



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



## 基于模拟退火算法的人机协同装配线平衡问题研究

毛照昉, 王威, 方侃, 黄典

引用本文:

毛照,王威,方侃,黄典. 基于模拟退火算法的人机协同装配线平衡问题研究[J]. *控制与决策*, 2024, 39(10): 3366–3374.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2023.1068>

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 基于接触状态感知发育的机器人柔性装配方法

Flexible assembly method based on contact state perception development

控制与决策. 2021, 36(4): 876–884 <https://doi.org/10.13195/j.kzyjc.2019.1079>

#### 考虑批量分割的单元装配系统工作量平衡联合决策模型

Joint decision model of Seru production system balancing considering lot-splitting

控制与决策. 2021, 36(10): 2517–2527 <https://doi.org/10.13195/j.kzyjc.2020.0167>

#### 带峰值能耗约束流水线调度的协同群智能优化

Cooperative memetic optimization for flowshop scheduling with peak power consumption constraint

控制与决策. 2021, 36(10): 2350–2358 <https://doi.org/10.13195/j.kzyjc.2020.0429>

#### 区间数可重入混合流水车间调度与预维护协同优化

Collaborative optimization of interval number reentrant hybrid flow shop scheduling and preventive maintenance

控制与决策. 2021, 36(11): 2599–2608 <https://doi.org/10.13195/j.kzyjc.2020.0973>

#### 离散蝙蝠算法在三阶段装配流水线调度问题的应用

Discrete bat algorithm in three-stage assembly flowshop scheduling problem

控制与决策. 2021, 36(9): 2267–2278 <https://doi.org/10.13195/j.kzyjc.2020.0054>

# 基于模拟退火算法的人机协同装配线平衡问题研究

毛照昉<sup>1</sup>, 王威<sup>1</sup>, 方侃<sup>1</sup>, 黄典<sup>1,2†</sup>

(1. 天津大学 管理与经济学部, 天津 300072; 2. 天津大学 新媒体与传播学院, 天津 300072)

**摘要:** 作为一种新兴的智能制造趋势, 将协作机器人引入装配线, 与工人以人机协同方式进行装配, 正受到越来越多的重视和实践. 人机协同装配线的特点在于工人和机器人可以在同一工位中独立执行、并行执行或协作执行装配任务, 从而提高传统装配线的生产效率. 以最小化节拍时间为优化目标, 对人机协同装配线平衡问题开展研究. 首先, 设计节拍时间下限、上限和初始解的加强策略, 并基于已有的人机协同装配线平衡问题模型, 构建一个新的增强的混合整数规划模型; 其次, 设计一类改进的模拟退火算法, 使用多种工位完工时间评估方法, 实现对问题的高效求解; 此外, 通过大量的计算实验, 验证所提模型和算法的有效性和适用性; 最后, 对相关参数开展敏感性分析, 为制造企业引入协作机器人开展人机协同装配提供管理启示和实践参考.

**关键词:** 协作机器人; 人机协同; 装配线平衡问题; 加强策略; 增强的混合整数规划模型; 模拟退火算法

中图分类号: TH186

文献标志码: A

DOI: 10.13195/j.kzyjc.2023.1068

引用格式: 毛照昉, 王威, 方侃, 等. 基于模拟退火算法的人机协同装配线平衡问题研究 [J]. 控制与决策, 2024, 39(10): 3366-3374.

## Research on the human-robot collaborative assembly line balancing problem based on simulated annealing algorithm

MAO Zhao-fang<sup>1</sup>, WANG Wei<sup>1</sup>, FANG Kan<sup>1</sup>, HUANG Dian<sup>1,2†</sup>

(1. College of Management and Economics, Tianjin University, Tianjin 300072, China; 2. School of New Media and Communication, Tianjin University, Tianjin 300072, China)

**Abstract:** As an emerging trend of intelligent manufacturing, collaborative robots are recently introduced and implemented in the assembly line systems with human-robot collaborative approach, which has received increasing attention and practice. In such system, workers and robots can perform tasks separately, simultaneously, or collaboratively at each station, which can significantly improve the efficiency of the production line. This study addresses an assembly line balancing problem with human-robot collaboration (ALBP-HRC) to minimize the cycle time. We first propose multiple enhancement strategies, including lower bound, upper bound and initial solution. Meanwhile, based on the existing ALBP-HRC model, we formulate an enhanced mixed integer programming model. Secondly, we present an improved simulated annealing algorithm, using multiple completion time estimation procedures for stations to solve the problem efficiently. Moreover, extensive experiments are conducted, and the results verify the effectiveness and applicability of the methods. Finally, we investigate sensitivity analysis on different parameters, revealing the managerial insights and practical references for enterprises implementing human-robot collaborative assembly line.

**Keywords:** collaborative robots; human-robot collaboration; assembly line balancing; enhancement strategies; enhanced mixed integer programming; simulated annealing algorithm

## 0 引言

在现代制造企业中, 机器人被广泛应用于生产装配领域, 发挥着越来越重要的作用. 其中, 工业机器人被大量应用于装配线, 助力企业以低成本开展批量生

产<sup>[1]</sup>.

随着中国人口老龄化和人力成本的增加, 制造业正受到劳动力不足和装配成本增加等方面的压力<sup>[2]</sup>. 为此, 制造企业逐渐尝试采用全自动机器人来

收稿日期: 2023-07-30; 录用日期: 2023-11-22.

基金项目: 国家自然科学基金项目 (92167206, 72361137004, 72231005); 科技部创新方法专项项目 (2020IM030300); 天津市哲学社会科学规划项目 (TJGL21-016).

†通讯作者. E-mail: huangdian@tju.edu.cn.

