

湖南大学

HUNAN UNIVERSITY

工程优化方法 课程报告

报告题目：面向制造运维集成的备件生产

和服务人员协同调度研究

学生姓名：胡洋洋

学生学号：S2302W0240

专业班级：机械 2303

学院名称：机械与运载工程学院

指导老师：王琥

2024 年 7 月 14 日

目录

摘要.....	
Abstract.....	
1. 问题背景.....	1
2. PIM-DCSP 模型构建	4
2.1 问题描述.....	4
2.2 数学模型.....	5
3. IACA 算法设计	10
3.1 IACA 算法框架.....	10
3.2 染色体编码.....	10
3.3 初始化.....	11
3.4 交叉.....	12
3.5 变异.....	14
3.6 生产相关的局部搜索.....	15
3.7 维护相关的局部搜索.....	17
4. 实验分析.....	20
4.1 算例构造.....	20
4.2 参数校准.....	20
4.3 PIM-DCSP 模型的有效性	22
4.4 IACA 算法特殊策略的有效性.....	23
4.5 和其他算法比较.....	24
5. 结论.....	28
参考文献.....	29

摘要

以往对分布式生产调度的研究主要集中在供给侧,忽略了供给侧和需求侧的互联互通——供给侧备品备件的交付时间会影响需求方分布式设备的维护调度,而分布式设备的维护调度会影响备品备件供应的调度决策。此外,在实践中,库存是制造商和客户之间的重要纽带。因此,本文首先提出备件生产方和需求方信息反馈的一体化生产、库存和不完善维护的最优调度问题 (production, inventory and imperfect maintenance with mutual feedback of supply side and demand side), 简称 PIM-DCSP。在 PIM-DCSP 中,生产资源和库存共同作用,为需求方提供备品备件,而需求侧则对其分布的设备进行不完善的维护调度,以延缓设备的劣化,最终延长设备的运行时间。

PIM-DCSP 的目标是制定一个最优调度,共同优化双方的调度,即合理安排生产资源、库存和工人,分别实现供应商总成本和需求方总成本的最小化。在本文中建立了数学模型来描述所提出的问题,并设计了一种改进的自适应合作算法 (IACA)。开发包括 2 种启发式初始化方法、6 种面向问题和 2 种随机局部搜索结构在内的有效算子,以增强种群多样性和搜索能力。在 96 个实例上进行了 IACA 与其他 3 种优秀算法的对比实验,证明了 IACA 在求解 PIM-DCSP 方面的优越性。

关键词: 分布式置换流水车间调度; 生产与库存; 维护; 协同进化算法

Abstract

The previous research on distributed production scheduling focuses on supply side, ignoring the interconnection of supply side and demand side: the delivery time of spare parts from the supply side will influence the maintenance scheduling of distributed equipment of demand side, while the maintenance scheduling of distributed equipment will affect the scheduling decision of supply of spare parts. In addition, in practice, inventory is an important link between manufacturers and customers. Therefore, we firstly propose an optimal scheduling problem of integrated production, inventory and imperfect maintenance with mutual feedback of supply side and demand side, shortened to PIM-DCSP.

In PIM-DCSP, production resources and inventory work together to provide spare parts for demand side, while demand side makes imperfect maintenance scheduling for its distributed equipment to postpone deterioration and finally extend the operating time of equipment. The goal of PIM-DCSP is to make an optimal scheduling that jointly optimizes the scheduling of both sides, that is, reasonably arrange production resources, inventory and workers to realize the minimization of the total cost of supplier and the total cost of demander respectively. A mathematical model is established to describe the presented problem and an improved adaptive cooperative algorithm (IACA) is designed. Effective operators including two heuristic initialization methods, six problem-oriented and two random local search structures are developed to strengthen population diversity and search capability. The comparison experiment of IACA and three other outstanding algorithms is carried out on 96 instances, and the superiority of IACA in solving PIM-DCSP is certificated thoroughly.

