

混合贪心策略人工蜂群的两阶段算法针对资源受限项目调度问题的研究

宁 飒, 白 羽*

北京建筑大学理学院, 北京

收稿日期: 2026年1月1日; 录用日期: 2026年1月28日; 发布日期: 2026年2月5日

摘 要

资源受限项目调度问题(Resource-Constrained Project Scheduling Problem, RCPSP)是最具代表性的一类项目调度问题, 由于问题包含多个约束, 因此当任务规模较大时难以求得全局最优解。本文提出了混合贪心策略人工蜂群的两阶段算法, 在第一阶段采用混合初始解生成方式, 在雇佣蜂阶段引入差分进化策略, 并将观察蜂划分为精英开采与随机开采两种模式, 设计自适应阈值生成机制; 将第一阶段生成解决方案作为第二阶段的初始解, 第二阶段开发了基于贪心策略的解优化机制。在PSPLIB基准数据集上, 将该算法与人工蜂群算法、差分进化算法以及分支定价算法在数据集上的最优解覆盖率进行对比, 分别优于经典算法5.93%, 6.72%和11.32%。并用该算法求解了工程项目实例, 得出了项目调度方案。

关键词

资源受限项目调度问题, 人工蜂群算法, 贪心策略, 两阶段, PSPLIB基准数据集

Research on a Two-Phase Hybrid Greedy Artificial Bee Colony Algorithm for Resource-Constrained Project Scheduling Problem

Sa Ning, Yu Bai*

School of Science, Beijing University of Civil Engineering and Architecture, Beijing

Received: January 1, 2026; accepted: January 28, 2026; published: February 5, 2026

*通讯作者。

文章引用: 宁飒, 白羽. 混合贪心策略人工蜂群的两阶段算法针对资源受限项目调度问题的研究[J]. 运筹与模糊学, 2026, 16(1): 80-90. DOI: 10.12677/orf.2026.161008

Abstract

The Resource-Constrained Project Scheduling Problem (RCPSP) is one of the most representative types of project scheduling problems. Due to the multiple constraints involved, it is difficult to obtain a globally optimal solution when the task scale is large. A two-phase hybrid greedy artificial bee colony algorithm is proposed. In the first stage, a hybrid initial solution generation method is adopted. A differential evolution strategy is introduced during the employed bee phase, and the onlooker bees are divided into two modes: elite exploitation and random exploitation, with an adaptive threshold generation mechanism designed. The solutions generated in the first stage serve as the initial solutions for the second stage, where a solution optimization mechanism based on a greedy strategy is developed. On the PSPLIB benchmark dataset, the proposed algorithm is compared with the artificial bee colony algorithm, differential evolution algorithm, and branch-and-price algorithm in terms of optimal solution coverage rates. The results show that it outperforms the classical algorithms by 5.93%, 6.72%, and 11.32%, respectively. Furthermore, the algorithm is applied to a project case example, yielding a feasible project scheduling plan.

Keywords

Resource-Constrained Project Scheduling Problem, Artificial Bee Colony Algorithm, Greedy Strategy, Two-Stage, PSPLIB Benchmark Dataset

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

资源受限项目调度问题(Resource-Constrained Project Scheduling Problem, RCPSP)可以被定义为, 存在任务顺序约束及可更新资源约束的条件下, 给定有限资源的需求方案, 得到实现求解目标最优化的调度方案, 属于 NP-hard 问题, 是经典的组合优化问题[1]。RCPSP 一直是调度领域的热门研究内容, 目前已有护士排班[2]、机组排班[3]、软件项目调度[4]等诸多细分领域, 其调度目标一般为工期最小化。但由于在现实情境中, 该问题往往包含多个项目和资源, 因此当项目规模较大时, 项目之间存在的大量任务顺序约束和资源需求约束使得求解全局最优解尤为复杂。

PSPLIB 数据集作为调度问题领域的经典数据集, 被广泛使用。Türkakın 等[5]在该数据集上测试了 17 种优先规则在串行与并行两种机制下的求解效果。Pellerin [6]比较了不同混合元启发式在该数据集上的结果。Sallam [7]提出了一种基于强化学习的元启发式切换方法, 混合多操作员差分演化和离散布谷鸟搜索算法, 在该数据集上进行实验。

国内外求解资源受限项目调度问题的方法主要分为精确算法、启发式算法和元启发式算法。在求解小规模的问题时, 精确算法可以求出最优解, 例如分支界定法[8]、整数规划法[9][10]等, 但随着项目规模日益扩大, 精确算法不再适用。启发式算法是指通过对过去经验的归纳, 推理以及实验分析来解决问题的方法, 是一种求解近似解的方法。启发式算法能够在较短的时间内得到一个解, 但解的质量通常不够优质。目前求解该问题普遍使用元启发式算法, 例如遗传算法、蚁群算法、模拟退火算法、人工蜂群算法等。王海鑫等[11]针对标准粒子群算法易早熟进而影响优化结果的问题, 引入动态变惯性权重求解资源受限项目调度问题。张博等[12]针对船舶维修这一项目调度问题, 建立了数学模型并提出了改进遗传算法进行求解。