\*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

代替人工。在装配过程中,全自动机器人能够以高速度和高精度执行重复的任务,可以有效提高产品的组装质量和速度。同时,全自动机器人的应用也存在一定局限。例如,现有机器人的灵活性有限,全自动机器人初期投资费用高且回报期长。在这种背景下,人机协同的生产方式受到越来越多的关注<sup>[3]</sup>,并广泛应用于工业制造中。一个典型的协作任务是拧螺丝,其中工人负责将螺丝钉放在工件上,协作机器人负责将螺丝钉拧入,工人和协作机器人共享相同的工作空间和时间。

人机协同装配线集合了人工装配线和自动化装配线的双重优势,即人对复杂、不确定任务的灵活特性,以及机器人对重复、高负荷任务的容忍度和精准度。得益于人机协同装配的优势,目前已经有许多制造企业将协作机器人引入装配线并实施协同装配模式<sup>[1]</sup>。其中,如何合理地将机器人分配到工位上,并决定每个任务的加工顺序和加工方式,从而实现装配线平衡,是企业实现人机协同高效生产的重要因素和关注点。

自从Salveson<sup>[4]</sup>提出装配线平衡问题(assembly line balancing problem, ALBP)的第1个数学模型以来,学者们针对ALBP领域进行了大量的研究,并在ALBP问题的模型建立和优化求解方面取得了丰富的研究成果<sup>[5-6]</sup>。由于ALBP问题的内在复杂性,往往无法利用现有的优化求解器进行有效求解。为此,Uğurdağ等<sup>[7]</sup>和Peeters等<sup>[8]</sup>学者针对ALBP的不同变型问题设计了更有效的模型和算法进行求解。随着机器人在生产装配领域的广泛应用,人机协同装配线平衡问题(assembly line balancing problem with human-robot collaboration, ALBP-HRC)逐渐受到学者们的关注<sup>[9-11]</sup>。Weckenborg等<sup>[12]</sup>提出了一类考虑人机协同加工方式的混合整数规划模型,并设计了一种改进的遗传算法进行求解。类似地,Dalle等<sup>[13]</sup>设计了一类遗传算法,以最小化工人数量、设备数量和工人负荷。Raatz等<sup>[14]</sup>也讨论了人机协同装配环境中的任务调度问题,并通过遗传算法来优化节拍时间。Kinast等<sup>[3]</sup>将人机协同装配线的生产成本和节拍时间归一化,建立了约束规划模型,并提出了一种带有偏置随机密钥编码和可变邻域搜索的混合遗传算法。Mao等<sup>[15]</sup>将人机协同装配扩展到了U型装配线中,并提出一套启发式的求解方法。总体来看,作为新兴的研究领域,由于ALBP-HRC问题的复杂性,其模型求解存在运算时间长、结果误差大等缺点,往往无法利用现有的优化求解器进行有效求解,当前对

ALBP-HRC问题所涉及到的求解算法多以遗传算法(genetic algorithm, GA)等元启发式算法为主。虽然上述所提到的算法在解决ALBP-HRC问题时比优化求解器和传统的启发式算法效果更好,但是为了保证解的质量,也存在收敛速度慢、运算时间开销大等缺陷,在计算时间方面仍有很大的改进空间。经过研究比较,发现给ALBP-HRC模型提供节拍时间上下限和初始解等加强策略能够优化模型求解过程。此外,模拟退火算法能够满足所构建ALBP-HRC模型的求解要求,并具有较好的求解性能。

本文以最小化节拍时间为优化目标,开展ALBP-HRC问题研究,主要贡献有以下几方面:1)针对该问题,提出更紧的节拍时间上下限计算方法,建立了增强的混合整数规划模型(enhanced mixed integer programming, EMIP),并为模型求解过程提供了一个高质量的初始解;2)提出一种改进的模拟退火算法(improved simulated annealing, ISA)对该问题求解,并设计多种邻域搜索策略以及两种工位完工时间评估方法;3)生成不同规模的算例,在MIP模型、EMIP模型、ISA算法和GA算法之间开展对比实验,实验结果验证了EMIP模型和ISA算法的有效性;4)根据不同算例的计算结果,进一步分析机器人密度等参数对装配线效率的影响,为开展人机协同装配工作实践提供管理启示。

## 1 问题描述及模型建立

### 1.1 问题分析

如图1(a)所示,传统人工装配线由一组工位组成,工位之间一般由传送带等物料运输系统连接,每个工位都分配有一个操作工人,具有确定优先关系和加工时间的任务被依次处理,直到所有任务都被加工完成。传统装配线上的任务只能由人工处理,因此,传统装配线平衡问题只需要在满足任务优先关系的前提下,将任务合理地分配到工位上,以实现相应目标的优化。

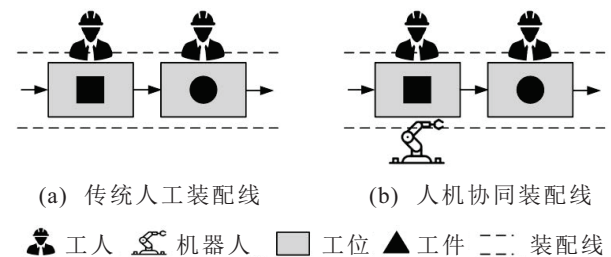


图1 传统人工装配线及人机协同装配线工作示意图

图1(b)展示了人机协同装配线工作示意,人机协同装配线平衡问题与传统的装配线平衡问题之间区

别体现在: 1) 对于人机协同装配线, 机器人数量通常少于工位数量, 需额外考虑如何将有限数量的机器人合理地分配到相应工位上. 2) 传统人工装配线只能由工人加工, 如图2(a), 对于人机协同装配线而言, 有工人加工、机器人加工和人机协同加工3种加工方式, 且相应的加工时间不同. 由于与工人共享物理操作空间, 出于安全考虑, 当协作机器人在工人周围单独作业时, 其加工速度会有一定降低, 此时协作机器人单独加工的效率低于工人加工, 所需加工时间长于工人加工<sup>[12,15-16]</sup>, 而人机协同的加工效率则高于工人加工, 其加工时间短于工人加工. 因此, ALBP-HRC问题需额外确定每个任务的加工方式. 3) 在人机协同装配线中, 对于分配有机器人的工位(以下简称为机器人工位), 工人与机器人可以独立或协作执行任务. 当工人和机器人分别独立加工时, 此时的装配线运行模式为图2(b)中的任务2和任务3; 当工人与机器人协作加工时, 此时装配线运行如图2(b)中的任务1和任务4, 人机协同装配线运行模式更为多样.