Keywords: Distributed permutation flow-shop scheduling; Production and Inventory; Maintenance; Cooperative evolution algorithm

1. 问题背景

迅速发展的新一代信息技术，正在以前所未有态势推动社会生产方式变革。为抢占未来产业竞争制高点，强化国家核心竞争力，众多发达国家纷纷提出深度融合信息化与工业化的“再工业化”战略，如德国的“工业 4.0”。与此同时，我国也相继提出《中国制造 2025》和“互联网+”协同制造^[1, 2]。区别于传统的生产型制造，现代网络协同制造强调系统工程的思想，充分利用互联网和各种集成技术来打破时间、空间上的约束，将制造、运行、维护、管理等各环节紧密连接，使产品制造、服务生命周期的经济、环境、社会等综合效用达到最优化^[3-5]。

其中，智能生产调度作为网络协同制造生产管理的核心环节，其决策方案的优劣对协同制造的服务水平、效率和成本产生重大影响^[6-8]。现有的生产调度理论主要针对传统制造模式，并围绕以下几个方面展开：①车间类型（并行机、流水车间、作业车间等）^[9-14]；②优化目标（经济指标、环境指标等）^[15-17]；③生产过程关键约束（流程约束、人因、设备批处理特性等）^[18-20]；④不确定事件（紧急订单、机器故障等）^[21-23]；⑤生产环境（单工厂与多工厂、同构工厂与异构工厂等）^[19, 24]；⑥集成调度（生产调度与物流的集成、生产调度与设备维护的集成等）^[25-27]。整体来看，目前生产调度研究中的约束条件、优化目标都是以生产服务商为主体，处于产品生命周期的生产制造阶段，属于单制造域内的优化。

此外，运维服务作为产品生命周期中另一个重要环节，对保障装备安全可持续运行、提升装备效用也具有重要作用。与之相关的研究主要集中于：①运行策略的主动优化（装备部件失效与运行参数的动态适配）^[28-30]；②维修策略与计划（从传统的“事后维修”与“计划维修”向基于状态监测技术的“视情维护”和“预知维护”转变^[31-33]）；③维护资源管理与调度（备件需求预测与库存控制^[34, 35]、服务工程师配置与路径规划^[36, 37]、多维服务资源协同优化^[38, 39]等）。以上研究将运维服务作为一个独立的环节，注重于运行策略、维护计划以及维护资源的单独优化，进而保障装备运维服务的及时、可靠与经济性，属于单运维域内的优化。

由此可见，现有研究基本上都将装备生产调度与运维服务割裂开来，作为两个独立的阶段进行决策与优化，以期实现单制造域或单运维域的最优。这种局部最优模式忽略了装备生产调度与运维之间的相互影响与关联，难以兼顾跨阶段、多资源的统筹优化，导致装备制造/运维集成程度低、装备效用发挥受限等诸多问题。尤其是对于复杂装备的生产调度与运维服务集成而言（本论文所指复杂装备主要为复杂机械装备，如隧道掘进机、轨道机车、通用工程机械等），由于其制造及运维的复杂特点，导致备件生产调度决策优化面临诸多挑战性难题，主要

体现在以下几个方面：

(1) 复杂装备不仅部件种类繁多、零部件制造呈现多场点分散的特点，而且其制造与运维在时空上存在高度交叉。以大国重器-隧道掘进机为例，它包含了掘进、出渣、支护、运输、吊装、供水、供电、通风等多种设备群组，零部件数量多达数千乃至上万种，制造场地分散在全国各地，导致设备群组之间的协同生产调度难度大。此外，与传统在工厂内部装配不同，其整机装配多为在施工现场集中进行装配，并且经常伴有制造与运维在施工现场交织进行的特征，这使得装备的生产调度与运维往往密不可分，二者之间的信息互馈需要高度集成。传统的单制造域生产调度方法不能有效满足这种协同互馈要求，导致装备的运维效用低。

(2) 由于装备的作业环境大多复杂多变，运行过程中其核心部件不可避免地经历各种退化，造成装备性能下降甚至停机。为保障装备安全可持续运行和工程项目的有序推进，核心制造企业建立了庞大的多级配件链以确保备件及时供应，避免（或降低）由于维护延误而造成的停机损失。但现有的多级配件链管理多采用“从生产，到库存，到运维”的串行、分阶段决策模式，难以实现制造、运维全生命周期全局资源的统筹优化，导致备件库存积压浪费严重、装备效用发挥受限等一系列问题。

因此，有必要打破传统的单制造域、单运维域决策优化的固有模式，构建制造/运维跨域协同优化的新模式。这种新模式需要综合考虑产品生命周期中制造、流通、运维多个域的生产、运输、仓储、运行监测、维修维护、备件供应等业务活动，涉及运维服务需求方及其对应的备件生产商、物流商、运维服务商等不同主体，由此对现有的生产调度理论与方法研究提出了新的要求与挑战：不仅能够根据运维需求实现制造域内生产资源的调度优化，而且能够通过“生产调度”这根纽带，反向推动运维域内的装备效用提升，进而实现装备制造、运维生命周期的跨域集成优化。