Merkle 等[13]提出了一种基于资源受限项目调度问题的蚁群优化算法, 其中蚂蚁通过结合两种信息素评估方式寻找新解, 引入启发式信息对蚂蚁决策影响的动态调整, 以及精英蚂蚁遗忘已发现最优解的策略。

人工蜂群算法(Artificial Bee Colony Algorithm, ABC)是一种模拟蜜蜂觅食行为的群体智能优化算法。通过雇佣蜂、观察蜂和侦察蜂的分工协作, 算法能在解空间中进行高效的全局搜索与局部开发, 常用于求解各类复杂的组合优化与函数优化问题。人工蜂群算法由 Karaboga [14]于 2005 年提出, Broderick Crawford 等[15]首次将人工蜂群算法应用于解决资源受限项目调度问题。石彦君等[16]将随机密钥引入人工蜂群算法, 并采用启发式优先级规则分配任务。Ming Chen [17]在传统的人工蜂群算法中引入了基于 Levy 飞行的群体逃离和觅食搜索策略, 其中侦察蜂、旁观者和自由蜂在寻找蜜源时有不同的搜索策略: 所有工蜂因受到干扰而放弃旧的蜜源, 并选择不同的方式寻找新的蜜源。人工蜂群算法利用仿生学, 模拟蜂群繁殖规律, 可以在短时间内求出可行解, 因此选用人工蜂群算法作为基础算法。但由于元启发式算法容易陷入局部最优解, 求解效果有一定随机性, 生成的大量可行解仍有改进空间。两阶段算法是在元启发式算法求解出的最优解基础上进行再优化, 是一种常用的算法改进策略。熊福力等[18]提出了一种两阶段混合迭代贪婪算法。第一阶段生成一个高质量置换解, 第二阶段则通过改变某些机器上的工件排序来改进第一阶段产生的置换解。唐媛媛等[19]设计了一种结合线性规划和启发式算法的两阶段混合算法, 混合算法中第一阶段将部分约束纳入到线性规划中求解, 获得初始结果, 第二阶段通过自研的启发式算法对初始结果进行修复后, 得到最终结果。

本文提出了一种混合贪心策略人工蜂群的两阶段算法, 第一阶段采用改进的人工蜂群算法, 求出的可行解作为第二阶段的初始解, 第二阶段采用贪心策略优化可行解。在 PSPLIB 数据集上进行实验, 并与其它三种算法的实验结果进行对比。最后利用该算法求解现实实例, 得出调度方案。

2. 资源受限项目调度问题模型

在 RCPSP 模型中, 项目包含多个任务, 且模型中的资源均为可更新资源, 每个任务对资源的需求量不同。可更新资源集合 $R = \{k | k = 1, 2, \dots, K\}$, 其中 R_k 表示第 k 种资源的总量。任务集合 $J = \{i | i = 1, 2, \dots, N\}$, 定义 s_i 表示任务 i 的开始时间, f_i 表示工序 i 的结束时间, p_i 表示执行工序 i 所需要的时间, r_{ik} 表示工序 i 对第 k 种资源的需求量。 P_i 为工序 i 的前序工序集, 工序 i 必须等待其前序工序集中的所有工序完成之后才能开始。本问题中的任务为连续任务, 即任务一旦开始, 在完成之前不能被中断。时间 $t \in [0, T]$, 设 t 时刻正在执行的任务集合为 $A(t)$ 。资源受限项目调度问题的标准模型如式(1)~式(6)所示。

$$\min \max \{f_i, \forall i \in J\} \quad (1)$$

s.t.

$$s_i + p_i = f_i, \forall i \in J, \quad (2)$$

$$f_j \leq s_i, \forall i \in J, \forall j \in P_i, \quad (3)$$

$$r_k(t) \leq R_k, \forall t \in [0, T], \quad (4)$$

$$r_k(t) = \sum_{i=1}^N r_{ik}(t), \quad (5)$$

$$r_{ik}(t) = \begin{cases} r_{ik}, & i \in A(t) \\ 0, & i \notin A(t) \end{cases} \quad (6)$$

其中, 式(1)表示目标函数定义为最小工期; 式(2)表示需要为任务分配足够的执行时间, 且执行不能被中断; 式(3)表示任务的开始时间必须在前序任务结束之后; 式(4)表示任务的资源约束。式(5)表示工序 i 对第 k 种资源的需求量 $r_{ik}(t)$ 的取值; 式(6)表示 t 时刻所有工序对第 k 种资源的需求总量 $r_k(t)$ 。

