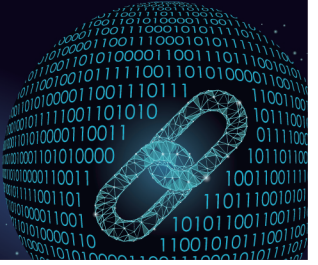




中国科技期刊卓越行动计划项目入选期刊

# 控制与决策

CONTROL AND DECISION



## 极小化加权完工时间的多资源工序的资源分配问题

翁武燕, 储诚斌, 吴鹏

引用本文:

翁武燕, 储诚斌, 吴鹏. 极小化加权完工时间的多资源工序的资源分配问题[J]. 控制与决策, 2024, 39(8): 2765–2772.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.1969>

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 基于双种群模糊引力搜索算法的舰载机甲板作业调度

Flight deck operations scheduling based on dual population fuzzy gravitational search algorithm

控制与决策. 2021, 36(11): 2751–2759 <https://doi.org/10.13195/j.kzyjc.2020.0523>

#### 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>

#### 带不相关并行机和有限缓冲MHFS调度的混合启发式算法

Hybrid heuristic algorithm for multi-stage hybrid flow shop scheduling with unrelated parallel machines and finite buffers

控制与决策. 2021, 36(3): 565–576 <https://doi.org/10.13195/j.kzyjc.2019.0835>

#### 铁路集装箱中心站资源分配与作业调度联合优化

Integrating optimization of resource allocation and handling scheduling in railway container terminal

控制与决策. 2021, 36(12): 3063–3073 <https://doi.org/10.13195/j.kzyjc.2020.0597>

#### 一种自适应拟牛顿-状态转移混合智能优化算法及应用

A hybrid state transition optimization algorithm based on adaptive quasi-newton method and its application

控制与决策. 2021, 36(10): 2451–2458 <https://doi.org/10.13195/j.kzyjc.2020.0214>

# 极小化加权完工时间的多资源工序的资源分配问题

翁武燕, 储诚斌<sup>†</sup>, 吴 鹏

(福州大学 经济与管理学院, 福州 350108)

**摘 要:** 针对现实中广泛存在的多资源工序的资源分配问题, 考虑基于资源使用的优先次序约束, 以最小化加权完工时间为优化目标, 构建一类新的资源分配混合整数线性规划模型. 其次, 提出 Benders 分解和禁忌搜索的混合算法, 该混合算法以 Benders 分解为基本框架, 将原问题分为提供资源分配方案的主问题和计算工序加权完工时间的子问题, 并通过改进数学模型和添加禁忌搜索提高混合算法的收敛速度. 最后, 通过 300 个随机仿真算例测试结果表明, 在相同时间下求解小规模问题时, 所提的 Benders 分解混合算法能获得距离商业求解器 CPLEX 最优解平均差距为 0.86% 的满意解; 在求解大规模问题时, 所提出的算法的性能表现优于 CPLEX、禁忌搜索算法、变邻域搜索算法和 Benders 分解嵌入遗传算法的混合方法, 能给出更好的资源分配方案, 与 CPLEX 相比, 上界和下界分别改善了 4.74% 和 9.62%.

**关键词:** 资源分配; 多资源工序; 基于资源使用的先后次序; Benders 分解; 禁忌搜索算法

中图分类号: O224

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.1969

**引用格式:** 翁武燕, 储诚斌, 吴鹏. 极小化加权完工时间的多资源工序的资源分配问题[J]. 控制与决策, 2024, 39(8): 2765-2772.

## Resource allocation to minimize the weighted completion time with multi-resource operations

WENG Wu-yan, CHU Cheng-bin<sup>†</sup>, WU Peng

(School of Economics and Management, Fuzhou University, Fuzhou 350108, China)

**Abstract:** This paper addresses a resource-allocation problem extracted from real life application involving multi-resource operations. A new mixed integer linear programming model is proposed to minimize the weighted completion time while considering resource-related precedence relationships. Then, a hybrid algorithm combining Benders decomposition and Tabu search is developed based on Benders decomposition as the basic framework. This method divides the original problem into a master problem for resource allocation and a subproblem of calculating the completion time of each operation. The convergence is sped up by improving the mathematical model and embedding the Tabu search approach. The experimental results on 300 randomly generated instances show that when solving small-scale problems, the proposed hybrid algorithm can yield satisfactory solutions with an average deviation of 0.86% from optimal ones provided by the commercial CPLEX solver; when solving large-scale problems, the proposed algorithm outperforms the CPLEX solver, the pure Tabu search algorithm, the variable neighborhood search algorithm and the Benders decomposition with embedded genetic algorithm. Compared with the CPLEX, the upper bound and lower bound are improved by 4.74% and 9.62% respectively.

**Keywords:** resource allocation; multi-resource operation; the resource-related precedence relationships; Benders decomposition; Tabu search

## 0 引言

出于对外部激烈竞争环境的担忧, 企业比以往任何时候都更热衷于提高生产效率, 追求生产效率

的主要策略包括投资新资源和提高资源利用率<sup>[1]</sup>. 然而, 受限于成本、场地或其他因素, 可用于生产资源的数量是有限的. 就经济角度而言, 可靠高效的资

收稿日期: 2022-11-14; 录用日期: 2023-06-16.

**基金项目:** 国家自然科学基金项目(71871159, 71701049, 71901069); 教育部人文社科基金一般项目(21YJA630096); 福建“雏鹰计划”青年拔尖人才项目(0470-00472214); 福建省自然科学基金面上项目(2022J01075); 福建省科技经济融合服务平台项目(0300-82321069).

**责任编委:** 刘士新.

<sup>†</sup>通讯作者. E-mail: chengbin.chu@gmail.com.

源分配方案是工业生产、项目管理和经济等领域企业所亟需的. 在实际的车间生产环境中, 广泛存在需要操作员、机床和辅助设备等多类(可重复使用)资源共同协作完成的多资源工序(multi-resource operation, MRO)<sup>[2]</sup>. 值得注意的是, 这类以多资源为特征的工序不限于制造业的生产工序, 也泛指服务业中诸多场景. 以篮球赛事为例, 同时需要裁判员、运动员和记分员等共同参与, 也可将其定义为多资源工序. 现实中每类资源往往不止一个, 从而产生为每个工序分配所需各类资源的资源分配问题<sup>[3]</sup>. 在实际生产过程中, 若任意两个工序需同一类资源, 则其可能共用同一个资源, 且每个资源在任何时刻只能执行一个工序. 因此, 在实际生产过程中企业决策者为方便管理, 通常会对使用同一个资源的工序提前排序. 如上所述, 研究考虑资源使用优先次序的多资源工序的资源分配优化具有重要的现实意义.