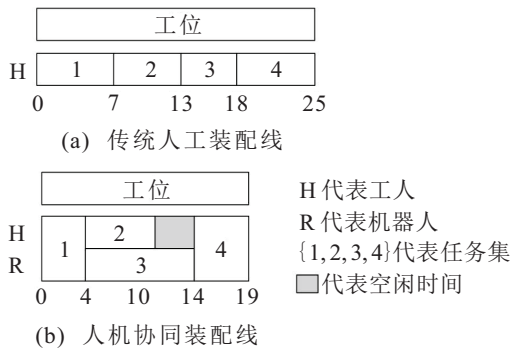


图2 人机协同装配线运行模式

1.2 条件假设

本文对所研究的ALBP-HRC问题作如下假设:

- 1) 人机协同装配线生产单品种的产品, 不考虑产品的切换时间;
- 2) 任务之间的优先关系是已知的;
- 3) 每个工位可被分配至多一个机器人;
- 4) 每个任务只能分配给一个工位, 并且只能采用一种加工方式;
- 5) 工人的操作熟练度相同, 能加工所有任务;
- 6) 机器人的灵活度有限, 不能保证每项任务都可以采用机器人或者人机协作的方式;
- 7) 在不同的加工方式下, 任务的操作时间都是确定的、已知的.

1.3 模型构建

为了对该问题进行详细描述, 现对相关数学符号进行描述, 如表1所示.

表1 数学符号

下标和集合	含义
$I$	任务集合, $I = \{1, 2, \dots, n\}$
$K$	工位集合, $K = \{1, 2, \dots, m\}$
$P$	加工方式集合, $P = \{p_h, p_r, p_c\}$ . 其中: $p_h$ 表示工人加工, $p_r$ 表示机器人加工, $p_c$ 表示人机协同加工
$i, j$	任务编号, $i, j \in I$
$k$	工位编号, $k \in K$
$p$	加工方式, $p \in P$
$E$	具有直接优先关系任务对 $(i, j)$ 的集合, $i, j \in I$

模型参数	含义
$t_{ip}$	任务 $i$ 采用加工方式 $p$ 的加工时间
$c_{\max}$	节拍时间上限, $c_{\max} = \max\{t_{\max}, 2 \cdot \lfloor t_{\text{sum}}/m \rfloor\}$ , $t_{\max} = \max\{t_{ip}   i \in I, p \in P\}$ , $t_{\text{sum}} = \sum_{i \in I} \max\{t_{ip}   p \in P\}$
$q$	机器人的数量

Weckenborg 等<sup>[12]</sup> 针对 ALBP-HRC 问题提出了一个混合整数规划 (mixed integer programming, MIP) 模型, 以下简称为 MIP 模型. 在所提出的 MIP 模型中, 针对节拍时间只给了上限计算公式, 即  $c_{\max} = \max\{t_{\max}, 2 \cdot \lfloor t_{\text{sum}}/m \rfloor\}$ ,  $t_{\max} = \max\{t_{ip} | i \in I, p \in P\}$ ,  $t_{\text{sum}} = \sum_{i \in I} \max\{t_{ip} | p \in P\}$ . 为进一步加快模型求解, 本文提出一个更紧的节拍时间上下限的计算方法, 得到新的节拍时间上限  $c_{\text{upper}}$  和下限  $c_{\text{lower}}$ , 并建立加强的混合整数规划模型, 以下简称为 EMIP 模型.

1.3.1 计算节拍时间下限

本文基于国际标准化组织和学者们的研究统计<sup>[12,17-18]</sup>, 对不同加工方式下的加工时间定义如下: 通常情况下, 机器人单独加工的效率低于工人单独加工的效率, 设定机器人的加工时间是人工的2倍, 即  $t_{ip_r} = 2 \cdot t_{ip_h}$ . 在人机协同加工方式下, 由于机器人对工人的加工过程给与了支持, 人机协同的加工速度会比人工快, 设定人机协同加工时间比人工减少了30%, 则人机协同加工时间  $t_{ip_c} = 0.7 \cdot t_{ip_h}$ .

为了求解节拍时间的下限, 假设机器人具有足够高的灵活性, 每项任务都有3种加工方式可以选择. 在人机协同装配线中, 工人和机器人并行执行任务是最有效的加工方式, 计算及证明过程如下: 1) 若任务采用人机并行加工方式, 且机器人和工人同时完工, 则总完工时间为  $A$ , 如图3(a)所示; 2) 若任务都由工人加工, 对于同一项任务而言, 工人的加工时间是机器人加工时间的一半, 则总完工时间为  $1.5A$  (如图3(b)所示), 且总完工时间等于所有任务由工人独自加工的时间之和, 即  $1.5A = \sum_{i \in I} t_{ip_h}$ ; 3) 若任务都采用人机协同加工方式, 由于  $t_{ip_c} = 0.7 \cdot t_{ip_h}$ , 总完工时间为  $0.7 \sum_{i \in I} t_{ip_h}$ , 则有  $0.7 \sum_{i \in I} t_{ip_h} = 0.7 \times 1.5A = 1.05A$ ,

如图 3(c) 所示.

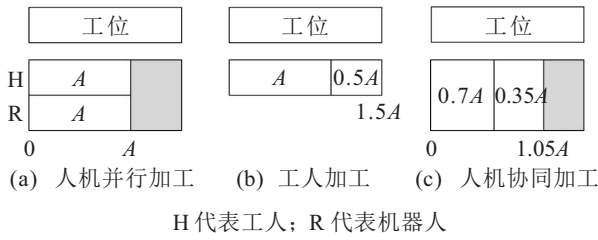


图 3 不同加工方式的完工时间

综上, 在理想情况下, 工人和机器人并行执行任务的总完工时间最短, 总完工时间为  $A = \frac{2}{3} \sum_{i \in I} t_{ip_h}$ . 对于有  $m$  个工位的人机协同装配线, 节拍时间的下限计算公式为

$$c_{\text{lower}} = \frac{2}{3} \sum_{i=1}^n t_{ip_h} / m.$$

### 1.3.2 计算节拍时间上限和初始解

对于节拍时间的上限, 本文提出一个基于贪心策略的多阶段算法 (greedy-based multi-phase algorithm, GMA), 计算 ALBP-HRC 问题的节拍时间, 此时得到的节拍时间是原问题的上限, 算法具体如下.