3. 混合贪心策略人工蜂群的两阶段算法

本文提出了混合贪心策略的两阶段人工蜂群算法, 改进主要包括以下方面: 首先, 采用基于优先规则的初始化, 提高初始种群质量, 并在雇佣蜂阶段引入差分进化的变异与交叉操作, 替代原有的单维度扰动, 从而增强算法的全局探索能力; 接着对观察蜂实施差异化处理, 通过划分精英开采与随机开采两种模式, 平衡局部深度寻优与种群多样性维持; 其次, 设计自适应阈值生成机制, 使侦查蜂的触发条件能根据种群状态动态调整, 实现智能化跳出局部最优; 最后, 在第二阶段引入基于贪心策略的解优化机制, 对当前解进行局部精细搜索, 进一步提升解的质量。算法流程如图 1 所示。

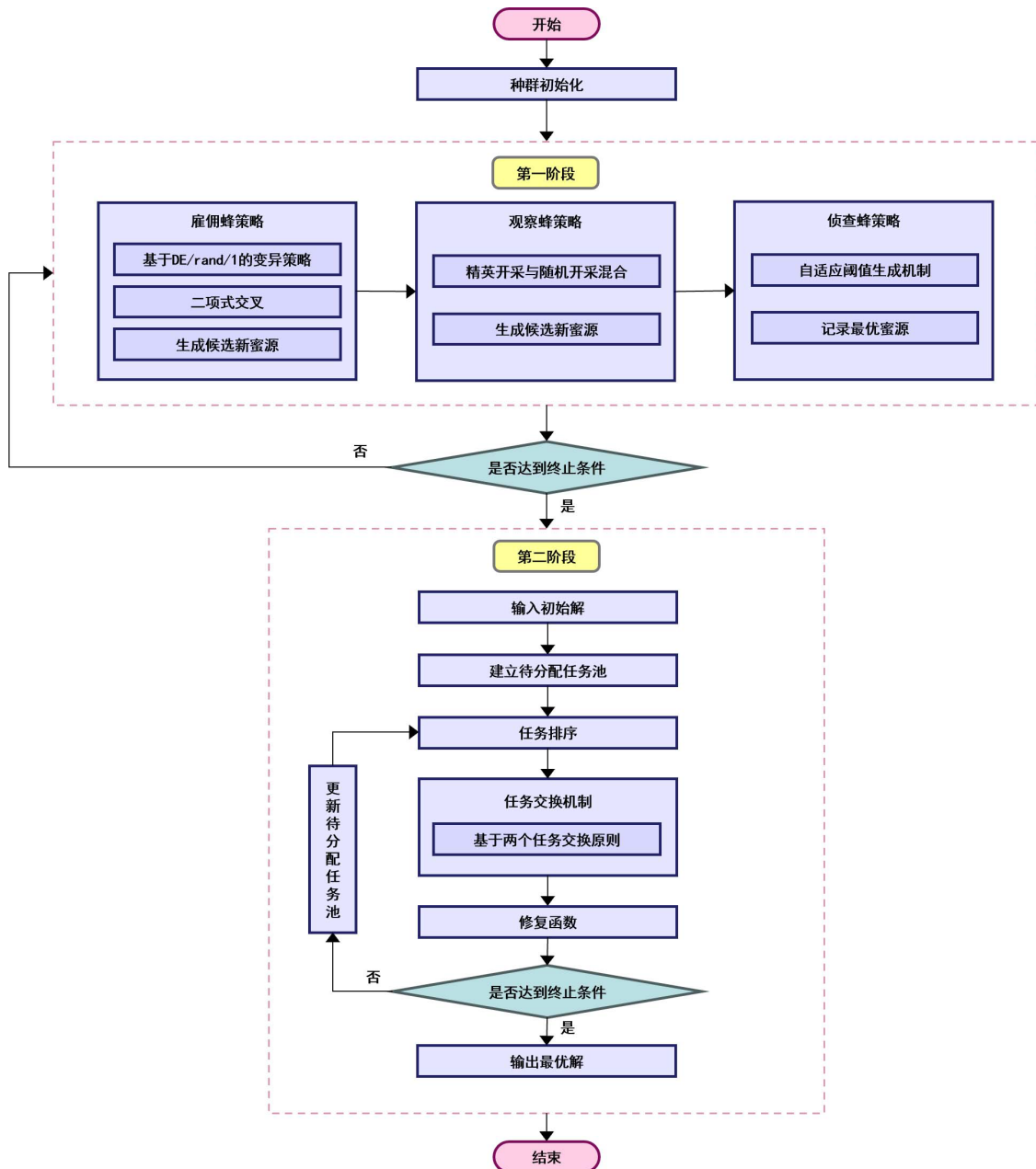


Figure 1. Flowchart of the two-phase hybrid greedy artificial bee colony algorithm
图 1. 混合贪心策略人工蜂群的两阶段算法流程图

3.1. 第一阶段

算法的第一阶段为改进的人工蜂群算法, 分别针对种群初始化、雇佣蜂操作、观察蜂操作以及侦查蜂操作进行改进。

3.1.1. 种群初始化

初始解的质量直接关系到最优解的生成, 采用基于优先规则的初始化方式, 初始解生成方式有如下三种: 随机生成初始解; 资源需求量大的优先: 根据任务资源需求量大小, 从大到小排列; 长任务优先: 工期较长的任务优先排列。

