于此,本文研究了多技能资源环境下求解最短工期的项目调度问题,设计了基于动态资源权重的启发式算法。该算法在整个调度过程中根据剩余项目网络动态地计算资源的权重,以优化资源分配,达到缩短项目工期的目的。新算法具有求解速度快、易实施等特点,对项目管理者而言,在应对现实项目的多变环境方面具有优势。

1 问题描述与数学模型

项目网络采用节点式(AON)表示,共包含 n 个节点,起始节点1和终止节点 n 代表虚活动,活动之间存在工序约束,执行顺序不可违背。以 $G = (V, E)$ 表示项目网络,其中: $V = \{1, 2, \dots, n\}$ 表示活动集合, E 为有向弧,表示活动的工序约束。用 P_j 表示活动 j 的前序活动的集合,活动的执行需要一定数量的技能, $F = \{1, \dots, l, \dots, L\}$ 表示项目需求的技能种类,以TF表示活动-技能需求矩阵,TF _{il} 表示活动 i 对技能 l 的需求量。将资源看作是技能的载体,以提供技能的形式供给各活动,这些资源可分为多种类型,典型的是可更新资源与不可更新资源,本研究只考虑可更新资源,以 $R = \{1, \dots, k, \dots, K\}$ 对项目可使用的每单位资源进行编号,不对相同资源归类,每个编号代表1单位资源,其最大可用量均为1。资源具备多技能,每个资源都至少具备一种项目所需求的技能,不考虑效率差异,以RF表示资源-技能矩阵,若RF _{kl} = 1则表示资源 k 具备技能 l ;若RF _{kl} = 0则表示资源 k 不具备技能 l 。在满足活动技能需求的条件下,活动 i 执行需要消耗一定的时间 d_i 。表1列出了本文所使用的符号。

此外,本文提出如下假设:

- 1) 活动一旦开始,不可中途中止或更改资源分配方案;
- 2) 不考虑资源的技能效率,即RF _{kl} ∈ {0, 1};
- 3) 在同一时刻,资源最多只能在一个活动中执行一项技能。

表1 符号说明

符号	含义	符号	含义	符号	含义
i, j	活动序号	k, l	资源序号, 技能序号	n	活动数量
d_i	活动 i 的工期	h	项目最大完工时间, $h = \sum_{i=1}^n d_i$	T	时刻点集合, $T = \{0, 1, \dots, h\}$
t	决策时点, $t \in T$	K	资源数量	L	技能种类
V_t^{pc}	t 时刻, 紧前活动均完成的集合	V_t^s	t 时刻, 加入排程的活动集合	F^k	资源 k 具备的技能集合
R^l	具备技能 l 的资源的集合	R^{rls}	释放的资源技能	V^l	需求技能 l 的活动的集合
V^c	已完工活动的集合	V^p	正在进行的活动的集合	V^w	待排程的活动的集合
V^{rank}	据活动优先规则排序的活动集合	F^{vw}	待排程活动所需技能的集合	F^{vr}	V^{rank} 中活动所需技能的集合
R^a	可用资源集合	S_i	活动 i 的开始时间	TF_{il}	活动 i 对技能 l 的需求数量
w_k^s	静态资源权重	w_k^d	动态资源权重	w_{kl}	当资源 k 执行技能 l 时的权重

1.1 引例

为了便于说明本文所研究的问题, 引入算例I, 其网络结构如图1所示. 项目执行需要5种技能, 活动1~活动6的活动时间分别为0、2、5、2、2、0, 各活动的技能需求标注在图中对应节点上方, 项目的可用

资源数量为3, 资源-技能矩阵为 $RF = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$.

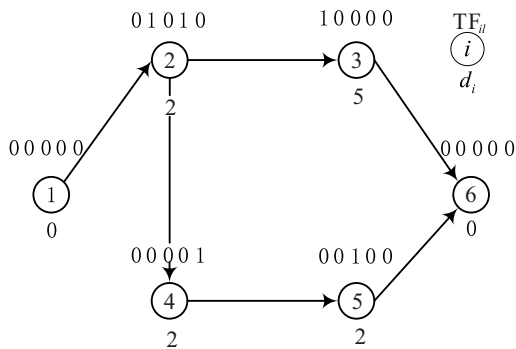
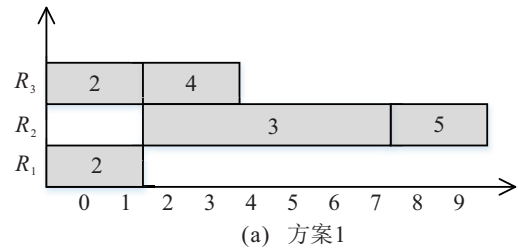