1) 任务分配策略. 如图 1 所示, 通过引入机器人资源, 传统的 ALBP 问题扩展为 ALBP-HRC 问题. 因此, ALBP-HRC 的节拍时间一定不会超过原始 ALBP 的节拍时间, ALBP 问题的节拍时间是 ALBP-HRC 的上限.

本文先将 ALBP-HRC 问题松弛为 ALBP 问题, 然后利用目前已经很成熟的 SALOME 算法<sup>[19]</sup> 高效求解相应的 ALBP 问题, 此时得到了各个任务的分配情况, 也是 ALBP-HRC 问题的初始解.

2) 基于贪心规则的机器人资源调度和加工方式分配方法. 初始解完成了任务分配, 但未确定机器人的分配及任务加工方式, 为此, 本文提出基于贪心的规则来解决上述问题.

① 机器人资源调度. 在初始解中, 任务分配情况和工位的完工时间已知 (任务工人加工时间的总和). 为了最小化节拍时间, 优先将机器人安排给完工时间长的工位上, 以减少该工位的完工时间、平衡各工位的完工时间, 最终实现降低节拍时间.

② 加工方式分配. 对于机器人工位, 每个任务有 3 种加工方式, 本文通过以下策略确定任务的加工方式: 首先, 分析当前工位中各任务的前驱任务 (当前任务开始之前必须完成的任务), 计算各任务的最早可加工时间; 然后, 按照各任务最早可加工时间的升序, 计算每项任务在 3 种加工方式下的完成时间, 选择完成时间最短的加工方式. 工位中所有任务的最后完

成时间就是当前工位的完工时间.

3) 节拍时间确定. 基于上述策略, 完成了任务分配、机器人资源调度以及加工方式的确定, 并计算各工位的完工时间, 因此所有工位完工时间的最大值即为当前人机协同装配线的节拍时间, 该时间可作为问题节拍时间的一个上限  $c_{\text{upper}}$ .

### 1.3.3 加强的混合整数规划模型构建

已知 ALBP-HRC 问题的节拍时间上限  $c_{\text{upper}}$  和下限  $c_{\text{lower}}$ , 则有如下约束:

$$c_{\text{lower}} \leq c \leq c_{\text{upper}}. \quad (1)$$

MIP 模型和约束 (1) 组合, 构成了一个加强的混合整数规划模型, 下文简称为 EMIP 模型. 本文使用求解器 Gurobi 求解 EMIP 模型, 但是由于问题的内在复杂性, 求解器有时找到一个可行解的时间会很长, 或者找到的可行解质量很差, 影响求解进程. 需要指出的是, GMA 算法所求的解是 ALBP-HRC 的一个高质量可行解, 因此使用 GMA 算法求得的可行解作为初始解传入 Gurobi 求解器, 以优化 EMIP 的求解过程.

## 2 改进的模拟退火算法

由于传统 ALBP 问题已经是 NP-困难问题<sup>[12]</sup>, 更为复杂的 ALBP-HRC 问题自然也是 NP-困难问题, 难以通过简单算法进行高效求解. 为此, 本文提出一类改进的模拟退火算法.

### 2.1 算法介绍

模拟退火算法 (simulated annealing, SA) 是基于 Monte-Carlo 迭代求解策略的一种随机寻优算法, 它能够以一定概率跳出局部最优, 尤其适用于组合优化问题<sup>[20]</sup>. 但是模拟退火算法在求解较大规模的问题时, 也存在收敛速度慢等缺点, 当计算时间有限时, 难以保证计算结果为全局最优.

针对模拟退火算法的特点和局限, 本文进行如下改进: 1) 解的表示采用工位分配列表方式. 因为模拟退火算法具有两层迭代循环过程, 所以采用这种编码方式可以很大程度缩短解码过程. 2) 设计一个新的邻域搜索方法, 在搜索过程中, 同时考虑机器人和任务的分配优化. 3) 设计启发式规则估算工位完工时间. 在初始解改善的过程中, 因为机器人工位需要考虑任务加工方式的选择, 计算难度较大. 为加快求解速度, 设计一种估算法, 以满足前期优化需要. 4) 设计一个精确的工位完工时间计算方法, 以降低计算量. 在对工位完工时间进行精确计算时, 将计算对象分为人工工位和机器人工位两部分, 并针对机器人工位建立一个新的混合整数规划模型, 这样问题由多工位调

度转变为单工位调度问题,进而降低计算的难度。

本文设计的模拟退火算法步骤如下:

step 1: 生成初始可行解  $X_0$ , 对初始可行解进行工位完工时间评估  $E_0$ , 令  $X_{\text{best}} = X_0, E_{\text{best}} = E_0$ .

step 2: 设置初始温度  $T_0$ , 降温系数  $\alpha$ , 邻域搜索次数  $\text{iters}$ , 终止温度  $T_{\text{min}}$ , 退温函数为  $T_{k+1} = \alpha T_k$ .

step 3: 在当前温度  $T_k$  下, 对  $X_{\text{best}}$  进行  $\text{iters}$  次邻域搜索, 邻域搜索有两个方向: 改变机器人的工位和改变任务分配方案. 生成一个新的邻域解  $X_{\text{new}}$ , 估算节拍时间  $E_{\text{new}}$ .

step 4: 计算节拍时间差  $\Delta E, \Delta E = E_{\text{new}} - E_{\text{best}}$ . 如果  $\Delta E < 0$ , 则接受新方案,  $X_{\text{best}} = X_{\text{new}}, E_{\text{best}} = E_{\text{new}}$ ; 否则, 根据 Metropolis 准则计算状态转移概率  $p = \exp(-\Delta E/T)$ , 以概率  $p$  接受新解.

step 5: 计算当前温度下最优解对应的精确节拍时间, 并更新  $E_{\text{best}}$ , 更新温度  $T_{k+1} = \alpha T_k$ .

step 6: 返回 step 3, 进行循环, 直到满足条件  $T_k < T_{\text{min}}$ , 当前的最优解就是最终方案.