现有资源分配问题的研究<sup>[4-8]</sup>大多仅考虑单一资源工序. 这类研究的隐含假设是将多个甚至多类资源简化为一个资源, 例如: 将操作员和机器设备固定组合简化为一个机器(资源)<sup>[9-10]</sup>. 这种简化虽能极大地降低问题的求解难度, 但会造成资源浪费且与企业实际的需求相差甚远. 与之相反, 多资源工序的资源分配问题的研究虽然较少, 但可避免上述强制固定的隐形假设, 进而提高资源使用的柔性, 为企业提供理论和实践参考. 因此, 为更加贴合实际情况, 学者们根据实际生产遇到的问题, 对多资源工序的资源问题进行了扩展和变形. 针对可重入生产系统, Desprez等<sup>[11]</sup>研究了现实中一个大规模且具有多资源特征工序的可重入生产系统问题, 并以最小化延迟任务数量为优化目标开发了遗传算法; 在此基础上, Chu等<sup>[12]</sup>考虑了资格匹配的约束, 并开发了爬山算法和自适应遗传算法两种启发式算法; 考虑到动态环境下的资源分配问题, Huh等<sup>[13]</sup>研究了可择期工作和紧急工作的多资源分配问题, 紧急工作需要立即执行, 而可择期工作可以等待, 主要优化目标是及时满足择期工作的需求, 并留出足够的容量以满足紧急需求, 还提出了近似的求解策略; 关叶青等<sup>[14]</sup>研究了多周期生产的资源动态分配问题, 并开发了遗传算法; 此外, Xie等<sup>[15]</sup>研究了资源均衡使用的资源分配问题, 并基于实际数据进行建模分析优化; 在多目标优化的研究中, 桓自强等<sup>[16]</sup>研究了资源受限环境中多任务间的多资源分配问题, 并提出了一种高级人工鱼群算法; 王悦宸等<sup>[17]</sup>设置了3种不同的资源分配优化目标: 效果优先目标、效率优先目标和执行时间最短目标, 3个优化目标的比重还可进行调整; 覃炳发等<sup>[18]</sup>

在资源分配的基础上, 研究了资源选址问题. 然而, 目前尚未有文献针对现实中存在的基于资源使用的优先次序约束进行研究, 该基于资源使用的优先次序是指任意两道工序若共享同一个资源则必须遵循的优先次序.

目前的研究成果在一定程度上推动和丰富了多资源工序的资源分配研究, 但仍存在如下不足: 1) 现有相关研究大多数仅考虑使用单一资源的工序; 2) 现有文献采用启发式算法求解较多, 使用精确算法较少, 不能保证收敛时解的最优性且获得解的质量难以评估; 3) 现有文献尚未发现考虑现实中存在的基于资源使用的优先次序约束的研究. 因此, 考虑到企业面临的实际问题, 进一步提炼现实中资源使用特征, 以最小化加权完工时间为优化目标, 建立一类新的考虑基于资源使用次序的多资源工序的资源分配数学模型. 为获得全局最优解, 本文根据问题的特性设计一类Benders分解精确算法, 即在Benders分解(Benders decomposition, BD)算法<sup>[19]</sup>中嵌入禁忌搜索(Tabu search, TS)<sup>[20]</sup>的混合算法(Benders decomposition embedded Tabu search hybrid algorithm, BDTS). 该混合算法以Benders分解算法为框架, 针对精确算法在大规模仿真算例中求解时间呈指数级增长的问题, 对数学模型进行改进并利用禁忌搜索算法局部搜索能力提升寻找最优解的效率. 最后, 通过大规模的随机仿真算例验证模型和算法的有效性.

## 1 问题建模

### 1.1 问题描述

本文采用析取图 $G = (V, E_1, E_2)$ 表示所研究的问题,  $V = \{0, n+1\} \cup J$ 表示顶点集合. 其中:  $J$ 表示工序集合; 顶点0和 $n+1$ 分别表示虚拟的起始工序和终止工序;  $E_1$ 表示由工艺约束关系产生的工序间先后顺序的析取弧(下称实际弧)的集合, 由于工艺约束的固定性, 实际弧表示的先后顺序为无条件约束, 即在任何情况下都必须遵守的约束; 此外, 基于资源使用的优先次序用潜在弧 $E_2$ 表示, 基于资源使用的次序当且仅当两个相关工序共享同一个资源时, 工序间必须遵守给定的先后次序, 即为有条件下才需遵守的约束. 因此, 本文基于现实中多资源工序的资源分配问题, 研究有条件约束与无条件约束间的相互影响, 并制定高效的资源分配方案.

### 1.2 假设条件

1) 任意一个资源只属于一种类型, 且任意一类资源至少有两个; 2) 每个工序至多需要每类资源中

的一个;3) 每个资源在任何时刻只能执行一个工序;  
4) 每道工序仅当所需资源全部到位时才能开始加工;  
5) 工序不能抢占: 工序一旦开始便不能中断;  
6) 对于任意两道需要同一类别资源的工序, 共用同一个资源时的先后顺序已知;  
7) 所有工序都在0时刻发布。

### 1.3 符号定义

$J$ : 工序集合。

$K$ : 资源类别集合。

$m_k$ : 类别为  $k$  的资源数量。

$Q_j$ : 工序  $j$  所需的资源类型的集合。

$d_j$ : 工序  $j$  的加工时间。

$w_j$ : 工序  $j$  的权重。

$M$ : 任意大的正数。

$x_j$ : 工序  $j$  完工时刻。

$y_{j,k,l}$ : 0-1 变量, 若类别为  $k$  的资源中的第  $l$  个资源分配给工序  $j$ , 则为1; 否则为0。

### 1.4 模型建立

根据上述描述, 本节对多资源工序的资源分配问题建立如下混合整数线性规划模型:

$$\min \sum_{j=1}^J w_j x_j. \quad (1)$$

$$\text{s.t.} \sum_{l=1}^{m_k} y_{j,k,l} = 1, \forall j \in J, \forall k \in Q_j; \quad (2)$$

$$x_j - x_i \geq d_j, \forall (i, j) \in E_1; \quad (3)$$

$$x_j - x_i \geq d_j + M(y_{ikl} + y_{jkl} - 2), \quad \forall (i, j) \in E_2, \forall k \in Q_i \cap Q_j, 1 \leq l \leq m_k; \quad (4)$$

$$x_j \geq d_j, \forall j \in J; \quad (5)$$

$$y_{j,k,l} \in \{0, 1\}, \forall j \in J, \forall k \in Q_j, 1 \leq l \leq m_k. \quad (6)$$