图1 引例I

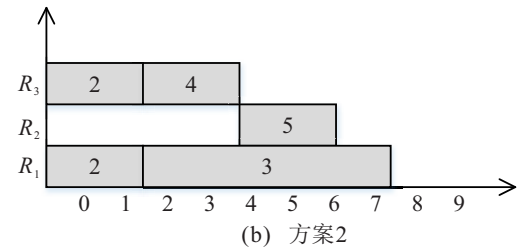
由RF可以看出, 资源 R_1 具备技能 F_1 、 F_2 、 F_4 , 资源 R_2 具备技能 F_1 、 F_3 , 资源 R_3 具备技能 F_2 、 F_5 . TF显示, 活动1和活动6为虚活动, 不需要资源和时间; 活动2需要技能 F_2 、 F_4 各1单位, 活动时间为2; 活动3需要1单位技能 F_1 ; 活动4需要1单位技能 F_5 ; 活动5需要1单位技能 F_3 .

图2所示的甘特图为引例I的两种调度方案, 这两种调度方案的工序均为 {1, 2, 3, 4, 5, 6}, 其不同之处在于资源的分配. 资源-技能矩阵RF显示资源 R_2 同时具备技能 F_1 和技能 F_3 , 这导致它既可分配给活动3也可分配给活动5. 在图2(a)中, 当 $t = 2$ 时, 资源 R_2 被用来满足活动3对技能 F_1 的需求, 项目的最后工期为9. 在图2(b)中, $t = 2$ 时, 被用来满足活动3对技能 F_1 需求的资源是 R_1 , 而资源 R_2 被分配给了活动5, 项目的最后工期为7.

以上案例表明, 多技能属性使得资源可以以不同的技能形式参与到不同活动之中, 这导致资源的分配会协同调度计划对项目的工期产生影响. 如何在对



(a) 方案1



(b) 方案2

图2 引例I两种调度方案的甘特图

项目活动进行排序的过程中就考虑多技能资源的分配以达到缩短项目工期的目的, 正是本文的研究内容.

1.2 模型构建

本研究的目标是通过资源合理调度以使得整个项目的总工期最短. 该问题的决策变量为 y_{ktli} , 当资源 k 在 t 时刻以技能 l 被活动 i 使用时, $y_{ktli} = 1$, 否则 $y_{ktli} = 0$, 数学模型如下:

$$\min S_n. \tag{1}$$

$$\text{s.t. } M_{it} = \sum_{k=1}^K \sum_{l=1}^L y_{ktli}, \forall i, t. \tag{2}$$

$$U_{it} = \begin{cases} 1, & M_{it} > 0; \\ 0, & M_{it} = 0; \end{cases} \forall i, t. \tag{3}$$

$$S_i = \frac{\sum_{t=1}^h (tU_{it})}{d_i} - \frac{d_i - 1}{2}, \forall i \in V \setminus \{1, n\}. \tag{4}$$

$$S_i + d_i \leq S_j, \forall i \in P_j, \forall j. \tag{5}$$

$$y_{ktli}(\text{RF}_{kl} - 1) = 0, \forall k, t, l, i. \tag{6}$$

$$\sum_{i=1}^n \left(U_{it} \sum_{l=1}^L y_{ktli} \right) \leq 1, \forall k, t. \tag{7}$$

$$\sum_{k=1}^K \sum_{t=0}^h y_{ktli} = U_{it} TF_{il}, \forall i, l, t. \quad (8)$$

$$\left(\sum_{t=0}^h y_{ktli} - d_i \right) \sum_{t=0}^h y_{ktli} = 0, \forall k, t, l, i. \quad (9)$$

$$y_{ktli} \in \{0, 1\}, \forall k, t, l, i. \quad (10)$$

目标函数(1)表示最小化项目工期;式(2)中 M_{it} 表示在时刻 t 活动 i 所占用的资源总量;式(3)中 U_{it} 表示在时刻 t 活动 i 的状态,若 $U_{it} = 1$ 则表示活动 i 正在进行,反之活动 i 未开始或已完工;式(4)表示由活动的状态系数推算活动的开始时间;式(5)表示活动排程不能违背项目的工序约束;式(6)表示资源不能执行其不具备的技能;式(7)表示资源在某一时刻最多只能执行一种技能,且最多只能分配给一个活动;式(8)表示各活动的技能需求必须得到满足;式(9)表示资源 k 或是完全不分配给活动 i ,或是在活动 i 的整个活动时间内都分配给它,即资源分配不可中途更改,也保证了活动不可中断;式(10)为决策变量的可行域。

2 算法设计

本研究在求解 MSPSP 时,主体采用并行调度方式,用 V_t^{pc} 表示在决策时刻 t 可排活动的集合,它表示不考虑资源约束,满足工序约束的可以加入排程的所有活动的集合.算法的难点是:1)如何根据当前剩余资源及可排活动集 V_t^{pc} 求解当前要加入排程的活动集 V_t^s ;2)如何分配资源-技能-活动以达到工期最短的目的。