## 2.2 解的编码

为高效解决人机协同装配线平衡问题, 本文引入两个列表用于解的编码. 第1个列表  $\text{List}_1$  长度为  $n$ , 表示任务在工位上的分配情况. 列表中的每个元素都是一个介于1到  $m$  (工位的数量) 之间的整数. 因此, 第  $i$  个元素的值表示任务  $i$  分配到的工位号. 第2个列表  $\text{List}_2$  长度为  $m$ , 表示机器人在工位上的分配情况. 每个元素的取值为0或1, 表示机器人是否被分配到工位.

## 2.3 初始解生成

对于 ALBP-HRC 问题, 其主要包括任务分配、机器人分配以及加工方式确定等核心环节, 为此, 本文提出如下的初始解生成方案.

### 1) 机器人分配和任务分配方式.

本文首先将给定数量的机器人随机分配到工位上, 再根据不同类型工位设置相应的加工负荷, 尽可能将任务在工位内均匀分配. 由于协作机器人能提高装配效率, 机器人工位可以承受更高的负荷. 本文定义了一个工作量分布系数, 在给定的工作量分布系数范围内选择一个随机数, 使得有机器人工位的负荷高于人工工位.

机器人工位负荷为

$$\text{Limit}_1 = \frac{\sum_{i=1}^n t_{ip_h}}{m + q\lambda} \cdot (1 + \lambda),$$

人工工位负荷为

$$\text{Limit}_2 = \sum_{i=1}^n t_{ip_h} / (m + q\lambda).$$

在确定工位负荷后, 开始对工位分配任务. 基本思路为: 在没有前驱任务或者前驱任务已分配的待分配任务中随机选择一项任务, 分配到当前工位. 当前工位的加工时间为任务工人加工时间的总和, 如果当前工位的加工时间超过了设置的工作负荷, 则任务将被分配给下一个工位. 重复上述过程, 直到所有任务都被分配到工位为止.

### 2) 任务加工方式确定和工位完工时间估算.

在确定机器人分配和任务分配后, 本文开展任务加工方式的确定以及节拍时间估算, 以减少计算量. 对于人工工位, 其完工时间等于工位中各任务人工加工时间之和; 对于机器人工位, 本文应用一种基于优先关系的启发式方法来计算工位的完成时间. 在这个启发式过程中, 待分配任务始终选择令当前工位完工时间最短的加工方式, 计算机器人工位的任务的分配和工位完工时间. 在得到各工位的完工时间后, 计算各工位完工时间的最大值, 从而得到装配线的节拍时间.

如前文1.3节所述, 本文提出了GMA算法计算节拍时间的上限, GMA算法所求的解也是ALBP-HRC的一个初始解. 本文对比了GMA算法的解以及本节所提出的初始解在算法运行中的效果. 随机选择40个实例, 使用两种不同的初始解生成方案, 最终的节拍时间均值偏差仅为0.35%, 但是使用GMA初始解生成方案能更有效地缩短总计算时间. 因此对于ISA算法, 本文选定GMA算法生成初始解, 以更有效地促进算法的后续迭代.

## 2.4 邻域搜索策略

为了保证邻域搜索的全面性, 本文针对机器人的分配方案和任务的分配方案设计不同的邻域变换算子. 改变机器人工位分配采用的是突变算子, 即以一定的概率随机改变机器人的所在工位, 并且保证机器人数量不会增加. 对于任务分配方案的改变策略, 采用的是交换算子和插入算子. 在邻域搜索过程中, 等概率随机选择一个算子, 生成新的邻域解. 其中: 插入算子是在不违背任务优先关系的前提下, 在完工时间最长的工位中随机选择一个任务插入到其他工位中, 得到一个新的可行解; 交换算子是在不违背任务优先关系的前提下, 交换随机挑选的两个任务的工位, 同样得到一个新可行解.

## 2.5 精确的工位完工时间计算机制

如前文所述, 为了提高算法的计算速度, 在模拟退火过程的内循环中设计了工位完工时间的估算

方法. 利用启发式调度方法所得到的工位完工时间并不是一个精确值, 这里提出一种精确的计算方式, 在模拟退火过程的外循环中计算精确的工位完工时间. 在人机协同装配线中, 存在两类工位, 即人工工位和机器人工位. 对于人工工位, 完工时间等于工位上各任务工人加工时间之和. 对于机器人工位, 则需要单独计算. 为精确计算各机器人工位的完工时间, 针对各机器人工位建立了一个混合整数规划模型.

### 3 实验分析

本节通过丰富的对比实验, 验证所提出方法的有效性. 3.1 节介绍不同规模算例的生成过程; 实验对比结果在 3.2 节中进行详细分析; 3.3 节对相关重要参数进行分析研究, 为相关企业的管理决策提供参考.

#### 3.1 算例生成

目前, ALBP-HRC 问题属于较为新兴的研究问题, 尚未形成标准的算例集. 基于 Otto 等<sup>[21]</sup>提出的传统 ALBP 问题的基准数据集构造 3 种规模的算例, 分别是小型算例 (20 个任务)、中型算例 (50 个任务) 和大型算例 (100 个任务). 在 Weckenborg 等<sup>[12]</sup>的基础上, 本文将相关参数设定如下:

1) 机器人密度 (robot density, RD): 机器人数量与工位数量的比值, 本文算例的  $RD \in \{0.2, 0.4\}$ .

2) 任务密度 (task density, TD): 任务数量与工位数量的比值, 本文算例的  $TD = 4$ .

3) 机器人灵活性 (robot flexibility, RF), 协作灵活性 (Collaboration flexibility, CF): 分别定义为机器人可以单独执行或协作执行任务数占总任务数的比例, 本文算例的  $RF = CF = 0.4$ .

4) 任务灵活性 (task flexibility, TF): 描述任务优先关系的特征, 反映向工位分配任务的自由度.  $TF = 1$  表示任务之间不存在优先关系, 本文将 TF 取值范围设定为  $0.9 < TF < 0.95$ .

5) 任务加工时间  $t_{ip}$ : 如 1.3 节所述, 设定  $t_{ip_r} = 2 \cdot t_{ip_h}, t_{ip_c} = 0.7 \cdot t_{ip_h}$ .