该数学模型描述: 在满足各多资源工序的资源需求和先后次序的前提下, 使得所有工序的加权完工时间和最小。式(1)为优化目标, 表示总加权完工时间最小化; 式(2)表示必须满足多资源工序所需的各类资源; 式(3)表示工序间在任何情况下都须遵守的加工顺序约束; 式(4)表示任意两个工序使用同一个资源时, 必须满足基于资源使用次序的约束; 式(5)和(6)表示决策变量的取值范围。

单资源工序的资源分配问题是NP难问题<sup>[21]</sup>, 而多资源工序的资源分配问题是其扩展研究, 因而也属于NP难问题。

## 2 算法设计

Benders 分解算法是求解具有复杂变量混合整数规划问题常用的精确算法之一, 相较于启发式算

法仅能获得问题的上界, Benders 分解算法不仅能保证收敛时解的最优性, 也能在可搜寻迭代过程中同时获得问题的上界和下界, 进而通过下界对上界的优劣做出准确的判断。利用 Benders 分解框架, 本文将原问题(模型)分解为一个提供资源分配方案的仅包含整数变量的主问题和一个整数变量给定后用以求算工序加权完工时间的仅包含连续变量的对偶子问题。由于对偶子问题属于一类仅含连续变量的线性规划(linear programming, LP)问题, 传统单纯形法能快速精确求解。因此, 本文直接调用 CPLEX 的 LP 求解器对对偶子问题进行最优求解。主问题和对偶子问题的最优目标函数值分别对应原问题的下界和上界。在通常的 Benders 分解算法搜寻过程中, 将主问题求解得到的资源分配方案(即整数变量的取值)直接代入对偶子问题, 然后精确求解对应偶子问题获得 Benders 割(Benders-cut)并添加至主问题, 反复迭代求解不断改进下界和上界, 直至获得问题的最优解。

为了加快 Benders 分解的收敛速度, 本文还提出改进的数学模型和在 Benders 框架中添加禁忌搜索的提升策略。首先, 通过对原数学模型进行改进, 有效减少  $M$  约束的数量。此外, 在每次迭代中, 求解完主问题模型, 即可获得一个资源分配方案, 通过对该资源分配方案进行禁忌搜索, 能有效提升上界和下界的质量。BDTS 混合算法整体流程如图1所示。

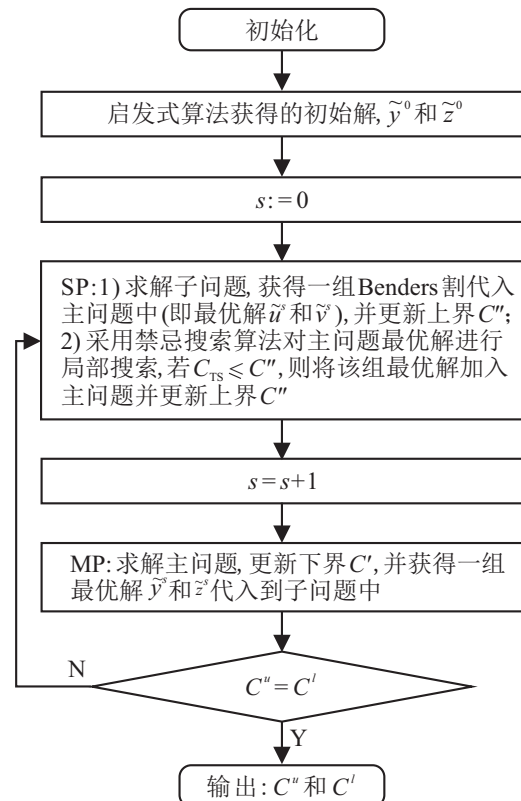


图1 BDTS 混合算法求解流程

## 2.1 初始解

高质量的初始解能够有效地减小BDTS算法的解空间而加快算法的求解速度. 因此, 本文根据问题特征开发出能够快速获得较好质量的初始解的启发式算法. 该启发式主要根据贪婪的启发式思想, 将无紧前工序或紧前工序已安排的工序添加到可选工序集并选取可开工时间最早的工序进行安排. 同时, 为尽可能提高资源的柔性, 从被选取工序所需的各类资源中挑选可用时间与该工序可开工时间最近的资源.

通过上述的启发式可以得到质量较好的初始解以及该初始解对应的资源分配方案.

## 2.2 改进数学模型

减少主问题的 $M$ 约束是提升Benders算法的一种有效途径. 本节通过对原问题数学模型进行改进来减小 $M$ 约束的数量.

设置一个0-1变量 $z_{i,j}$ ,  $(i, j) \in E_2$ . 因此, 式(4)可改为

$$\begin{aligned} z_{i,j} &\geq y_{j,k,l} + y_{i,k,l} - 1, \forall (i, j) \in E_2, \\ \forall k \in Q_i \cap Q_j, 1 \leq l \leq m_k; \end{aligned} \quad (7)$$

$$x_j - x_i \geq d_j + M(z_{i,j} - 1), \forall (i, j) \in E_2. \quad (8)$$

式(8)中的 $M$ 是为保证当 $z_{i,j} = 0$ 时, 约束为冗余. 为了进一步收紧式(8)的约束, 可以取一个较小的值代替 $M$ . 在图 $G$ 中, 指向顶点 $j$  ( $j \in J$ )的弧长为工序 $j$ 的加工时间. 因此, 所研究的目标函数本质上为图 $G$ 中从顶点0到各顶点 $j$  ( $j \in J$ )最长路径的加权和. 为了保证当 $z_{i,j} = 0$ 时, 约束仍为冗余, 只需取一个最优解时工序 $i$ 与 $j$ 完工时间之差的下界即可. 设 $\gamma_j$  ( $j \in J$ )为在图 $(V, E_1)$ 中从顶点0到顶点 $j$ 的最长路径, 其代表工序 $j$ 完工时间的下界; 而 $\delta_j$  ( $j \in J$ )则为图 $(V, E_1)$ 中从顶点 $j$ 到 $n+1$ 的最长路径, 代表终止工序 $n+1$ 完工时间与工序 $j$ 完工时间之差的下界. 这些值可通过Bellman-Ford算法<sup>[22]</sup>得到. 值得注意的是 $\sum_{j=1}^J w_j \gamma_j$ 为本问题最优目标函数值的一个下界. 此外, 设 $\bar{C}$ 为终止工序的完工时间, 其可通过启发式算法获得. 因此有