为解决以上问题,本文设计二分图最大匹配法求解 V_t^s ,而后基于动态资源权重确定资源的分配.其思想是:得到资源的权重后,首先将确定资源执行哪一技能,即求解资源-技能分配矩阵,而后根据活动资源分配的优先规则,从资源-技能分配矩阵中选择执行其所需技能的资源,最终完成资源到活动的分配。

2.1 基于二分图最大匹配法的 V_t^s 求解

V_t^s 指的是在时刻 t 满足工序约束且满足当前剩余资源约束的活动集,它是可排活动集 V_t^{pc} 的一个子集, V_t^s 中所有活动将于当前时刻 t 开始。

在传统 RCPS 的并行调度过程中,求解 V_t^s 只需要简单对比当前剩余资源与 V_t^{pc} 的资源需求,并根据活动优先规则依次从 V_t^{pc} 中移除活动,直至当前剩余资源能满足 V_t^{pc} 的资源需求即可,此时的 V_t^{pc} 即为 V_t^s .而在 MSPSP 中,多技能属性导致即使资源在供应量确定的情况下,各技能的供应量也是不确定的,所以并不能通过简单对比技能需求和资源供给来判

断资源是否充足.考虑引例 II,其网络结构如图 3 所示,项目执行需要 3 种技能,各活动的需求标注在图中,现有可用资源量为 5.其中:资源 R_1 和 R_2 具备技能 F_1 ,资源 R_3 具备技能 F_1, F_2, F_3 ,资源 R_4 和 R_5 具备技能 F_2 .

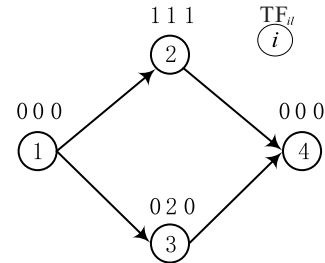


图 3 引例 II

由以上可知,在 $t = 0$ 时,活动 2、3 都在可排活动集 V_0^{pc} 中,若进行资源量对比, V_0^{pc} 需要资源 5 单位,当前资源剩余 5 单位,得 $V_0^s = \{2, 3\}$;若进行技能量对比, V_0^{pc} 需要 1 单位技能 F_1 、3 单位技能 F_2 和 1 单位技能 F_3 ,当前资源可提供 3 单位技能 F_1 、3 单位技能 F_2 和 1 单位技能 F_3 ,技能量均满足,得 $V_0^s = \{2, 3\}$ 。

进一步研究 RF 可知,若资源 R_3 执行技能 F_3 ,则剩余资源最多仅能提供 2 单位的 F_2 ,无法满足 V_0^{pc} 对 F_2 的需求;同样的,若 R_3 执行技能 F_2 或 F_1 ,则技能 F_3 的需求得不到满足,也即 $V_0^s \neq \{2, 3\}$ 。

基于此,本文提出二分图最大匹配法以求解 V_t^s .如图 4,将引例 II 中的当前剩余资源和技能需求分别列出,左侧每个节点代表 1 单位剩余资源,右侧每个节点代表 1 单位技能需求,这样就形成了一个二分图。

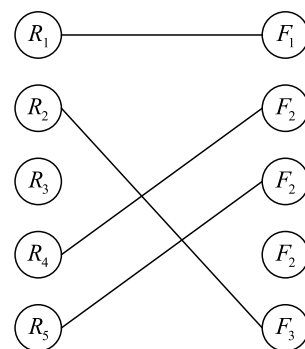


图 4 供给-需求二分图

在图论中,二分图中的每条连线代表一个“匹配”,它从资源顶点出发,连接到技能顶点,受到资源-技能的约束,它只能连接到该资源可执行的技能顶点.同时,任意两条匹配线都不能存在公共顶点.在所有满足二分图约束的匹配组合中,匹配数量最多的组合称为二分图的一个最大匹配 (NEMM).图 4 中的匹配即为该二分图的一个最大匹

配,它包含4个匹配(如图中连线所示),这表示该二分图的右侧会存在1个顶点未得到匹配(F_2),即 V_t^{pc} 需求的技能会有1个(即 F_2)得不到满足,由此可得活动2和活动3不可能同时开始, $V_0^s \neq \{2, 3\}$. 这样,根据二分图的最大匹配数与技能需求数量的对比,并结合活动优先规则,即可求解 V_t^s . 在求解二分图的最大匹配数时,本文采用已经得以广泛研究的匈牙利算法^[20]. 算法1为求解 V_t^s 算法的伪代码.

算法1 求解 V_t^s .

input: t ;

output: V_t^s .

1) 求解 V_t^{pc} , R^a , $V^{\text{rank}} \leftarrow V_t^{pc}$

2) 求解 F^{vr} , NEMM

3) while NEMM < $|F^{vr}|$ do

 根据活动优先规则对 V^{rank} 中 k 的活动进行排序,并移除最后一个活动;

 更新 V^{rank} , F^{vr} , 通过匈牙利算法更新 NEMM.