3.1.2. 种群初始化

雇佣蜂负责在对应蜜源的邻域内进行搜索, 以尝试改进当前解。为增强算法的全局探索能力并克服传统方法开发能力弱、收敛速度慢的缺陷, 采用混合差分进化策略来生成新蜜源, 以替代原算法中依赖单维度随机扰动的机制。新蜜源的生成融合了差分进化的变异与交叉操作。变异阶段采用 DE/rand/1 策略, 变异公式如下:

$$U_i = X_{r_1} + F \cdot (X_{r_2} - X_{r_3}) \quad (7)$$

其中 r_1, r_2, r_3 是不同于当前索引 i 的随机整数。 X_{best} 是当前全局最优解。 F 是缩放因子, 通常取值 $[0.4, 1.0]$, 用于控制差分向量的影响力。随后, 通过二项式交叉操作, 将变异向量 U_i 与原蜜源 X_i 进行混合, 最终产生候选新蜜源。

3.1.3. 观察蜂操作

在雇佣蜂完成对解空间的初步探索后, 观察蜂阶段将依据蜜源的适应度进行选择性地开采。为有效平衡局部深度寻优与种群多样性, 将观察蜂划分为两类并设定差异化分工: 其中约 30% 的观察蜂作为“精英开采蜂”, 剩余 70% 则作为“随机开采蜂”。

首先将所有蜜源按适应度排序, 选取排名前 N 的个体构成精英集合; 精英开采蜂将随机从该精英集合中选取一个蜜源进行精细搜索, 而随机开采蜂则采用经典的轮盘赌选择机制, 依据适应度概率从所有蜜源中进行选择, 从而在强化对优质区域开发的同时维持全局探索潜力。

3.1.4. 侦查蜂操作

在传统人工蜂群算法中, 侦查蜂阶段负责维持种群的全局探索能力。当某个蜜源经过连续 $limit$ 次尝试仍未被改进时, 该蜜源将被判定为陷入局部最优区域, 其对应的雇佣蜂会随即转变为侦查蜂, 放弃当前蜜源, 并在整个搜索空间内随机生成一个新蜜源。此机制是算法跳出局部最优、探索新区域的关键保障。然而, 标准机制中的 $limit$ 值为固定常数, 可能导致过早放弃有潜力的解, 或在种群已陷入停滞时仍反应迟缓。因此, 本研究引入一种自适应阈值生成机制, 对侦查蜂阶段进行智能化改进。核心在于将固定阈值 $limit$ 动态化, 使其根据种群实时的多样性状态进行自适应调整。自适应阈值 $limit_{adapt}$ 的计算公式如下:

$$limit_{adapt} = limit_{min} + (limit_{max} - limit_{min}) * D_{norm}, \quad (8)$$

其中, $limit_{min}$ 和 $limit_{max}$ 分别为预设的放弃阈值下界与上界, $D_{norm} \in [0, 1]$ 为当前种群的归一化多样性度量。该度量基于个体在搜索空间中的分布分散程度计算而得, 具体步骤如下:

$$D_{avg} = \frac{2}{NP \times (NP - 1)} \sum_{i=1}^{NP-1} \sum_{j=i+1}^{NP} \sqrt{\sum_{K=1}^D (x_i^K - x_j^K)^2}, \quad (9)$$

$$D_{\max} = \sqrt{\sum_{k=1}^D (U_k - L_k)^2}, \quad (10)$$

$$D_{\text{norm}} = \frac{D_{\text{avg}}}{D_{\max}}. \quad (11)$$

其中, 种群规模为 NP , 每个个体为 D 维向量 $X_i = (x_i^1, x_i^2, \dots, x_i^D)$ 。 D_{avg} 为所有个体两两之间的欧氏距离的平均值, 计算定义域为 $[L_k, U_k]$ 的搜索空间 ($k = 1, 2, \dots, D$) 的最大可能距离, 即对角线长度 D_{\max} 作为归一化基准, D_{norm} 值越大表征种群分布越分散, 多样性越高。此公式建立了阈值与多样性之间的正向关联: 当种群多样性高时, $\text{limit}_{\text{adapt}}$ 趋近于 limit_{\max} , 允许蜜源在各自区域进行更充分的局部搜索, 避免在广泛探索期进行不必要的位置重置; 当多样性降低、种群趋于收敛时, $\text{limit}_{\text{adapt}}$ 减小, 算法将更快地放弃长期无改进的蜜源, 从而更敏锐地触发重启机制以逃离局部最优。一旦蜜源因达到自适应阈值而被放弃, 将采用初始解生成的方式重新生成新蜜源。

3.2. 基于贪心策略的解优化机制

在资源受限项目调度问题中, 总的项目时长主要受限于工期较长的项目和资源需求量较大的项目。第一阶段生成的调度方案中, 往往会出现一些由于任务排序不当和资源没有最大化利用引发的不必要的空闲时间, 为了使调度方案更紧凑, 采用基于贪心策略的解优化机制, 伪代码如算法 1 所示, 具体步骤如下:

步骤 1. 将所有任务根据所需资源的种类进行分类。例如任务 1 只需要资源 1 和资源 4, 对资源 2 和资源 3 的需求为 0, 任务 2 也只需要资源 1 和资源 4, 那么任务 1 和任务 2 为一类任务。任务 3 需要资源 1 和资源 3, 那么任务 3 和任务 1、任务 2 不是一类任务。将资源需求不同的任务分别放在不同的待分配任务池中, 例如任务 1 和任务 2 就可以放在待分配任务池 1 中。

步骤 2. 将处于待分配任务池中的任务, 分别根据任务工期长短和任务开始时间进行排列。

步骤 3. 在待分配任务池内部进行任务交换, 以减少交换后为不可行解的概率。遵循两个交换原则, 分别是: 任务开始时间靠后的任务与任务开始时间靠前的任务相交换; 工期较长的任务和工期较短的任务相交换。

步骤 4. 利用修复函数验证新解的可行性。如果为不可行解, 那么将二者从待分配任务池中剔除, 并返回步骤 3, 重新进行任务交换; 如果是可行解, 那么记录新解的适应度。

步骤 5. 当交换次数达到设定的最大交换次数时, 交换终止。并输出该过程中适应度最小的作为最优解。

Algorithm 1. Task exchange optimization procedure

算法 1. 任务交换优化机制

输入: 任务列表 T , 资源集合 R , 最大交换次数 max_swaps

输出: 优化后的调度方案 S^*

```

1: // 步骤 1: 按资源需求类别分类任务
2: 任务池集合  $\leftarrow \{\}$ 
3: for each  $t$  in  $T$  do
4:   if  $t$ .资源需求类别 not in 任务池集合 then
5:     任务池集合[ $t$ .资源需求类别]  $\leftarrow \{\}$ 
6:     任务池集合[ $t$ .资源需求类别].append( $t$ )
7:
8: 当前解  $\leftarrow$  初始调度( $T$ )

```

续表

```

9: 最优解  $\leftarrow$  当前解
10: 最优适应度  $\leftarrow$  计算适应度(当前解)
11: swap_count  $\leftarrow$  0
12:
13: while swap_count < max_swaps do
14:     // 步骤 2: 排序每个任务池
15:     for each pool in 任务池集合.values() do
16:         按开始时间升序排序(pool)
17:         按工期升序排序(pool)
18:
19:     // 步骤 3: 任务交换
20:     for each pool in 任务池集合.values() do
21:         if len(pool) < 2 then continue
22:
23:          $t_1 \leftarrow$  pool[0] // 开始时间最早
24:          $t_2 \leftarrow$  pool[-1] // 开始时间最晚
25:
26:         // 步骤 4: 验证可行性
27:         候选解  $\leftarrow$  交换任务(当前解,  $t_1$ ,  $t_2$ )
28:         if 验证可行性(候选解) then
29:             if 计算适应度(候选解) < 最优适应度 then
30:                 最优解  $\leftarrow$  候选解
31:                 最优适应度  $\leftarrow$  计算适应度(候选解)
32:                 当前解  $\leftarrow$  候选解
33:             else
34:                 // 从任务池中移除不可行任务
35:                 pool.remove( $t_1$ )
36:                 pool.remove( $t_2$ )
37:
38:         swap_count  $\leftarrow$  swap_count + 1
39:         if swap_count  $\geq$  max_swaps then break
40:
41: return 最优解

```

4. 实验验证

为了测试混合贪心策略人工蜂群的两阶段算法在资源受限的项目调度问题上的求解能力, 选用经典的 PSPLIB 标准实例库(<http://www.om-db.wi.tum.de/psplib>)作为数据集, 该数据集包含任务节点数分别为 30、60、90 和 120 的四类调度问题, 每类问题中的可更新资源数量固定为 4。各子集(J30, J60, J90, J120)所含问题实例数量依次为 480、480、480 和 600 个, 即总实例数为 2040 个。采用对于不同规模数据的最优解覆盖度作为对比指标, 目前仅 j_30 数据集有当前最优解, j_60、j_90 和 j_120 都只有启发式最优解。

为实验设定最大迭代次数为 500, 人工蜂群算法的蜜源数量和差分进化算法的种群数量均为 50。将混合贪心策略人工蜂群的两阶段算法的实验结果加粗, 并与传统的人工蜂群算法, 差分进化算法以及分支定价算法[20]以及进行比较, 实验比较结果汇总在表 1, 选择算例 J3012_7 作为示例, 生成的调度方案甘特图如图 2 所示。混合贪心策略人工蜂群的两阶段算法在 4 个规模的数据集上求解出的最优解个数分别为 378, 325, 303 和 89, 最优解覆盖度为 53.68%, 相比较改进前的人工蜂群算法提升了 5.93%。经典的差分进化算法和分支定价算法的最优解覆盖度分别为 46.96% 和 43.35%, 分别低于混合贪心策略人工蜂群的两阶段算法 6.72% 和 11.32%。算例 J3012_7 的总工期为 55, 与当前已知的最优解一致。