$$\gamma_j \leq x_j \leq \bar{C} - \delta_j, \forall j \in J; \quad (9)$$

$$x_j - x_i \geq \gamma_j + \delta_i - \bar{C}, \forall (i, j) \in E_2. \quad (10)$$

由此可知, 式(8)可改为

$$\begin{aligned} x_j - x_i &\geq d_j + (\gamma_j + \delta_i - \bar{C} - d_j)(1 - z_{i,j}), \\ \forall (i, j) &\in E_2. \end{aligned} \quad (11)$$

改进的完整数学模型如下所示:

$$\min \sum_{j=1}^J w_j x_j. \quad (12)$$

$$\text{s.t.} \sum_{l=1}^{m_k} y_{j,k,l} = 1, \forall j \in J, \forall k \in Q_j; \quad (13)$$

$$x_j - x_i \geq d_j, \forall (i, j) \in E_1; \quad (14)$$

$$\begin{aligned} x_j - x_i &\geq d_j + (\gamma_j + \delta_i - \bar{C} - d_j)(1 - z_{i,j}), \\ \forall (i, j) &\in E_2; \end{aligned} \quad (15)$$

$$\begin{aligned} z_{i,j} &\geq y_{j,k,l} + y_{i,k,l} - 1, \forall (i, j) \in E_2, \\ \forall k \in Q_i \cap Q_j, 1 &\leq l \leq m_k; \end{aligned} \quad (16)$$

$$x_j \geq d_j, \forall j \in J; \quad (17)$$

$$z_{i,j} \in \{0, 1\}, \forall (i, j) \in E_2; \quad (18)$$

$$y_{j,k,l} \in \{0, 1\}, \forall j \in J, \forall k \in Q_j, 1 \leq l \leq m_k. \quad (19)$$

## 2.3 主问题

主问题(master problem, MP)模型在第 $r$  ( $r \geq 1$ )次迭代后如下所示:

$$\min \eta. \quad (20)$$

$$\text{s.t.} \sum_{l=1}^{m_k} y_{j,k,l} = 1, \forall j \in J, \forall k \in Q_j; \quad (21)$$

$$\begin{aligned} z_{i,j} &\geq y_{j,k,l} + y_{i,k,l} - 1, \forall (i, j) \in E_2, \\ \forall k \in Q_i \cap Q_j, 1 &\leq l \leq m_k; \end{aligned} \quad (22)$$

$$\begin{aligned} \eta &\geq \sum_{(i,j) \in E_2} [d_j + (\gamma_j + \delta_i - \bar{C} - d_j)(1 - z_{i,j})] \tilde{v}_{i,j}^p + \sum_{j \in V} d_j \tilde{u}_j^p + \sum_{(i,j) \in E_1} d_i \tilde{v}_{i,j}^p, \quad p \leq r; \end{aligned} \quad (23)$$

$$z_{i,j} \in \{0, 1\}, \forall (i, j) \in E_2; \quad (24)$$

$$y_{j,k,l} \in \{0, 1\}, \forall j \in J, \forall k \in Q_j, 1 \leq l \leq m_k. \quad (25)$$

MP包括与0-1变量有关的约束和变量, 并引入辅助变量 $\eta$ 表示主问题的目标函数. 其中,  $u^p$ 和 $v^p$ 是通过求解子问题的对偶模型在第 $p$ 次迭代中获得的.

设 $\hat{y}^r$ 和 $\hat{z}^r$ 为主问题在第 $r$ 次迭代中获得的最优解.

## 2.4 对偶子问题

问题模型为式(12)、(14)、(15)和(17). 令 $E_2^s = \{(i, j) | (i, j) \in E_2, \tilde{z}_{i,j}^s = 1\}$ , 其中 $\tilde{z}^0$ 为通过启发式算法获得的初始解. 子问题的对偶模型在第 $s$  ( $s \geq 0$ )次迭代后如下所示:

$$\max \sum_{(i,j) \in E_2} [d_j + (\gamma_j + \delta_i - \bar{C} - d_j)(1 - z_{i,j})]v_{i,j} + \sum_{j \in J} d_j u_j + \sum_{(i,j) \in E_1} d_i v_{i,j}. \quad (26)$$

$$\text{s.t. } u_j + \sum_{(i,j) \in E_1 \cup E_2^s} v_{i,j} - \sum_{(j,i) \in E_1 \cup E_2^s} v_{j,i} = w_j, \quad \forall j \in J; \quad (27)$$

$$u_j \geq 0, \quad \forall j \in J; \quad (28)$$

$$v_{i,j} \geq 0, \quad \forall (i,j) \in E_1 \cup E_2. \quad (29)$$

设  $C_s^u$  为对偶子问题在第  $s$  次迭代中求得的最优目标值;  $\eta_s$  为主问题第  $s$  次迭代时得到的最优目标值;  $C^*$ 、 $C^u$  和  $C^l$  分别为原问题的最优解、上界和下界,  $C^u$  和  $C^l$  可以在第  $s$  次迭代后如下式进行更新:

$$\begin{cases} C^u \leftarrow \min(C_s^u, C^u), \\ C^l \leftarrow \eta_s, \\ C^u \geq C^* \geq C^l. \end{cases} \quad (30)$$

## 2.5 禁忌搜索算法

尽管原来  $M$  约束中的  $M$  值已被更为接近实际的值所代替,但这些冗余约束的存在仍然限制主问题的求解速度. 值得指出的是,主问题求解得到的资源分配方案 ( $\hat{y}$  和  $\hat{z}$ ) 会直接影响子问题求解后获得 Benders 割的好坏,进一步改进主问题求解得到的资源分配方案有助于生成更紧的 Benders 割<sup>[23]</sup>,提升上界和下界的质量,从而减少算法的迭代次数并提高算法的收敛速度. 因此,本文采用禁忌搜索算法对主问题求解所得的资源分配方案进行进一步改进以提升 Benders 分解算法的性能. 具体过程分析说明如下.