图 1 展示了一种新的面向装备运维需求方、涵盖分布式生产资源提供方和多级备件仓库的网络协同制造/运维服务场景。该场景强调装备运维、生产制造和备件库存三者之间的信息互馈及业务协作。其中，分布式生产资源与多级仓储相互协同，并行为装备运维提供备件，但同时又受多场点装备运维决策的反馈影响。与传统面向单制造域的生产调度相比较，网络协同制造/运维服务场景下进行生产调度时，面临着同时满足生产资源利用、备件库存成本、维护策略与运行效用综合最优化的挑战。这包括两方面的内涵：一是通过生产调度统筹分布式生产资源并协同备件库存优化，来满足装备运维的备件需求；二是通过对运行参数和维护策略的优化，调节部件的失效时点，降低对生产资源的响应时效性需求及备件

库存水平,实现生产资源利用、备件库存设置、维护策略与运行效用的综合最优。然而,目前尚未见有关制造/运维跨域含备件库存协同优化的分布式生产调度研究报道。

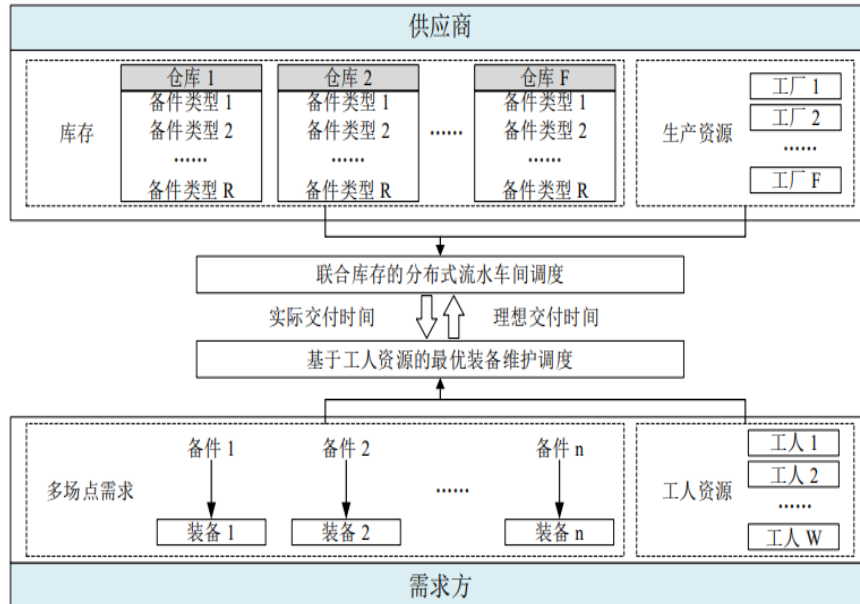


图1 “装备运维-备件库存-生产资源”集成的网络协同制造/运维服务场景

为此,本文在国家“加快发展新质生产力”的宏观政策指引下,针对网络协同制造战略需求,开展面向装备运维效用与库存控制协同优化的备件分布式生产调度研究,将协同维护和库存控制策略引入生产调度决策模型,从系统工程角度出发,将装备维修维护、备件库存、生产资源作为一个整体进行统筹优化调度研究,拟通过“生产调度优化”这根纽带,实现分布式生产资源利用、备件库存设置与装备维护的跨域集成优化。这种全新的面向多主体、跨阶段的生产调度理论与方法,可以为突破“产品全生命周期跨阶段集成资源统筹优化及装备效用提升”瓶颈问题提供一种新的思路与方法,具有重要的理论研究意义。此外,由于复杂装备(例如隧道掘进机)大都单台套价值极高(少则成百上千万元,多则数亿元)、所涉及的施工项目更是金额巨大(几十甚至上百亿元),对国民经济发展及国家战略具有重要意义,因此本论文又具有广阔的工程应用前景及显著的经济价值。