Table 1. Optimal solution coverage rate
表 1. 最优解覆盖度

	工序个数				最优解覆盖度
	30	60	90	120	
混合贪心策略人工蜂群的两阶段算法	378	325	303	89	53.68%
人工蜂群算法	314	307	286	67	47.75%
差分进化算法	326	284	290	58	46.96%
分支定价算法	301	269	263	31	42.35%

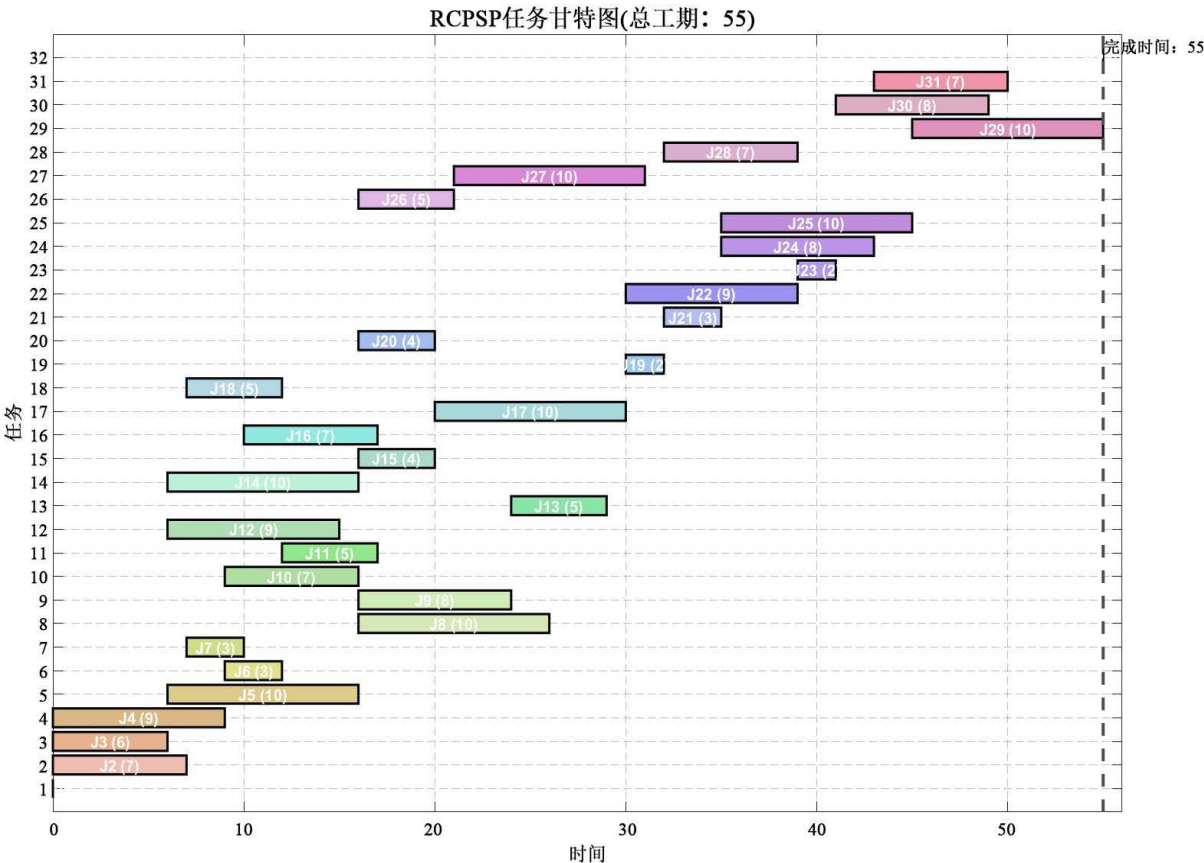


Figure 2. Gantt chart for test case J3012_7
图 2. 算例 J3012_7 甘特图

5. 工程项目实例

为了验证算法的有效性和可信度，选择了一个现实的工程项目案例[21]，案例的详细信息如表 2 所示。案例一共 12 个工序，其中包含开始与结束的虚工作，即 $n=12$ 。根据现场情况与施工组织计划得到各工序之间的前序任务约束、工期、资源需求量，如表 2 所示。可更新资源 4 种($K=4$)，资源最大可使用量分别为(16, 20, 12, 9)。

图 3 的甘特图是采用混合贪心策略人工蜂群的两阶段算法生成的调度计划。可以看出，由结束的虚任务标志的任务工期为 24 天，达到了已知最小工期。

Table 2. Project instance parameters
表 2. 工程项目实例参数

工序名称	前序任务	资源消耗(R1、R2、R3、R4)	持续时间
A1	空	0, 0, 0, 0	0
A2	1	3, 2, 2, 1	4
A3	1	12, 14, 1, 4	10
A4	1	9, 12, 1, 4	8
A5	1	5, 4, 6, 2	4
A6	2	3, 2, 4, 1	2
A7	6	1, 2, 1, 5	2
A8	7	8, 7, 12, 3	12
A9	4	3, 3, 9, 2	2
A10	3	3, 3, 6, 2	2
A11	5、7	2, 12, 0, 3	6
A12	11	0, 0, 0, 0	0