主问题在每次迭代后生成一个资源分配方案 ( $\hat{z}$  和  $\hat{y}$ , 其中  $\hat{z}$  可基于式 (22) 用  $\hat{y}$  计算), 本文采用禁忌搜索算法对该资源分配方案 ( $\hat{y}$ ) 进行局部搜索, 寻找到更优的资源分配方案, 代入到子问题模型中计算更好的上界并获得一个额外具有更高质量的 Benders 割 ( $\tilde{u}$  和  $\tilde{v}$ ). 此外, 由 (23) 可知, 通过嵌入禁忌搜索算法获得的较好的  $\bar{C}$  和更紧的 Benders 割也可有效提升下界.

### 2.5.1 邻域设计

由 MP 可知各工序选取的资源序号, 因而, 禁忌搜索算法解的形式是基于工序使用资源的方式构造的, 以工序的序号对使用的资源进行排序, 每个工序占  $K$  个位置, 设  $H_{K(j-1)+k}$  ( $j \in J, k \in K$ ) 代表工序  $j$  使用第  $k$  类资源所用的资源序号. 例如, 当  $K = 3$  时, 工序 1 分配到第 1 类资源中的第 2 个, 则  $H_1 = 2$ , 工序 3 分配到第 3 类资源中第 1 个, 则  $H_9 = 1$ , 若工序

$j$  不需要第  $k$  类资源, 则令  $H_{K(j-1)+k} = 0$ . 因此, 本部分解的长度为  $KJ$ , 解的形式为  $U = \{H_{K(j-1)+k} | j \in J, k \in K\}$ . 本文采用“两两交换”的方式构造邻域解, 通过将主问题每次迭代后产生的解交换工序  $i$  和  $j$  所需某同一类别  $k$  资源的序号, 即交换  $H_{K(i-1)+k}$  与  $H_{K(j-1)+k}$  的值来构造相邻解.

### 2.5.2 禁忌表

本文的禁忌长度设为  $\text{Random}(|10|) + \beta|J|$ , 通过预先的算例测试, 参数  $\beta$  设置为 0.5. 为使搜索更加多样化, 令  $\text{Random}(|N|)$  为  $0 \sim (N-1)$  间随机生成的一个整数, 本文  $N = 10$ . 在每次迭代后, 将禁忌表中禁忌长度减 1, 直至禁忌长度为 0 时解禁. 若当前最好解优于之前所有解, 则将这个解暂定为最好解  $C_{TS}$ , 并加入禁忌表中. 若在  $\psi$  (禁忌搜索深度, 设为 50) 次连续迭代期间未更新最好解, 则禁忌搜索停止.

## 3 数值实验

为验证所提 BDTS 算法的性能, 对随机生成算例进行测试. 值得注意的是, 本文研究的是一类新的多资源工序的资源分配优化问题, 因此文献中没有针对该问题设计的近似或精确算法可供对比. 为了进一步评估所提出算法的有效性, 将 BDTS 算法与 CPLEX 的 LP 求解器 (MILP 法)、CPLEX 的 Benders 分解求解器 (CBS 方法)、TS 算法、变邻域搜索算法 (variable neighborhood search algorithm, VNS)<sup>[24]</sup> 和在 Benders 分解算法中嵌入遗传算法 (genetic algorithm, GA) 的混合算法 (Benders decomposition embedded with genetic algorithm, BDGA) 进行对比. 本文算法采用 C 语言进行编码, 并在 CPU 2.60 GHz、内存为 16 GB 的台式机上对 30 组且每组有 10 个相同规模的仿真算例进行测试, CPU 的最大运算时间为 18 000 s. 在测试中, TS 算法的参数设置与 BDTS 算法里禁忌搜索算法参数一致. BDGA 算法中种群规模设为 100, 交叉概率为 0.8, 变异概率为 0.1; VNS 算法设置 2-opt, or-opt 和 swap 三个常用的局部搜索算子<sup>[25]</sup>, 种群规模设为 120.

对于每个仿真算例, 先通过 MILP 方法在所设置的最大运算时间内求解, 获取其运算时间, 再以此运算时间作为其他算法的计算时间上限. 因此, 对于任意仿真算例, 所有算法 (方法) 所消耗的计算时间相同. 由于算法的随机性, 每个仿真算例重复运行 3 次, 并取其平均值, 本文其余的计算结果也均为平均值. 由于目前没有与本文相关的标准数据集, 本文以 MILP 方法的运算结果为基准, 根据是否运行到最大

运算时间将仿真算例分为小规模和大规模,并分别与其他算法(方法)的运算结果进行比较,从而验证BDTS算法的寻优能力.  $UP(LB)_{Gap}$  的定义如下:

$$UP(LB)_{Gap}(\%) = \frac{UP(LB)_{approach} - UP(LB)_{MILP}}{UP(LB)_{MILP}} \times 100. \quad (31)$$

其中:  $UP(LB)_{MILP}$  和  $UP(LB)_{approach}$  分别是MILP方法和其余方法所求的上(下)界.  $\alpha$ 、 $J$  和  $T$  分别代表资源比例因子、工序数量和所有算法的运行时间. 每个例子中的有条件约束和无条件约束是随机生成的无回路有向图,且资源类别为3. 各类资源数量在所有资源总数为  $\alpha\xi/\gamma_{n+1}$  的基础上随机生成,资源比例因子  $\alpha$  分别取 80 %、100 % 和 120 %;  $\xi$  为所有工序总的资源需求量.