4) end while

5) $V_t^s \leftarrow V^{\text{rank}}$

 return V_t^s .

2.2 静态与动态资源权重的计算

通过二分图最大匹配理论可得到 V_t^s , 在当前资源约束下, V_t^s 中的活动可同时开始,但由于资源具备多技能,最优的资源分配方案尚不能确定. 为此,本文设计了动态资源权重,以解决资源分配问题. MSPSP 问题中的静态资源权重的概念是由 Almeida 等^[19]提出的,其意义是资源 k 对整个项目网络的重要程度,在进行资源分配时,权重值更大的资源被分配给优先权更高的活动,资源权重值计算公式为

$$w_k^s = |F^k| \max_{l \in F^k} \left\{ \frac{|R^l|}{|R^a|} \sum_{i \in V^l} (d_i \text{TF}_{il}) \right\}, \forall k \in R. \quad (11)$$

它包含对资源3方面的考量:资源所具技能的多样性、资源所具技能的稀缺程度以及项目对资源所具技能的需求程度.

然而,随着项目的进行,在决策时刻 t ,项目可以分为3部分:已完成活动 V^c 、进行中的活动 V^p 和待排程活动 V^w . 此时,当前剩余资源对 V^c 和 V^p 并无任何影响,同时 V^c 和 V^p 也不影响这些资源的重要性. 因此,这部分影响应从资源权重中减去,权重随之发生变化,其更新周期为两相邻决策时刻的间隔,这即是本文提出的动态资源权重的概念.

动态资源权重 w_k^d 的意义是资源 k 对项目剩余网络 V^w 的重要程度,在进行资源分配时,一方面尽可能

将 w_k^d 更大的资源闲置,以留待剩余网络中的活动使用;另一方面,若无资源可闲置,则将 w_k^d 更大的资源分配给优先权更高的活动,其大小为

$$w_k^d = \frac{|R^k|}{|F^k \cap F^{vw}|} \max_{l \in F^k \cap F^{vw}} \left\{ \frac{|R^l|}{|R^l \cap |R^a| - \sum_{i \in V_t^s} \text{TF}_{il} + 1} \sum_{i \in V^w} (d_i \text{TF}_{il}) \right\}, \forall k \in R, t \in T. \quad (12)$$

动态权重跟随计划的进行而更新,将已经加入排程的活动对资源的影响从权重中剔除,它同样包含3方面的考虑:资源所具备的对待排活动有用技能的多样性、资源所具备的对待排活动有用技能的稀缺程度以及待排程活动对资源所具技能的需求程度.

2.3 基于整数规划法的资源分配

由以上理论可知,得到 V_t^s 后,只要资源-技能的分配是供给-需求二分图的一个最大匹配,这一分配方案就可满足 V_t^s 的技能需求,然而二分图的最大匹配往往不止一个,且当前剩余资源的数量也不一定恰好等于 V_t^s 需求的资源数量(通常会多于),这将导致可能会存在部分资源在当前决策时刻被闲置. 基于此,本文引入虚拟技能,将其编号为 $L + 1$,其需求量为剩余的资源数量减去 V_t^s 需求的资源数量,即

$$\sum_{i \in V} \text{TF}_{i(L+1)} = |R^a| - \sum_{i \in V_t^s} \sum_{l \in F} \text{TF}_{il}, \forall t \in T. \quad (13)$$

最终分配给技能 $L + 1$ 的资源将会被闲置,用 $y_{kt(L+1)*} = 1$ 表示. 这样,资源数量与技能需求数量相等,资源-技能的分配问题可以描述为:

1) 如何确定各资源执行哪种技能,使技能供给恰好等于技能需求;

2) 如何将未排活动 V^w 更有用的资源闲置下来,以待后续决策时使用,即分配给虚拟技能 $L + 1$.

为解决以上问题,本文建立0-1整数规划模型求解资源-技能分配矩阵 (ARF),目标函数是最大化闲置资源的总权重,以 x_{kl} 为决策变量,表示在资源-技能的分配方案中,资源 k 是否执行了技能 l ,求得的矩阵 $\text{ARF}_{K(L+1)} = (x_{kl})_{K(L+1)}$ 即为资源-技能分配矩阵 ARF.