每种规模下算例数量的计算公式为

$$\text{算例数量} = \text{优先关系种类数} \times \text{RD 的种类数} \times \text{TD 的种类数} \times \text{RF (CF) 的种类数}.$$

本文算例在给定的灵活性范围内, 小、中、大规模的算例各生成 20 种不同的优先级关系, 由于  $RD \in \{0.2, 0.4\}, RF(CF) = 0.4, TD = 1$ , 小、中、大规模的算例数量各为  $20 \times 2 \times 1 \times 1 = 40$ , 算例总数为 120.

本文在 MIP 模型、EMIP 模型、ISA 算法和 GA 算法的结果之间开展对比. 本文所有算法和实验均通过 Python 编程实现, Gurobi 求解器的版本号为 Gurobi 9.5.2, MIP 模型和 EMIP 模型求解的时间限制在 1800 s, 当程序运行时间超过时间上限后, 记录当前结果的相对误差 (Gap). 本文所有的实验分析均在一台具有英特尔酷睿 i5-8265 处理器、1.6 GHz 和 8 GB 内存的个人计算机上开展.

#### 3.2 实验结果分析

本节给出 MIP、EMIP、ISA 和 GA 的计算结果, 比较各种算法的性能, 其中 GA 改编自文献 [12]. 对比实验结果所涉及到的数据指标有: 1) MIP 和 EMIP 程序终止后剩余的相对误差 Gap; 2) 4 种方法在给定运行时间内获得可行解和最优解的数量; 3) 程序运行结束时的总计算时间 CPU (单位为 s) 及其标准差  $\sigma$ ; 4) 不同计算方法在节拍时间 CT 和总计算时间 CPU 方面的偏差  $p_x^y$ , 计算公式为

$$p_x^y = \frac{CT_y(CPU_y) - CT_x(CPU_x)}{CT_x(CPU_x)} \times 100\%,$$

其中  $x, y = \{MIP, EMIP, ISA, GA\}$ , 表示不同的计算方法.

##### 3.2.1 EMIP 模型与 MIP 模型对比

首先, 对于 120 个算例, 本文统计了 4 种方法的平均节拍时间 CT 和平均计算时长 CPU. 表 2 表明了 EMIP 模型中节拍时间上下限的有效性: 两模型的节拍时间和总计算时间的均值相近 (偏差在 2% 以内), 但是 EMIP 的平均误差 Gap 远小于 MIP, 这表明 EMIP 能够找到更紧的下限, 验证了所提出的下限计算公式的有效性.

表 2 MIP 和 EMIP 的平均实验结果

	小规模算例 ( $N = 20$ )			中规模算例 ( $N = 50$ )			大规模算例 ( $N = 100$ )		
	CT	CPU/s	Gap/%	CT	CPU/s	Gap/%	CT	CPU/s	Gap/%
MIP	1064.0	919.4	13.9	1108.6	1800.0	38.8	676.7	1800.0	31.4
EMIP	1063.7	911.9	9.5	1094.7	1800.0	27.5	675.2	1800.0	29.4
$p_{MIP}^{EMIP} / \%$	-0.03	-0.8	-31.6	-1.25	0	-29.1	-0.22	0.0	-6.4

此外,对比了在不同算例规模下,MIP和EMIP获得最优解和可行解的数量,验证了EMIP模型在求解问题时更具有优势.如表3所示,最优解、可行解和无可行解分别用 optimal、feasible 和 infeasible 表示.对于小、中型规模算例,MIP和EMIP的求解效果相同;对于大型算例,MIP找到了6个算例的可行解,而EMIP找到了20个算例的可行解.

同时,表4对MIP和EMIP的求解质量进行了详

细对比,进一步验证了EMIP模型在求解质量方面的优越性.不同规模的算例被分为3类,EMIP、Tie和MIP分别代表EMIP所求节拍时间更短、两者节拍时间相等和MIP所求节拍时间更短. $n$ 为求解方法对应的算例数量.在绝大多数算例中,EMIP找到的解比MIP的解更好或者相同.具体而言,EMIP在120个算例中总共得到了89个比MIP的当前最佳值更好或相同的解.

表3 MIP与EMIP最优解和可行解的数量

	小规模算例 ( $N = 20$ )			中规模算例 ( $N = 50$ )			大规模算例 ( $N = 100$ )		
	optimal	feasible	infeasible	optimal	feasible	infeasible	optimal	feasible	infeasible
MIP	20	20	0	0	40	0	0	6	34
EMIP	20	20	0	0	40	0	0	20	20

表4 MIP与EMIP的求解质量对比

		小规模算例 ( $N = 20$ )			中规模算例 ( $N = 50$ )			大规模算例 ( $N = 100$ )		
		EMIP	Tie	MIP	EMIP	Tie	MIP	EMIP	Tie	MIP
	$n$	4	33	3	9	25	6	18	0	2
MIP	CPU	1800.0	919.4	1800.0	1800.0	1800.0	1800.0	1800.0	1800.0	1800.0
	$\sigma$	(0)	(881.1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
EMIP	CPU	1800.0	911.9	1800.0	1800.0	1800.0	1800.0	1800.0	1800.0	1800.0
	$\sigma$	(0)	(888.5)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

3.2.2 ISA算法与EMIP模型、GA算法对比

由于EMIP的求解效果更好,本节在ISA、EMIP和GA之间展开对比.首先对比了不同算例规模下的平均实验结果,如表5所示.对于不同规模算例,ISA与EMIP的节拍时间均值偏差都在4%以内,但是ISA的平均计算时长比EMIP缩短了88%以上.对于大规模算例,ISA算法的平均节拍时间低于EMIP,这也说明了ISA算法的稳定性和有效性.相比之下,对于小、中规模算例,GA和EMIP的节拍时间均值偏差都在4%以上,且对于大规模算例,GA算法的平均节拍时间高于EMIP.在计算时长方面,GA算法始终优于EMIP,但ISA算法仍然是最快、最稳定的求解方法.