如表 1 所示:在小规模仿真算例中各方法的运算时间都随着工序量的增大而增大,且在大多数情况下未达到所设置的最大运行时间. 以MILP方法的运算时间为基准,MILP方法的表现优于其余方法. 相反,CBS方法表现的性能低于其余方法,与MILP方

法上界和下界的差距分别为 16.30 % 和 37.36 %. 此外,BDTS算法表现仅次于MILP方法,BDTS算法在工序量为 10 规模时均求得最优解,虽然在下界不具优势,但上界均接近MILP方法的上界,与MILP方法上界的差距为 0.86 %. 当  $\alpha = 1.2$ , 工序数量为 50 时,运算时间达到所设置的最大时间,BDTS算法表现优于MILP方法. TS算法和VNS算法与MILP方法上界差距分别为 6.05 % 和 3.83 %. BDGA算法在工序数为 10 时均求得最优解;当工序量大于 10 时,BDGA算法表现略差于BDTS算法,且与MILP方法上界和下界的差距分别为 2.15 % 和 35.44 %. BDTS算法和BDGA算法性能低于MILP方法的主要原因是Benders算法框架每迭代一次,都需要计算一次主问题模型、子问题模型和改进策略的算法,需要一定耗时. 在小规模仿真算例中由于时间有限,导致迭代次数受限,进而影响算法的收敛速度. 而BDGA算法性能在大部分仿真算例测试中低于BDTS算法的性能可能是因为BDGA算法中GA算法性能不如BDTS算法中设计的TS算法,使得BDGA算法收敛效果不佳.

表 1 小规模随机仿真算例计算结果对比

| $\alpha$ | $\tau$ | $T/s$  | BDTS 算法       |               | CBS 方法        |               | TS 算法         |               | VNS 算法        |               | BDGA 算法       |               |
|----------|--------|--------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|          |        |        | $UP_{Gap}/\%$ | $LB_{Gap}/\%$ | $UP_{Gap}/\%$ | $LB_{Gap}/\%$ | $UP_{Gap}/\%$ | $LB_{Gap}/\%$ | $UP_{Gap}/\%$ | $LB_{Gap}/\%$ | $UP_{Gap}/\%$ | $LB_{Gap}/\%$ |
| 0.8      | 10     | 12     | 0.00          | 0.00          | 2.09          | -20.04        | 1.24          | -             | 1.73          | -             | 0.00          | 0.00          |
|          | 20     | 26     | 0.20          | -35.81        | 13.45         | -41.08        | 4.78          | -             | 5.64          | -             | 1.06          | -40.11        |
|          | 30     | 287    | 1.76          | -44.30        | 16.09         | -60.06        | 7.49          | -             | 3.75          | -             | 2.85          | -49.95        |
|          | 40     | 14 201 | 2.52          | -45.61        | 21.96         | -59.53        | 9.07          | -             | 5.43          | -             | 3.07          | -50.19        |
|          | 50     | 18 000 | 0.46          | -26.56        | 30.81         | -51.50        | 9.16          | -             | 4.01          | -             | 4.22          | -47.42        |
| 1.0      | 10     | 10     | 0.00          | 0.00          | 0.00          | -0.01         | 1.18          | -             | 2.04          | -             | 0.00          | 0.00          |
|          | 20     | 19     | 1.60          | -36.92        | 1.24          | -18.20        | 5.09          | -             | 6.83          | -             | 1.15          | -41.45        |
|          | 30     | 2 141  | 1.64          | -43.53        | 9.46          | -51.57        | 7.56          | -             | 4.36          | -             | 2.18          | -55.40        |
|          | 40     | 13 923 | 1.88          | -40.67        | 24.15         | -56.75        | 8.93          | -             | 5.28          | -             | 3.64          | -56.36        |
|          | 50     | 16 905 | -0.19         | -20.82        | 37.39         | -41.76        | 10.98         | -             | 3.92          | -             | 3.48          | -38.70        |
| 1.2      | 10     | 17     | 0.00          | 0.00          | 0.00          | -0.03         | 0.00          | -             | 2.19          | -             | 0.00          | 0.00          |
|          | 20     | 39     | 0.66          | -33.51        | 7.07          | -39.01        | 3.94          | -             | 4.82          | -             | 1.72          | -37.66        |
|          | 30     | 5 519  | 1.70          | -41.30        | 15.01         | -46.96        | 6.25          | -             | 3.37          | -             | 2.23          | -43.70        |
|          | 40     | 14 838 | 1.25          | -32.87        | 27.20         | -42.02        | 7.02          | -             | 2.21          | -             | 3.38          | -41.52        |
|          | 50     | 18 000 | -0.53         | -16.19        | 38.59         | -31.85        | 8.08          | -             | 1.94          | -             | 3.30          | -29.18        |
| average  |        | 6 927  | 0.86          | -27.87        | 16.30         | -37.36        | 6.05          | -             | 3.83          | -             | 2.15          | -35.44        |

如表 2 所示:在大规模随机仿真算例结果对比中,各方法的运行时间都达到所设置的最大运行时间. 其中,CBS方法的求解能力进一步降低,与MILP方法上界和下界的差距分别为 70.39 % 和 9.11 %. 相反,BDTS算法在不同资源比例的算例中的求解效果好于其余方法. 其中,BDTS算法的上界均好于MILP方法的上界,且随着工序规模的增加,与MILP方法上界的差距进一步增大,上界获得优于MILP方法 4.74 % 的高质量解. 此外,BDTS算法的下界在绝大多数情况下优于MILP方法,其优势也随着工序规模

的增大而增大,最大提升了 24.58 % 的下界,平均提升 9.62 %. 这主要是由于CPLEX随着工序量的增大,求解能力不断弱化,而BDTS算法则具有较好的搜索能力. BDGA算法性能虽弱于BDTS算法,但相比MILP方法,上界和下界的优势分别为 0.99 % 和 0.76 %. TS算法和VNS算法上界随着工序量的增大与MILP方法的上界差距逐渐缩小,TS算法获得与MILP方法上界差距为 2.79 % 的解,VNS算法获得优于MILP方法上界 0.76 % 的解. 以上结果充分验证了本文提出的BDTS算法在大规模问题中求解的稳定性和适应性.