2. PIM-DCSP 模型构建

2.1 问题描述

PIM-DCSP 由制造方和需求方组成。制造方统筹利用其生产资源和库存，为需求方提供生产调度计划。相应地，需求方对其装备合理安排不完美的维护 (IM) 活动，并将延长的交货期反馈给制造商。需要注意的是，IM 意味着在维护后，装备的健康状态只能得到部分改善，而不能恢复到初始状态。PIM-DCSP 的目的是联合优化双方相应的调度计划，将其总成本降至最低。

对于生产商来说，有 F 个分布的工厂 $\{F_1, F_2, \dots, F_F\}$ ，可以提供 R 种类型的备件。每个工厂由一个流水车间 (m 台机器 $\{M_1, M_2, \dots, M_m\}$) 和一个仓库组成，仓库预先存放一定数量和种类的备件。备品备件在仓库的存放会产生库存成本，同一类型备品备件单位时间的库存成本相等。在调度开始之前，生产商收到 n 个订单 $\{J_1, J_2, \dots, J_n\}$ 。每个订单 J_j 有一个初始交货期 d_j ，对应于 R 类备件中的一种。显然，每个订单应该由工厂提供或从仓库转移。

对于需求方来说，它拥有一个包含 n 套装备的装备群 $\{E_1, E_2, \dots, E_n\}$ 和一个工人团队 $\{W_1, W_2, \dots, W_w\}$ ，负责在备件到达之前执行 IM。每一个订单 J_j 是一台装备 E_j 的核心部件。每个 E_j 都有一个初始健康值 HV_j ，该健康值以一定的速率 v_j 恶化，意味着 J_j 的理想交付时间 d_j 为 HV_j/v_j 。由于工作环境的不同，不同装备的劣化率也不同。每个工人都熟练掌握 P 种维护策略 $\{S_1, S_2, \dots, S_P\}$ 中的一种。不同策略对装备的修复程度不同，需要不同的维护时间和维修成本，满足关系 1：维护时间越短，单位时间的维护成本越高。IM 后修正的理想交付时间表示为 Id_j 。如果 J_j 早于或晚于 Id_j 交付，将对生产商产生提前或延迟惩罚。

图 2 描绘了 IM 的实现机制。这表明只有当 E_j 的当前健康状态值 NHV_j 满足条件 1： $HV_j^- < NHV_j < HV_j^+$ 时，才可以执行 IM 活动。事实上，在 $[HV_j^-, HV_j^+]$ 区间范围之外的 IM 操作是没有意义且无用的。根据图 2，条件 1 可以转换成一个时间窗 $[t_j^-, t_j^+]$ ，相应变量之间的关系满足： $HV_j^- = HV_j - t_j^+ \times v_j$ 和 $HV_j^+ = HV_j - t_j^- \times v_j$ 。由于工人数量有限，只有当条件 1 和工人可用同时满足时，才能执行 IM 活动。

如图 2 所示，介绍了一种策略下 IM 活动的几种可能情况，其中 LR_s 和 Id_j 分别表示策略 S_s 的修复程度和维护后的理想交货期。

(1) 曲线“ABDEF”表示选中的工人在 t_j^- 或之前可用。在这种情况下，只有当健康值达到触发阈值 HV_j^+ 时，才能执行维护操作，其中“BD”表示维护过程。