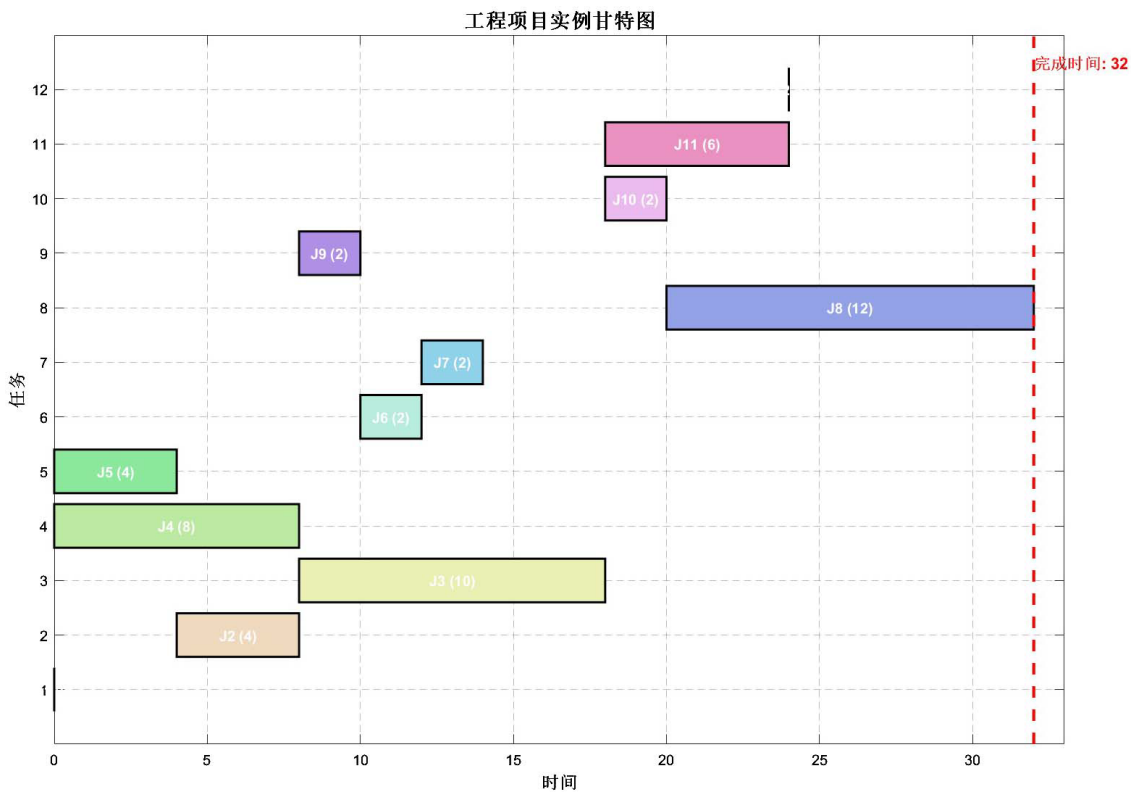


Figure 3. Gantt chart for the engineering project case
图 3. 工程项目实例甘特图

6. 结果与展望

针对资源受限项目调度问题(RCPSP), 提出了一种混合贪心策略人工蜂群的两阶段算法。该算法第一阶段采用改进的人工蜂群算法生成初始解, 并在第二阶段引入贪心策略进行再优化, 有效提升了算法整

体性能。在基准数据集 PSPLIB 上进行实验, 并与分支定界算法、差分进化算法及传统人工蜂群算法进行对比。最后将该算法应用在工程项目实例中, 得到项目调度方案。

基于上述研究, 形成以下结论:

(1) 混合贪心策略人工蜂群的两阶段算法, 通过两阶段协同优化机制, 提高了解决方案的质量, 在 PSPLIB 数据集上的实验结果表明, 该算法的最优解覆盖率分别优于其它三种算法 5.93%, 6.72% 和 11.32%。

(2) 在实际案例应用中, 该算法成功求得已知最优解, 并输出了可行的项目调度方案与直观的甘特图, 验证了其在现实场景中的适用性与有效性。

(3) 该算法为 RCPSP 提供了一种新的有效求解思路, 也为智能优化算法与实际调度问题的结合提供了可参考的实现路径。

尽管混合贪心策略人工蜂群的两阶段算法在测试中表现出良好性能, 未来仍可在以下方面进一步探索: 一是针对更复杂的资源约束情形进行算法适应性改进; 二是尝试引入更多元化的局部搜索机制, 以平衡算法收敛速度与求解精度; 三是将该混合策略拓展至其他类型的调度或组合优化问题, 检验其通用性与可扩展性。