表 2 小规模随机仿真算例计算结果对比

| $\alpha$ | $\tau$ | $T/s$  | BDTS 算法               |                       | CBS 方法                |                       | TS 算法                 |                       | VNS 算法                |                       | BDGA 算法               |                       |
|----------|--------|--------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|          |        |        | UP <sub>Gap</sub> / % | LB <sub>Gap</sub> / % | UP <sub>Gap</sub> / % | LB <sub>Gap</sub> / % | UP <sub>Gap</sub> / % | LB <sub>Gap</sub> / % | UP <sub>Gap</sub> / % | LB <sub>Gap</sub> / % | UP <sub>Gap</sub> / % | LB <sub>Gap</sub> / % |
| 0.8      | 60     | 18 000 | -2.75                 | -3.08                 | 41.17                 | -32.15                | 8.64                  | —                     | 4.23                  | —                     | 2.38                  | -18.63                |
|          | 70     | 18 000 | -4.12                 | 2.88                  | 55.82                 | -19.86                | 5.61                  | —                     | 2.39                  | —                     | 0.99                  | -10.02                |
|          | 80     | 18 000 | -6.63                 | 18.27                 | 64.91                 | -5.79                 | 2.29                  | —                     | -0.88                 | —                     | -1.89                 | 6.20                  |
|          | 90     | 18 000 | -6.17                 | 19.42                 | 78.00                 | -4.44                 | -1.33                 | —                     | -2.31                 | —                     | -2.02                 | 11.17                 |
|          | 100    | 18 000 | -6.58                 | 20.31                 | 84.41                 | -2.08                 | -2.19                 | —                     | -1.84                 | —                     | -2.93                 | 11.65                 |
| 1.0      | 60     | 18 000 | -1.33                 | -3.88                 | 44.02                 | -31.53                | 9.12                  | —                     | 1.57                  | —                     | 2.00                  | -16.69                |
|          | 70     | 18 000 | -3.14                 | 8.62                  | 60.69                 | -8.42                 | 6.44                  | —                     | 0.35                  | —                     | -0.80                 | -0.89                 |
|          | 80     | 18 000 | -4.98                 | 14.74                 | 68.78                 | -7.79                 | 3.27                  | —                     | -1.39                 | —                     | -0.09                 | 0.02                  |
|          | 90     | 18 000 | -6.21                 | 12.90                 | 87.76                 | -1.52                 | -1.60                 | —                     | -3.01                 | —                     | -1.69                 | 7.56                  |
|          | 100    | 18 000 | -5.06                 | 24.58                 | 77.41                 | -4.81                 | -1.36                 | —                     | -3.27                 | —                     | -2.47                 | 12.63                 |
| 1.2      | 60     | 18 000 | -0.71                 | 2.48                  | 55.43                 | -11.66                | 7.59                  | —                     | 1.26                  | —                     | 2.14                  | -5.83                 |
|          | 70     | 18 000 | -2.96                 | 4.14                  | 70.69                 | -3.81                 | 5.25                  | —                     | 0.39                  | —                     | -0.23                 | -0.27                 |
|          | 80     | 18 000 | -6.52                 | 5.56                  | 78.36                 | -1.69                 | 2.51                  | —                     | -1.68                 | —                     | -1.89                 | 3.49                  |
|          | 90     | 18 000 | -6.81                 | 6.84                  | 97.52                 | -0.84                 | -2.21                 | —                     | -3.02                 | —                     | -3.02                 | 4.45                  |
|          | 100    | 18 000 | -7.11                 | 10.53                 | 90.82                 | -0.23                 | -0.13                 | —                     | -4.26                 | —                     | -5.36                 | 6.50                  |
| average  |        | 18 000 | -4.74                 | 9.62                  | 70.39                 | -9.11                 | 2.79                  | —                     | -0.76                 | —                     | -0.99                 | 0.76                  |

4 结 论

本文针对现实中广泛存在的多资源工序且同类资源有多个的情况,结合现实中资源使用规律,提炼出一类新的多资源工序的资源分配问题.在此基础上,构建了以最小化加权完工时间为优化目标的混合整数线性规划模型,并开发了一类Benders分解混合算法.该算法基于Benders分解框架,通过迭代求解主问题和对偶子问题,分别获得原问题的下界和上界,并嵌入禁忌搜索算法进行局部搜索来提高上下界逼近最优解的收敛速度.随机仿真算例结果表明:BDTS混合算法在小规模问题中能获得距离商业求解器CPLEX最优解平均差距为0.86%的满意解;BDTS混合算法在大规模问题中性能表现优于CPLEX、禁忌搜索算法和Benders分解嵌入遗传算法的混合方法,能提供更为合理的资源分配方案;小规模问题可采用本文建立的数学模型调用CPLEX直接求解.本文研究的模型和算法丰富了资源分配研究领域的理论和方法,可在生产实际中为管理者提供技术指导.进一步的工作将在此基础上,考虑更为全面的现实约束,例如负载均衡、机器周期性维护等,并根据相应的问题特性开发出具有较好性能的算法.

参考文献(References)

[1] Naderi B, Roshanaei V. Critical-path-search logic-based Benders decomposition approaches for flexible job shop scheduling[J]. *Inform Journal on Optimization*, 2022, 4(1): 1-28.

[2] Dauzère-Pérès S, Roux W, Lasserre J B. Multi-resource shop scheduling with resource flexibility[J]. *European Journal of Operational Research*, 1998, 107(2): 289-305.

[3] Mazzola J B, Neebe A W. Resource-constrained assignment scheduling[J]. *Operations Research*, 1986, 34(4): 560-572.

[4] 轩华, 李文婷, 李冰. 混合离散人工蜂群算法求解含不相关并行机的分布式柔性流水线调度[J]. *控制与决策*, 2023, 38(3): 779-789.

(Xuan H, Li W T, Li B. Hybrid discrete artificial bee colony algorithm for distributed flexible flowline scheduling with unrelated parallel machines[J]. *Control and Decision*, 2023, 38(3): 779-789.)

[5] 王汝言, 梁颖杰, 崔亚平. 车辆网络多平台卸载智能资源分配算法[J]. *电子与信息学报*, 2020, 42(1): 263-270.

(Wang R Y, Liang Y J, Cui Y P. Intelligent resource allocation algorithm for multi-platform offloading in vehicular networks[J]. *Journal of Electronics & Information Technology*, 2020, 42(1): 263-270.)

[6] 马英, 储诚斌, 杨善林. 带不可用时间段的部分可续型单机加权完工时间和调度[J]. *系统工程理论与实践*, 2009, 29(2): 128-134.

(Ma Y, Chu C B, Yang S L. Minimizing total weighted completion time in semiresumable case of single-machine scheduling with an availability constraint[J]. *Systems Engineering — Theory & Practice*, 2009, 29(2): 128-134.)

[7] Zhou Z Y, Liu P J, Feng J H, et al. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(4): 3113-3125.