(2) 曲线“ACGHI”表示工人在 t_j^+ 前不可用，其中“CG”表示维护过程。

(3) 曲线“ACJKL”表示装备需要停止工作等待工人。在这种情况下，“CM”表示等待过程，“MJ”表示维护过程。

(4) 如果工人在 t_j^- 和 t_j^+ 之间可用, 则维护将在该工人可用时立即执行。

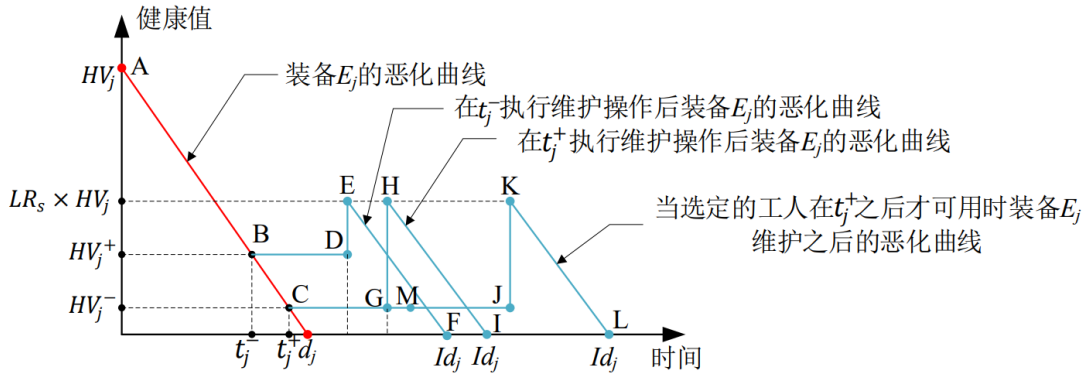


图2 几种维护情况的描述

需要注意的是：（1）加工时间是提前知道的，随订单类型的不同而不同，维护时间也是提前知道的，随策略的不同而不同；（2）已知所有工厂/仓库与所有需求点之间的运输时间，在假设工厂/仓库到需求点之间的运输时间小于所有备件的截止交货期的前提下，从仓库转出的订单将按时交付；（3）如果在所有订单完成后仍有备品备件留在仓库，则存储时间按公式（1）计算；（4）考虑工人在两个不同工作场所之间的转移时间，且已知。

$$T = \frac{\sum_{j=1}^n \sum_{r=1}^R \sum_{k=1}^m P_{rk} \times Q_{jr}}{F} \quad (1)$$

在整个调度问题中，有一些需要注意的假设：（1）每台装备至多执行一次IM活动；（2）对于制造商而言，单位时间延误的惩罚大于单位时间的运输成本，单位时间的运输成本大于单位时间的储存成本；（3）对需求方而言，单位时间停机损失大于单位时间部件剩余价值浪费；（4）工厂的机器在零时刻可用；（5）不考虑工人开始维护第一台装备的转移时间。

2.2 数学模型

基于对 PIM-DCSP 模型的描述，建立相应的数学模型。下面首先列出了模型中使用的索引、参数和决策变量。

表1 模型中相关符号

符号	含义
i, j	订单索引, $i, j=0, 1, \dots, n$, 其中 0 表示虚拟订单
f, h	工厂或仓库索引
k	机器索引
q	每个工厂内的加工位置索引
g	每个工人上的维护位置索引
r	订单类型索引
s	维护策略索引
e	维护工人索引
n	订单数
m	工厂内机器数

续上表:

符号	含义
F	工厂和仓库数
R	订单类型数
P	策略个数
ws	一种策略对应的工人数量
w	工人总数, $w=ws \times P$
F_f/C_f	第 f 个工厂/仓库
TF_{fj}	工厂 F_f 或 C_f 到需求点 j 间的运输时间
TR_{ij}	工人在需求点 i 和 j 之间的转移时间
P_{rk}	类型为 r 的订单在机器 M_k 上的加工时间
SC_r	类型为 r 的订单单位时间的库存成本
TC	单位时间的运输成本
DC_j	订单 J_j 的延迟惩罚权重
PC_j	装备 E_j 单位时间的停机损失
WC_j	对 E_j , 更换备件后旧部件单位时间的剩余价值
MC_e	工人 e 单位时间的维护成本
TM_e	工人 e 的维护时间
v_j	装备 E_j 的恶化速率
HV_j	装备 E_j 的初始健康值
LR_s	维护策略 s 对应的修复程度
N_{jr}	仓库 C_f 中类型为 r 的备件的储存数量
d_j	订单 J_j 的初始理想交货期 j
$[t_j^-, t_j^+]$	装备 E_j 的维护开始时间窗口
B	一个足够大的正数
N	集合 $\{1, 2, \dots, n\}$
SM	集合 $\{1, 2, \dots, m\}$
SF	集合 $\{1, 2, \dots, F\}$
SW	集合 $\{1, 2, \dots, w\}$
SR	集合 $\{1, 2, \dots, R\}$
Q_{jr}	如果订单 J_j 属于类型 r 为 1; 否则为 0
U_{es}	如果工人 e 使用维护策略 s 为 1; 否则为 0

注: Q_{jr} 和 U_{es} 提前已知。

◆决策变量:

$$x_{jfq} = \begin{cases} 1, & \text{如果 } J_j \text{ 安排到工厂 } F_f \text{ 的第 } q \text{ 个位置} \\ 0, & \text{否则} \end{cases}$$

$$y_{jh} = \begin{cases} 1, & \text{如果 } J_j \text{ 由仓库 } C_h \text{ 供应} \\ 0, & \text{否则} \end{cases}$$

$$z_{jeg} = \begin{cases} 1, & \text{如果 } J_j \text{ 所属装备分配到工人 } e \text{ 的第 } g \text{ 个位置执行维护活动} \\ 0, & \text{否则} \end{cases}$$