表5 EMIP、ISA和GA的平均实验结果

	小规模算例		中规模算例		大规模算例	
	CT	CPU/s	CT	CPU/s	CT	CPU/s
EMIP	1063.7	911.9	1094.7	1800.0	675.2	1800.0
ISA	1080.1	27.5	1113.5	84.3	650.5	201.0
GA	1107.0	69.4	1177.6	292.4	699.8	1371.3
$p_{EMIP}^{ISA}/\%$	1.5	-97.0	1.7	-95.3	-3.7	-88.8
$p_{EMIP}^{GA}/\%$	4.1	-92.4	7.6	-83.8	3.6	-23.8

同时,本文对EMIP和ISA算法的求解质量进行详细对比.在计算结果方面,EMIP在小、中规模算例上具有一定优势,但在大规模算例上明显不及ISA算法;在计算时间方面,ISA算法全方位优于EMIP.表6详细呈现了两种方法对不同规模算例的求解质量表现,其中ISA、Tie和EMIP分别代表ISA所求节拍时间更短、两者节拍时间相等和EMIP所求节拍时间更短.具体而言,在给定时间内,EMIP可以对一半的小规模算例(20个)求得最优解,而在中型和大型算例中均无法获得最优解.可以看出,EMIP的优势随着算例规模的增大而降低.对于ISA,随着算例规模的增大,可以找到比EMIP更好或相同的解,且数量逐渐增加.可见,ISA算法的优势会随着算例规模的增大而更加明显.

为了说明ISA比传统的启发式算法效果更好,本文对比了ISA和GA的求解质量.表7详细呈现了两种算法对不同规模算例的求解质量,其中ISA、Tie和GA分别代表ISA所求节拍时间更短、两者节拍时间相等和GA所求节拍时间更短.结果表明,对于不同规模的算例,ISA均能在40个算例中找到36个及以上比GA更好的解,且求解速度更快.

表 6 ISA 与 EMIP 的求解质量对比

		小规模算例 ( $N = 20$ )			中规模算例 ( $N = 50$ )			大规模算例 ( $N = 100$ )		
		ISA	Tie	EMIP	ISA	Tie	EMIP	ISA	Tie	EMIP
$n$		3	7	30	12	2	26	40	0	0
EMIP	CPU	1 800.0	26.7	919.4	1 800.0	1 800.0	1 800.0	1 800.0	—	—
	$\sigma$	(0)	(42.5)	(881.1)	(0)	(0)	(0)	(0)	—	—
ISA	CPU	35.8	14.5	27.5	87.2	91.0	81.6	208.0	—	—
	$\sigma$	(25.0)	(19.9)	(24.8)	(38.0)	(39.7)	(39.8)	(53.2)	—	—

表 7 ISA 与 GA 的求解质量对比

		小规模算例 ( $N = 20$ )			中规模算例 ( $N = 50$ )			大规模算例 ( $N = 100$ )		
		ISA	Tie	GA	ISA	Tie	GA	ISA	Tie	GA
$n$		34	4	2	39	0	1	36	0	4
GA	CPU	72.0	52.6	58.6	294.0	—	230.9	1 368.5	—	1 325.4
	$\sigma$	(18.1)	(8.6)	(1.4)	(51.6)	—	(0)	(273.9)	—	(252.1)
ISA	CPU	31.0	8.0	7.4	85.2	—	49.4	211.4	—	180.4
	$\sigma$	(25.3)	(1.8)	(0.3)	(40.4)	—	(0)	(52.9)	—	(45.0)

此外,限于篇幅,各个算例的详细结果无法全部展示.为此,从小、中、大型算例中分别随机抽取 10 个算例,对比 4 种方法的求解结果,通过详细结果对比,得到以下结论: 1) 相同规模的算例求解难度有所不同,这主要归因于参数设置的差异,这也说明了人机协同问题的复杂性和多元性; 2) MIP 和 EMIP 难以求解中大规模算例, MIP 求解中规模算例的平均误差 Gap 在 40% 以上, EMIP 求解中规模算例的平均误差 Gap 在 26% 以上,而对于大部分大规模算例, MIP 和 EMIP 甚至无法在 1 800 s 内给出一个可行解,这进一步说明了 ALBP-HRC 问题的复杂度; 3) ISA 算法表现较为稳健,相比于 MIP、EMIP 和 GA 算法, ISA 可以为各种规模的算例找到较好的可行解或最优解,且求解时间变化不大,这也说明 ISA 算法对于求解复杂人机协同装配线平衡问题具有较好的适用性.

### 3.3 关键参数影响分析

为了量化机器人的作用,本文分析了机器人密度对节拍时间优化的影响程度  $\beta$ . 在给定的任务密度下,以人工装配线的节拍时间  $\bar{c}$  为基准,节拍时间因机器人密度增加而减少的时间为  $\Delta c$ ,影响程度的计算公式为  $\beta = \Delta c / \bar{c}$ . 根据上述计算方法,计算机器人密度对节拍时间优化的影响,结果见表 8.

根据表 8,可以看出提高机器人密度可以大幅度减少节拍时间,随着机器人密度增加,节拍时间改善的幅度会降低,其趋势符合边际效用递减规律. 从研究结果中可以得出以下管理启示: 首先,企业通过合理部署协作机器人,可以有效缩短装配线的节拍时间,相比于人工装配线,人机协作装配线可以大大提

高生产效率,提升企业的竞争力. 其次,协作机器人的密度越大,装配线的改善效果越明显,但边际效用也会随之降低.

表 8 机器人密度对节拍时间的影响 %

RD	$N = 20$	$N = 50$	$N = 100$
0	0	0	0
0.2	5.92	5.23	12.1
0.4	12.22	8.77	15.5

## 4 结 论

近年来,不断提高的自动化程度改善了现代制造企业的生产效率. 作为一种新兴装配资源,协作机器人受到越来越多的关注,它可以与工人以独立、并行或协作的方式进行装配,在提升装配线自动化程度的同时,又保留了人工处理任务的灵活性,研究这种人机协同装配模式具有重要的理论意义和实践价值.