$$\min \sum_{k=1}^K \sum_{l=1}^{L+1} (x_{kl} w_{kl}). \quad (14)$$

$$\text{RF}_{k(L+1)} = 1, \forall k. \quad (15)$$

$$w_{kl} = \begin{cases} w_k, & \text{RF}_{kl} = 1, \forall k, l; \\ +\infty, & \text{RF}_{kl} = 0, \forall k, l; \\ 0, & l = L + 1, \forall k. \end{cases} \quad (16)$$

$$\sum_{l=1}^{L+1} x_{kl} = \begin{cases} 1, \forall k \in R^a; \\ 0, \forall k \notin R^a. \end{cases} \quad (17)$$

$$\sum_{k=1}^K x_{kl} = F_l^{vst}, \forall l \in F \cup \{L+1\}. \quad (18)$$

$$x_{kl} = \{0, 1\}, \forall k, l \in F \cup \{L+1\}. \quad (19)$$

式(14)表示目标函数最小化资源分配的总权重,其中 x_{kl} 为决策变量,当资源 k 在某活动中执行技能 l 时 $x_{kl} = 1$,否则 $x_{kl} = 0$.动态资源权重的意义就是资源对剩余网络的重要性,所以最小化资源分配的总权重在这里的含义就是最大化闲置资源的权重,即在满足当前活动资源需求的基础上,将对剩余网络更有用的资源闲置下来,其中 w_{kl} 为资源 k 执行技能 l 时的权重.式(15)表示任意资源都可执行虚拟技能 $L+1$,即任意资源都可闲置.式(16)表示若资源 k 具备技能 l ,则 $w_{kl} = w_k$;若资源 k 不具备技能 l ,则 w_{kl} 为无穷大,对于虚拟技能 $L+1$,任意 $w_{k(L+1)} = 0$.式(17)表示当前可用资源约束.式(18)表示资源的需求约束,其中 F_l^{vst} 表示 V_t^s 对技能 l 的需求总量.式(19)为决策变量可行域.通过以上0-1型整数规划模型,解得资源-技能分配矩阵ARF,而后结合资源权重、各活动的技能需求及相关活动的优先规则,可进一步得到资源-技能-活动的分配方案,而后根据当前时刻 t ,即可求得MSPSP总体模型的决策变量 y_{ktli} .

2.4 算法流程

算法2 MSPSP求解算法.

input: G, R, F ;

output: S , 资源分配方案.

1) 初始化 $t, V_t^{pc}, V^c, V^w, R^a, Feasibility = 1$

2) for $i \in V$ do

考虑任意两个活动都不并行,通过二分图最大匹配法检验项目可行性.

3) if $NEMM < \sum_{l \in F} TF_{il}$ then
Feasibility = 0

4) end if

5) end for

6) if Feasibility = 0 then
资源不足,项目不可行.

7) else

8) while $V^w \neq \emptyset$ do

$R^a \leftarrow R^a \cup R^{rls}$, 执行算法1, 得 V_t^s , 更新 w_k^d , 通过0-1规划及资源分配规则分配资源.

9) for $i \in V_t^s$ do

$S_i \leftarrow t$

10) end for

11) end while

$t \leftarrow t + 1$

12) end if

return S 及资源分配方案.

为了进一步解释算法,其具体流程如下.

step 1: 初始化各项参数, $t = 0, V_0^{pc} = \emptyset, V^c = \emptyset, V^p = \emptyset, V^w = V$; 用二分图最大匹配法检查项目是否可行, 若项目可行, 则转入step 2; 否则, 终止算法.

step 2: 检查 t 时刻 V^p 中活动的完成情况, 将已完成的活动从 V^p 中移入 V^c , 并将这些活动所占用的资源释放, 即 R^{rls} . 将其加入到可用资源集 R^a , 从 V^w 中选取所有紧前活动都已在 V^c 中的活动, 将之从 V^w 移入 V_t^{pc} . 若此时 $V^w = \emptyset, V_t^{pc} = \{n\}, V^c = V \setminus \{n\}$, 则转入step 9; 否则转入step 3.

step 3: 根据 R^a , 利用二分图最大匹配法判断 V_t^{pc} 中活动能否同时开始, 若可以, $V_t^s = V_t^{pc}, V^p = V^p \cup V_t^s$, 则转入step 5; 若否, 则转入step 4.

step 4: 对 V_t^{pc} 中所有活动, 按照活动排序的优先规则(LST+LFT+Random)进行排序, 将优先权最低的一个活动从 V_t^{pc} 中移入 V^w , 而后转入step 3.

step 5: 更新 R^a 中所有资源的资源权重 w_k^d , 转入step 6.

step 6: 根据 V_t^s 中活动对技能的需求和当前可用资源, 用整数规划法求解资源-技能分配方案ARF, 确定资源将执行的技能, 转入step 7.

step 7: 根据活动资源分配的优先规则(SPT+Random), 将资源逐一分配给 V_t^s 中各活动, 并将 V_t^s 中活动 i 的开始时间设为 $S_i = t$, 转入step 8.

step 8: 计算 V^p 中所有活动的结束时间 $FT_i = S_i + d_i$, 使 $t = \min_{i \in V^p} FT_i$, 转到step 2.