 S_{jk} : 订单 J_j 在机器 M_k 上的加工开始时间 C_{jk} : 订单 J_j 在机器 M_k 上的加工结束时间 BT_{je} : 装备 E_j 在工人 e 上执行维护活动的开始时间 ET_{je} : 装备 E_j 在工人 e 上执行维护活动的结束时间

Rd_j : 订单 J_j 的预计交付时间

Id_j : 订单 J_j 的理想交货时间

◆目标函数:

(1) 最小化供应商的总成本 (ob_1)

$$\min ob_1 = CS + CT_1 + CED \quad (2)$$

其中 CS、CT1、CED 分别代表供应商的库存成本、运输成本、提前和延误成本。分别由式 (3) – (5) 计算。

$$CS = \sum_{r=1}^R SC_r \times \left[\sum_{j=1}^n \sum_{h=1}^F Q_{jr} \times y_{jh} \times (Id_j - TF_{hj}) + T \times \left(\sum_{h=1}^F N_{hr} - \sum_{j=1}^n \sum_{h=1}^F Q_{jr} \times y_{jh} \right) \right] \quad (3)$$

$$CT_1 = \sum_{j=1}^n \sum_{f=1}^F \sum_{q=1}^n x_{jfq} \times TF_{fj} \times TC \quad (4)$$

$$CED = \sum_{r=1}^R \sum_{j=1}^n \sum_{f=1}^F \sum_{q=1}^n Q_{jr} \times x_{jfq} \times \left[\max(0, Id_j - Rd_j) \times SC_r + \max(0, Rd_j - Id_j) \times DC_j \right] \quad (5)$$

(2) 最小化需求方的总成本 (ob_2)

$$\min ob_2 = CM + CT_2 + CP + CW \quad (6)$$

其中 CM 表示维修成本, CT2 表示工人在不同需求点之间的转移成本, CP 表示备件延迟交付造成的停工损失, CW 表示备件提前交付造成的旧部件的剩余价值。分别由式 (7) – (10) 计算。

$$CM = \sum_{j=1}^n \sum_{e=1}^w \sum_{g=1}^n z_{jeg} \times TM_e \times MC_e \quad (7)$$

$$CT_2 = \sum_{e=1}^w \sum_{j=1}^n \sum_{i=1}^n \sum_{g=1}^{n-1} \max \left\{ (x_{jeg} + x_{ie(g+1)} - 1) \times TR_{ji} \times TC, 0 \right\} \quad (8)$$

$$CP = \sum_{j=1}^n \sum_{e=1}^w \sum_{g=1}^n \left[\max(0, BT_{je} - t_j^+) \times TM_e \right] \times PC_j \times z_{jeg} + \sum_{j=1}^n \sum_{e=1}^w \sum_{g=1}^n z_{jeg} \times \max(0, Rd_j - Id_j) \times PC_j \quad (9)$$

$$CW = \sum_{j=1}^n \sum_{e=1}^w \sum_{g=1}^n z_{jeg} \times \max(0, Id_j - Rd_j) \times WC_j \quad (10)$$

◆生产和供应调度约束:

任何订单只能通过在工厂加工或从仓库中调拨来完成。

$$\sum_{f=1}^F \sum_{q=1}^n x_{jfq} + \sum_{h=1}^F y_{jh} = 1 \quad \forall j \in N \quad (11)$$

任一工厂的任一位置一次最多可以处理一个订单。

$$\sum_{f=1}^F \sum_{q=1}^n x_{jfq} + \sum_{h=1}^F y_{jh} = 1 \quad \forall j \in N \quad (12)$$

任意工厂某一加工位置前没有空缺。

$$\sum_{f=1}^F \sum_{q=1}^n x_{ifq} + \sum_{h=1}^F y_{jh} = 1 \quad \forall j \in N \quad (13)$$

从某一仓库调拨的某种类型的订单数量，不得超过该仓库原有的存储容量。

$$\sum_{j=1}^n y_{jh} \times Q_{jr} \leq N_{hr} \quad \forall h \in SF, r \in SR \quad (14)$$