本文针对人机协同装配线平衡问题开展研究,以最小化节拍时间为优化目标,提出了新的节拍时间下限计算方法,设计了基于贪心策略的多阶段算法用于计算问题的上限和初始解,并建立了有效的 EMIP 模型;同时,本文设计了一类改进的模拟退火算法来实现对问题的高效求解;再者,本文构造了不同规模的 120 个测试算例,从多个维度对比了 4 种求解方法. 实验结果表明, EMIP 的求解质量和速度均优于 MIP, ISA 算法在求解结果和计算时间方面具有较好的稳健性, ISA 在求解大规模问题时具有显著优势,优于 EMIP 和传统的 GA 算法;此外,本文对相关参数开展了敏感性分析,随着协作机器人密度增大,装配效率提升,但边际效用下降,揭示了管理启示,为制

造企业引入协作机器人开展人机协同装配提供决策参考。

未来研究可从以下几个方面展开: 1) 考虑机器人不是由外部提供, 而是需要企业自行购置, 此时需要确定经济上最优情况下的机器人数量; 2) 考虑到上述情况, 构造并解决多目标优化问题; 3) 分析任务灵活性等参数变化对结果的影响; 4) 本文仅考虑将协作机器人引入最简单的装配线平衡问题, 可进一步拓展考虑其他更复杂类型的装配线平衡问题。

### 参考文献(References)

- [1] Weckenborg C, Spengler T S. Assembly line balancing with collaborative robots under consideration of ergonomics: A cost-oriented approach[J]. IFAC-PapersOnLine, 2019, 52(13): 1860-1865.
- [2] 郑晨昱, 李梓响, 唐秋华, 等. 基于改进人工蜂群算法的人机协作装配线平衡优化[J]. 计算机集成制造系统, <https://kns.cnki.net/kcms/detail/11.5946.tp.20220715.1002.006.html>. (Zheng C Y, Li Z X, Tang Q H, et al. Balancing man-robot cooperative assembly line utilizing enhanced artificial bee colony algorithm[J]. Computer Integrated Manufacturing Systems, <https://kns.cnki.net/kcms/detail/11.5946.tp.20220715.1002.006.html>.)
- [3] Kinast A, Braune R, Doerner K F, et al. A hybrid metaheuristic solution approach for the cobot assignment and job shop scheduling problem[J]. Journal of Industrial Information Integration, 2022, 28: 100350.
- [4] Salvesson M E. The assembly-line balancing problem[J]. Journal of Fluids Engineering, 1955, 77(6): 939-947.
- [5] Betts J, Mahmoud K I. A method for assembly line balancing[J]. Engineering Costs and Production Economics, 1989, 18(1): 55-64.
- [6] 刘俨后, 麻娟, 左敦稳, 等. 基于改进烟花算法的随机装配线混流调度[J]. 控制与决策, 2019, 34(4): 751-758. (Liu Y H, Ma J, Zuo D W, et al. Mixed-model sequencing of stochastic assembly line based on improved fireworks algorithm[J]. Control and Decision, 2019, 34(4): 751-758.)
- [7] Uğurdağ H F, Rachamadugu R, Papachristou C A. Designing paced assembly lines with fixed number of stations[J]. European Journal of Operational Research, 1997, 102(3): 488-501.
- [8] Peeters M, Degraeve Z. An linear programming based lower bound for the simple assembly line balancing problem[J]. European Journal of Operational Research, 2006, 168(3): 716-731.
- [9] Wang X V, Kemény Z, Vánca J, et al. Human-robot collaborative assembly in cyber-physical production: Classification framework and implementation[J]. CIRP Annals, 2017, 66(1): 5-8.
- [10] Pellegrinelli S, Orlandini A, Pedrocchi N, et al. Motion planning and scheduling for human and industrial-robot collaboration[J]. CIRP Annals, 2017, 66(1): 1-4.
- [11] Tsarouchi P, Makris S, Chryssolouris G. Human-robot interaction review and challenges on task planning and programming[J]. International Journal of Computer Integrated Manufacturing, 2016, 29(8): 916-931.
- [12] Weckenborg C, Kieckhäfer K, Müller C, et al. Balancing of assembly lines with collaborative robots[J]. Business Research, 2020, 13(1): 93-132.
- [13] Dalle Mura M, Dini G. Designing assembly lines with humans and collaborative robots: A genetic approach[J]. CIRP Annals, 2019, 68(1): 1-4.
- [14] Raatz A, Blankemeyer S, Recker T, et al. Task scheduling method for HRC workplaces based on capabilities and execution time assumptions for robots[J]. CIRP Annals, 2020, 69(1): 13-16.
- [15] Mao Z F, Zhang J X, Fang K, et al. Balancing U-type assembly lines with human-robot collaboration[J]. Computers & Operations Research, 2023, 159: 106359.
- [16] Liu L, Guo F, Zou Z S, et al. Application, development and future opportunities of collaborative robots (cobots) in manufacturing: A literature review[J]. International Journal of Human-Computer Interaction, <https://doi.org/10.1080/10447318.2022.2041907>.
- [17] International Organization for Standardization. 2010.ISO 13855:2010: Safety of machinery—Positioning of safeguards with respect to the approach speeds of parts of the human body[S]. Geneva: International Standards Organization, 2010.
- [18] Li Z X, Kucukkoc I, Nilakantan J M. Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem[J]. Computers & Operations Research, 2017, 84: 146-161.
- [19] Scholl A, Klein R. SALOME: A bidirectional branch-and-bound procedure for assembly line balancing[J]. INFORMS Journal on Computing, 1997, 9(4): 319-334.
- [20] Suman B, Kumar P. A survey of simulated annealing as a tool for single and multiobjective optimization[J]. Journal of the Operational Research Society, 2006, 57(10): 1143-1160.
- [21] Otto A, Otto C, Scholl A. Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing[J]. European Journal of Operational Research, 2013, 228(1): 33-45.

### 作者简介

毛照昉(1977—), 男, 教授, 博士, 博士生导师, 从事智能制造、精益供应链管理、运营管理等研究, E-mail: maozhaofang@tju.edu.cn;

王威(1999—), 女, 博士生, 从事装配线平衡、车间调度等研究, E-mail: ww0710@tju.edu.cn;

方侃(1985—), 男, 副教授, 博士, 从事组合优化、生产调度优化等研究, E-mail: kfang@tju.edu.cn;

黄典(1989—), 男, 讲师, 博士, 从事装配线平衡、人机协同调度优化等研究, E-mail: huangdian@tju.edu.cn.