[8] 熊福力, 张杏, 曹劲松, 等. 预制构件并行生产线资源配置与生产调度集成优化[J]. *控制与决策*, 2022, 37(9): 2399-2406.

(Xiong F L, Zhang X, Cao J S, et al. Integrated resource

- allocation and production scheduling for parallel concrete precast production lines[J]. *Control and Decision*, 2022, 37(9): 2399-2406.)
- [9] 杨媛, 钱斌, 胡蓉, 等. 精确动态规划算法求解绿色单机调度问题[J]. *控制与决策*, 2021, 36(8): 1891-1900.  
(Yang A, Qian B, Hu R, et al. Exact dynamic programming algorithm for green single machine scheduling problem[J]. *Control and Decision*, 2021, 36(8): 1891-1900.)
- [10] 李佳磊, 顾幸生. 双种群混合遗传算法求解具有预防性维护的分布式柔性作业车间调度问题[J]. *控制与决策*, 2023, 38(2): 475-482.  
(Li J L, Gu X S. Two-population hybrid genetic algorithm for distributed flexible job-shop scheduling problem with preventive maintenance[J]. *Control and Decision*, 2023, 38(2): 475-482.)
- [11] Desprez C, Chu F, Chu C. Minimising the weighted number of tardy jobs in a hybrid flow shop with genetic algorithm[J]. *International Journal of Computer Integrated Manufacturing*, 2009, 22(8): 745-757.
- [12] Chu F, Liu M, Liu X, et al. Reentrant flow shop scheduling considering multiresource qualification matching[J]. *Scientific Programming*, 2018, 2018: 1-8.
- [13] Huh W T, Liu N, Truong V A. Multiresource allocation scheduling in dynamic environments[J]. *Manufacturing & Service Operations Management*, 2013, 15(2): 280-291.
- [14] 关叶青, 朱颖, 谢乃明. 考虑多成本约束的柔性作业车间制造资源动态分配模型[J]. *控制与决策*, 2018, 33(11): 2037-2044.  
(Guan Y Q, Zhu Y, Xie N M. Dynamic allocation model of manufacturing resources in flexible job shop considering multi-cost constraints[J]. *Control and Decision*, 2018, 33(11): 2037-2044.)
- [15] Xie J, Zhuang W F, Ang M, et al. Analytics for hospital resource planning — Two cases studies[J]. *Production and Operations Management*, 2021, 30(6): 1863-1885.
- [16] 桓自强, 倪宏, 胡琳琳, 等. AAFSA-RA: 一种采用高级人工鱼群算法的多资源分配方法[J]. *西安交通大学学报*, 2014, 48(10): 120-125.  
(Huan Z Q, Ni H, Hu L L, et al. AAFSA-RA: A multi-resource allocation method based on an advanced artificial fish swarm algorithm[J]. *Journal of Xi'an Jiaotong University*, 2014, 48(10): 120-125.)
- [17] 王悦宸, 苏醒, 贾熹滨, 等. 应急救援中基于线性规划的多目标多资源分配模型[J]. *中国科学技术大学学报*, 2018, 48(6): 458-466.  
(Wang Y C, Su X, Jia X B, et al. A linear programming-based model for multi-object and multi-resource allocation in emergency rescue[J]. *Journal of University of Science and Technology of China*, 2018, 48(6): 458-466.)
- [18] 覃炳发, 王东, 金焱, 等. 无人机救灾中的多目标物资装配和投放点定位研究[J]. *无线互联科技*, 2019, 16(16): 103-106.  
(Qin B F, Wang D, Jin Y, et al. Research on assembly of multi-objective supplies and cargo container location of drone disaster relief[J]. *Wuxian Hulan Keji*, 2019, 16(16): 103-106.)
- [19] Nieto-Isaza S, Fontaine P, Minner S. The value of stochastic crowd resources and strategic location of mini-depots for last-mile delivery: A Benders decomposition approach[J]. *Transportation Research — Part B: Methodological*, 2022, 157: 62-79.
- [20] 李国明, 李军华. 基于混合禁忌搜索算法的随机车辆路径问题[J]. *控制与决策*, 2021, 36(9): 2161-2169.  
(Li G M, Li J H. Stochastic vehicle routing problem based on hybrid Tabu search algorithm[J]. *Control and Decision*, 2021, 36(9): 2161-2169.)
- [21] Vallikavungal Devassia J, Salazar-Aguilar M A, Boyer V. Flexible job-shop scheduling problem with resource recovery constraints[J]. *International Journal of Production Research*, 2018, 56(9): 3326-3343.
- [22] 高杰, 李承高. 根据普通紧前工序表绘制最优箭线式网络图的程序化方法[J]. *系统工程理论与实践*, 2001, 21(3): 38-43.  
(Gao J, Li C G. The programmed method of drawing an optimal AOA network diagram according to the schedule of ordinary precedence activities[J]. *Systems Engineering — Theory & Practice*, 2001, 21(3): 38-43.)
- [23] 殷允强, 王杜娟, 余玉刚. 整数规划: 基础、扩展及应用[M]. 北京: 科学出版社, 2022: 229-230.  
(Yin Y Q, Wang D J, Yu Y G. Integer programming: Foundation, extension and application[M]. Beijing: Science Press, 2022: 229-230.)
- [24] Liang Y C, Chuang C Y. Variable neighborhood search for multi-objective resource allocation problems[J]. *Robotics and Computer-Integrated Manufacturing*, 2013, 29(3): 73-78.
- [25] 王征, 张俊, 王旭坪. 多车场带时间窗车辆路径问题的变邻域搜索算法[J]. *中国管理科学*, 2011, 19(2): 99-109.  
(Wang Z, Zhang J, Wang X P. A modified variable neighborhood search algorithm for the multi depot vehicle routing problem with time windows[J]. *Chinese Journal of Management Science*, 2011, 19(2): 99-109.)

## 作者简介

翁武燕(1994—), 男, 博士生, 从事运筹与优化管理、资源分配等研究, E-mail: wengwuyanfu@136.com;

储诚斌(1965—), 男, 教授, 博士生导师, 从事生产调度、物流与供应链管理等研究, E-mail: chengbin.chu@gmail.com;

吴鹏(1987—), 男, 教授, 博士生导师, 从事运筹与优化管理、智能交通管理等研究, E-mail: wupeng88857@126.com.