step 9: 调度结束, $S_n = t$ 即为该调度计划的项目工期, $S = \{S_1, \dots, S_i, \dots, S_n\}$ 即为项目的调度计划.

step 7的具体步骤为: 从 V_t^s 中工期最短的活动中随机选一个活动 i , 任选其需求的一个技能 F_i , 从资源-技能的分配方案中, 找出执行技能 F_i 的权重最高的资源 $k, k = \{k | w_k^d = \max_{RF(m, F_i)=1, m \in R^a} w_m\}$, 若多于一个则从中随机选择一个, 将之以技能 F_i 的形式分配给活动 i , 即 $y_{ktF_i, i} = 1$, 并将其移出 R^a , 循环执行该过程, 直至活动 i 的所有技能需求都得以满足, 将活动 i 从 V_t^s 中移入 V^p , 选择下一个工期最短的活动, 循环此过程, 直至 $V_t^s = \emptyset$.

3 数值实验

为验证算法的有效性, 本部分将对其进行数值实

验,相关程序用 Matlab R2014b 进行编写,测试平台为 Windows 10 操作系统,处理器为 Intel(R) Core(TM)i3-6100 3.70 GHz,内存 4.00 G.

3.1 算例集

本文参考文献[21]提出的多技能资源受限项目算例生成方法,所生成的算例集包含 4 个子集 J30、J60、J90、J120,命名中的数字表示该子集内的每个项目网络所包含的活动数,活动所需求及资源能提供的技能种类有 4 种,资源数量不固定. 设置 3 个参数: 1) 网络复杂性 NC(network complexity),代表项目网络的复杂程度,以活动的平均紧后工序数表示, $NC \in \{1.5, 1.8, 2.1\}$; 2) 需求因素 RF(requirement factor),代表活动对所提供技能的需求强度,以活动平均需求技能种类数与资源能提供的技能种类数的比值表示,如 $RF = 0.5$ 表示项目中活动的执行平均需求 $4 \times$

$0.5 = 2$ 种技能, $RF \in \{0.25, 0.5, 0.75, 1\}$; 3) 技能水平 SL(skill level),代表资源的多技能强度,以资源具备的平均技能数与资源能提供的技能种类数的比值表示,如 $SL = 0.6$,表示项目中的资源平均具备 $4 \times 0.6 = 2.4$ 种技能, $SL \in \{0.4, 0.6, 0.8\}$. 每个参数组合下设置 10 个算例,所以每个算例子集包含 $3 \times 4 \times 3 \times 10 = 360$ 个算例,共 1440 个算例.

3.2 结果分析

3.2.1 总体结果分析

采用动态权重算法对算例集进行实验,并与资源随机分配算法以及文献[19]中的静态资源权重算法对比,得到如下实验结果. 表 2 中,平均活动工期是每个算例子集内 360 个算例的平均工期,最优解指的是 3 种算法的解中的最小值,最优解个数占比指该算法求得的最优解个数在 360 个算例中所占的比例.

表 2 基于活动数量的算法结果对比

	平均活动工期				最优解个数占比/%			
	J30	J60	J90	J120	J30	J60	J90	J120
随机	89.1	136.2	149.6	154.8	27.2	18.3	19.2	22.2
静态	66.0	89.3	100.9	106.5	66.3	62.2	58.6	62.5
动态	65.1	88.6	100.3	105.7	85.3	84.4	85.8	87.2

从总体看,基于资源权重的启发式算法显著缩短了项目工期,求得最优解的几率也远高于随机分配方式,且这种优化效果不受活动数量的影响;对比两种启发式算法可知:相较于静态资源权重算法,动态资源权重算法不仅求得了更短的项目工期,且求解效果稳定,同时,求得最优解的几率显著提高,这一结论不受项目规模的影响.

3.2.2 算法在不同项目环境中的表现分析

算法在不同的项目环境中其求解效果会存在差异,为了研究本文算法在不同参数的项目算例中的表现,对以上算例集进行相同的实验步骤,并最终根据 NC、SL、RF 分别统计实验结果.

由图 5 可以看出,无论在何种项目环境中,基于动态资源权重的启发式算法都优于其他两种算法. 表 3 表明,相较于随机分配,两种启发式算法能够缩短平均项目工期 20% 左右,并提高求得最优解的几率 40% 以上. NC 的变化并不影响 3 种算法的相对结果,而 SL 却不然,由图 5 和表 3 可以看出,随着 SL 的增加,随机资源分配算法性能渐增,与其他两种算法的差距也逐渐缩小. 这是由于技能水平的提高会导致资源向“全能型”资源转化,即所有资源都拥有较多的技能,资源间的功能差异变小,资源的权重趋于相等,从而导致基于资源权重的算法更趋近于随机算法,随机算法跟两种启发式算法的差距也就变小.