参考文献

- [1] Blazewicz, J., Lenstra, J.K. and Kan, A.H.G.R. (1983) Scheduling Subject to Resource Constraints: Classification and Complexity. *Discrete Applied Mathematics*, **5**, 11-24. [https://doi.org/10.1016/0166-218x\(83\)90012-4](https://doi.org/10.1016/0166-218x(83)90012-4)
- [2] 孙泽浩, 李颖, 白丹宇, 等. 离散人工蜂群算法求解手术室护士排班问题[J]. 重庆师范大学学报(自然科学版), 2025, 42(2): 38-48.
- [3] 宋启航, 张丽. 基于改进启发式算法的机组优化排班问题研究[J]. 智能计算机与应用, 2024, 14(2): 55-61.
- [4] Nigar, N., Shahzad, M.K., Islam, S., Oki, O. and Lukose, J.M. (2023) Multi-Objective Dynamic Software Project Scheduling: A Novel Approach to Handle Employee's Addition. *IEEE Access*, **11**, 39792-39806. <https://doi.org/10.1109/access.2023.3265716>
- [5] Türkakın, O.H., Arditi, D. and Manisalı, E. (2021) Comparison of Heuristic Priority Rules in the Solution of the Resource-Constrained Project Scheduling Problem. *Sustainability*, **13**, Article 9956. <https://doi.org/10.3390/su13179956>
- [6] Pellerin, R., Perrier, N. and Berthaut, F. (2020) A Survey of Hybrid Metaheuristics for the Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, **280**, 395-416. <https://doi.org/10.1016/j.ejor.2019.01.063>
- [7] Sallam, K.M., Chakraborty, R.K. and Ryan, M.J. (2021) A Reinforcement Learning Based Multi-Method Approach for Stochastic Resource Constrained Project Scheduling Problems. *Expert Systems with Applications*, **169**, Article 114479. <https://doi.org/10.1016/j.eswa.2020.114479>
- [8] Brucker, P., Knust, S., Schoo, A. and Thiele, O. (1998) A Branch and Bound Algorithm for the Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, **107**, 272-288. [https://doi.org/10.1016/s0377-2217\(97\)00335-4](https://doi.org/10.1016/s0377-2217(97)00335-4)
- [9] Talbot, F.B. and Patterson, J.H. (1978) An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems. *Management Science*, **24**, 1163-1174. <https://doi.org/10.1287/mnsc.24.11.1163>
- [10] 白羽, 宁飒, 王坚. 基于员工满意度和人力成本的智能楼宇保洁工人排班研究[J]. 北京建筑大学学报, 2025, 41(1): 46-52.
- [11] 王海鑫, 王祖和, 温国锋, 等. 自适应粒子群算法求解资源受限多项目调度问题[J]. 管理工程学报, 2017, 31(4): 220-225.
- [12] 张博, 陈志敏, 张利平. 基于改进遗传算法的船舶维修项目调度问题研究[J]. 中国修船, 2024, 37(2): 36-39.
- [13] Merkle, D., Middendorf, M. and Schmeck, H. (2002) Ant Colony Optimization for Resource-Constrained Project Scheduling. *IEEE Transactions on Evolutionary Computation*, **6**, 333-346. <https://doi.org/10.1109/tevc.2002.802450>
- [14] Karaboga, D. (2005) An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Department of Computer Engineering, Engineering Faculty, Erciyes University.

-
- [15] Crawford, B., Soto, R., Johnson, F., Norero, E. and Olguín, E. (2015) An Artificial Bee Colony Algorithm for the Resource Constrained Project Scheduling Problem. In: Stephanidis, C. Ed., *Communications in Computer and Information Science*, Springer International Publishing, 582-586. https://doi.org/10.1007/978-3-319-21380-4_99
 - [16] Shi, Y., Qu, F., Chen, W. and Li, B. (2010) An Artificial Bee Colony with Random Key for Resource-Constrained Project Scheduling. In: Li, K., Fei, M., Jia, L. and Irwin, G.W. Eds., *Lecture Notes in Computer Science*, Springer, 148-157. https://doi.org/10.1007/978-3-642-15597-0_17
 - [17] Chen, M. (2019) Improved Artificial Bee Colony Algorithm Based on Escaped Foraging Strategy. *Journal of the Chinese Institute of Engineers*, **42**, 516-524. <https://doi.org/10.1080/02533839.2019.1611479>
 - [18] 熊福力, 陈思远, 熊宁馨, 等. 基于两阶段混合迭代贪婪算法的分布式异构非置换流水车间调度[J]. 计算机集成制造系统, 2025, 31(8): 2870-2883.
 - [19] 唐媛媛, 张双武, 曾宇清. 基于两阶段混合算法的卷烟车间排产优化[J]. 工业控制计算机, 2024, 37(7): 132-133+141.
 - [20] 张宇哲, 董兴业, 周正. 求解资源受限项目调度问题的分支定价算法[J]. 计算机科学, 2022, 49(12): 274-282.
 - [21] 齐国庆. 资源受限条件下施工项目调度多目标优化研究[D]: [硕士学位论文]. 济南: 山东建筑大学, 2024.