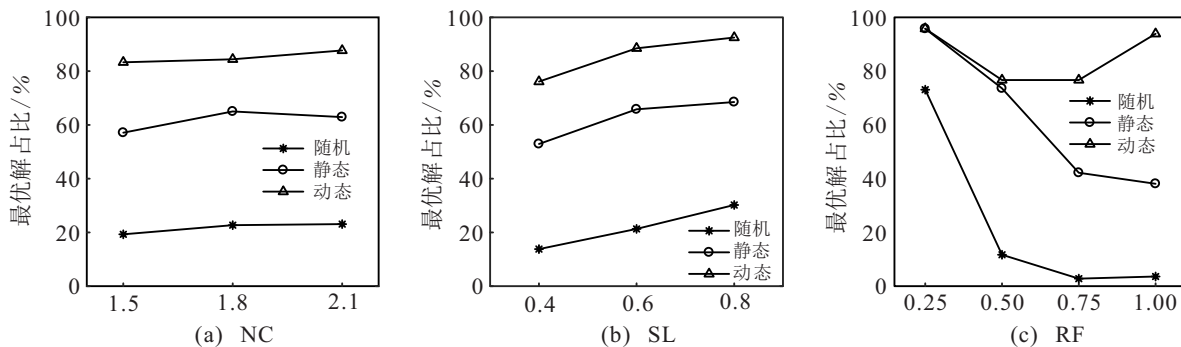


图 5 不同项目参数下各算法的实验结果

表3 基于NC、SL及RF的算法结果对比

参数		平均活动工期			最优解个数占比/%		
		随机	静态	动态	随机	静态	动态
NC	1.5	132.3	87.7	86.9	19.3	57.1	83.3
	1.8	131.6	90.9	90.4	22.7	65.0	84.4
	2.1	133.7	93.4	92.6	23.1	62.9	87.7
SL	0.4	171.5	92.5	91.7	13.8	52.9	76.1
	0.6	166.6	119.5	118.6	21.3	65.8	88.5
	0.8	134.5	119.9	119.1	30.2	68.5	92.5
RF	0.25	80.1	77.1	77.1	73.1	95.8	95.8
	0.5	105.4	77.7	77.6	11.7	73.6	76.7
	0.75	146.7	92.3	91.2	2.8	42.2	76.7
	1	197.6	115.5	114.2	3.6	38.1	93.9

在表3中,当 $RF > 0.25$,即项目活动平均需求技能超过一个时,随机资源分配算法和静态资源权重算法的最优解占比出现明显下降,而动态资源权重算法依旧能保持较高的性能,这表示在项目中,当活动的需求越复杂时,动态资源权重算法越具有优越性。这是由于,在资源有限这一约束条件下,当活动对资源需求更加复杂时,项目的工期更易受到资源分配方案的影响,动态资源权重算法综合考虑了排程过程中当前活动的调度及对后续活动的影响,能更合理地对资源调配,从而获得工期更短的调度计划。

4 结论

根据多技能资源受限项目的特点,本文设计了基于动态资源权重的启发式算法,算法主体采用并行调度,通过二分图最大匹配理论筛选可排活动,而后综合考虑项目对资源的需求及资源自身的属性,引入动态资源权重这一指标,以最小化资源的分配费用为目标进行资源-技能的分配,最后根据相关优先规则完成资源-技能-活动的分配。通过测试大量不同规模及复杂程度的算例,得出以下结论:

- 1) 新算法在解的质量方面超越了随机算法及静态资源权重算法,能有效缩短调度计划的工期;
- 2) 新算法在求解稳定性方面超越了随机算法及静态资源权重算法,能在多种参数组合下保证求解质量;
- 3) 技能水平(SL)越高,新算法求解优势越低,但仍是最优算法;
- 4) 活动的需求水平(RF)越高,新算法优越性越明显。

参考文献(References)

[1] Kolisch R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and

computation[J]. *European Journal of Operational Research*, 1996, 90(2): 320-333.

- [2] 刘士新, 宋健海, 唐加福. 基于关键链的资源受限项目调度新方法[J]. *自动化学报*, 2006, 32(1): 60-66.
(Liu S X, Song J H, Tang J F. Critical chain based approach for resource-constrained project scheduling[J]. *Acta Automatica Sinica*, 2006, 32(1): 60-66.)
- [3] 何正文, 刘人境, 胡信布, 等. 现金流平衡约束下的Max-npv项目调度[J]. *系统工程理论与实践*, 2009, 29(3): 132-141.
(He Z W, Liu R J, Hu X B, et al. Max-npv project scheduling problems with cash flow balance constraints[J]. *Systems Engineering—Theory & Practice*, 2009, 29(3): 132-141.)
- [4] Leyman P, Vanhoucke M. Capital- and resource-constrained project scheduling with net present value optimization[J]. *European Journal of Operational Research*, 2017, 256(3): 757-776.
- [5] Rieck J, Zimmermann J, Gather T. Mixed-integer linear programming for resource leveling problems[J]. *European Journal of Operational Research*, 2012, 221(1): 27-37.
- [6] Markou C, Koulinas G K, Vavatsikos A P. Project resources scheduling and leveling using multi-attribute decision models: Models implementation and case study[J]. *Expert Systems with Applications*, 2017, 77: 160-169.
- [7] Correia I, Lourenço L L, Saldanha-da-Gama F. Project scheduling with flexible resources: Formulation and inequalities[J]. *OR Spectrum*, 2012, 34(3): 635-663.
- [8] 贾艳, 王致民, 张跃刚. 柔性资源约束项目调度问题的启发式求解方法[J]. *计算机集成制造系统*, 2015, 21(7): 1846-1855.
(Jia Y, Wang Z M, Zhang Y G. Heuristic algorithm for flexible resource-constrained project scheduling

- problem[J]. *Computer Integrated Manufacturing System*, 2015, 21(7): 1846-1855.)
- [9] 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.
- [10] 陈蓉, 梁昌勇, 叶春森, 等. 考虑随机离职的新产品研发项目组合多技能员工调度模型[J]. *系统工程理论与实践*, 2018, 38(1): 164-176.
(Chen R, Liang C Y, Ye C S, et al. A multi-skilled staff scheduling model for new product R&D project portfolio under stochastic turnover[J]. *Systems Engineering — Theory & Practice*, 2018, 38(1): 164-176.)
- [11] 任逸飞, 陆志强. 多技能资源投入项目调度问题的建模与优化[J]. *同济大学学报: 自然科学版*, 2017, 45(11): 1713-1721.
(Ren Y F, Lu Z Q. Modeling and optimization of resource investment project scheduling problem with multi-skill[J]. *Journal of Tongji University: Natural Science*, 2017, 45(11): 1713-1721.)
- [12] 李明, 李前进. 项目多技能人力资源指派与调度混合算法[J]. *数学的实践与认识*, 2017, 47(19): 22-28.
(Li M, Li Q J. A hybrid algorithm to solve multi-skilled human resource allocation and project scheduling problem[J]. *Mathematics in Practice and Theory*, 2017, 47(19): 22-28.)
- [13] Peteghem V V, Vanhoucke M. A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem[J]. *European Journal of Operational Research*, 2010, 201(2): 409-418.
- [14] Javanmard S, Afshar-Nadjafi B, Niaki S T A. Preemptive multi-skilled resource investment project scheduling problem: Mathematical modelling and solution approaches[J]. *Computers & Chemical Engineering*, 2017, 96: 55-68.
- [15] Bellenguez O, Néron E. Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills[C]. *The 5th International Conference on Practice and Theory of Automated Timetabling V*. Pittsburgh, 2004: 229-243.
- [16] Ciro G C, Dugardin F, Yalaoui F, et al. A fuzzy ant colony optimization to solve an open shop scheduling problem with multi-skills resource constraints[J]. *IFAC-Papers Online*, 2015, 48(3): 715-720.
- [17] 陈君兰, 叶春明. 柔性资源受限多项目调度的混沌粒子群算法研究[J]. *计算机应用研究*, 2013, 30(1): 117-120.
(Chen J L, Ye C M. Chaos particle swarm optimization on flexible-resource constrained multi-project scheduling research[J]. *Application research of Computers*, 2013, 30(1): 117-120.)
- [18] Maghsoudlou H, Afshar-Nadjafi B, Niaki S T A. A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem[J]. *Computers & Chemical Engineering*, 2016, 88: 157-169.
- [19] Almeida B F, Correia I, Saldanha-da-Gama F. Priority-based heuristics for the multi-skill resource constrained project scheduling problem[J]. *Expert Systems with Applications*, 2016, 57: 91-103.
- [20] Kuhn H W. Statement for naval research logistics: "The hungarian method for the assignment problem" [J]. *Journal of the Royal Statistical Society*, 2015, 64(4): 611-633.
- [21] Almeida B F, Correia I, Saldanha-da-Gama F. An instance generator for the multi-skill resource-constrained project scheduling problem[ED/OL]. [2019-10-06]. https://www.researchgate.net/publication/309410985_An_Instance_Generator_for_the_Multi-Skill_Resource-Constrained_Project_Scheduling_Problem.

作者简介

胡振涛(1988—), 男, 博士生, 从事项目调度的研究, E-mail: gentlehzt@163.com;

崔南方(1963—), 男, 教授, 博士生导师, 从事项目调度、供应链管理等研究, E-mail: nfcui@mail.hust.edu.cn;

张艳(1993—), 女, 博士生, 从事项目调度的研究, E-mail: 1137057178@qq.com;

胡雪君(1989—), 女, 讲师, 博士, 从事项目调度的研究, E-mail: xuejun_hu@hnu.edu.cn.

(责任编辑: 孙艺红)