每台机器上一个订单的开始时间和完成时间之间的关系。

$$C_{jk} = S_{jk} + \sum_{f=1}^F \sum_{q=1}^n \sum_{r=1}^R x_{ifq} \times Q_{jr} \times P_{rk} \quad \forall j \in N, k \in SM \quad (15)$$

一台机器上两个相邻操作的加工顺序约束，即同一台机器的两个相邻操作，只有在前一个操作完成后，才能进行后一个操作。

$$S_{ik} \geq C_{jk} + B \times (x_{if(q-1)} + x_{ifq} - 2) \quad \forall i \in N, j \in N \cup \{0\}, q \in N, k \in SM, f \in SF \quad (16)$$

订单在两个相邻机器上的加工顺序。

$$S_{jk} \geq C_{j(k-1)} \quad \forall j \in N, k \in SM \quad (17)$$

出每个订单的实际交货时间。

$$Rd_j = \max \left\{ C_{jm} + \sum_{f=1}^F \sum_{q=1}^n x_{ifq} \times TF_{fj} + \sum_{h=1}^F y_{jh} \times TF_{hj}, Id_j \right\} \quad \forall j \in N \quad (18)$$

从仓库调拨的订单 J_j 的 C_{jk} 值始终为 0。

$$-B \times \left(1 - \sum_{h=1}^F y_{jh} \right) \leq C_{jk} \leq B \times \left(1 - \sum_{h=1}^F y_{jh} \right) \quad \forall j \in N, k \in SM \quad (19)$$

初始条件：

$$C_{0k} = 0, x_{0f0} = 1, x_{jf0} = 0 \quad \forall j \in N, k \in SM, f \in SF \quad (20)$$

◆维护调度相关约束：

一个工人的任意一个位置一次最多执行一个维护活动。

$$\sum_{j=1}^n z_{jeg} \leq 1 \quad \forall e \in SW, g \in N \quad (21)$$

工人上任一维护位置前没有空缺。

$$\sum_{j=1}^n z_{jeg} \geq \sum_{j=1}^n z_{je(g+1)} \quad \forall e \in SW, g \in N \setminus \{n\} \quad (22)$$

每个工人任一 IM 活动的开始时间和完成时间之间的关系。

$$ET_{je} = BT_{je} + \sum_{g=1}^n z_{jeg} \times TM_e \quad \forall j \in N, e \in SW \quad (23)$$

两个相邻的 IM 活动在一个工人上的处理顺序约束。

$$BT_{ie} \geq ET_{je} + TR_{ij} + B \times (z_{je(g-1)} + z_{ieg} - 2) \quad \forall i \in N, j \in N \cup \{0\}, e \in SW, g \in N \quad (24)$$

每台装备的 IM 活动的启动时间约束。

$$\sum_{g=1}^n z_{jeg} \times t_j^+ \geq BT_{je} \geq \sum_{g=1}^n z_{jeg} \times t_j^- \quad \forall j \in N, e \in SW \quad (25)$$

E_j 在无维护情况下的 ET_{je} 值为 0。

$$-B \times \left(1 - \sum_{g=1}^n z_{jeg}\right) \leq ET_{je} \leq B \times \left(1 - \sum_{g=1}^n z_{jeg}\right) \quad \forall j \in N, e \in SW \quad (26)$$

每个订单的理想交货时间。

$$Id_j = \sum_{e=1}^w \sum_{g=1}^n (1 - z_{jeg}) \times d_j + \sum_{e=1}^w \sum_{g=1}^n z_{jeg} \times \left[ET_{je} + HV_j \times \frac{\sum_{s=1}^P (U_{es} \times LR_s)}{v_j} \right] \quad \forall j \in N \quad (27)$$

初始条件：

$$z_{je0} = 0, z_{0e0} = 1, ET_{0e} = 0 \quad \forall j \in N, e \in SW \quad (28)$$

3. IACA 算法设计

鉴于近年来 NSGA-II 在多目标调度问题中的广泛应用以及 PIM-DCSP 模型的复杂性, 本文提出了一种基于 NSGA-II 的改进自适应协同进化算法 (IACA)。为了避免搜索不准确的缺点并获得更好的解决方案, 设计了几种重要的启发式初始化规则和局部搜索算子。

3.1 IACA 算法框架

IACA 的主要流程图如图 3 所示。每个主要步骤的细节将在后续章节中详细说明。根据 PIM-DCSP 中生产和维护调度结果相互反馈的特点, IACA 算法在每次迭代中依次对生产部分和维护部分进行优化。

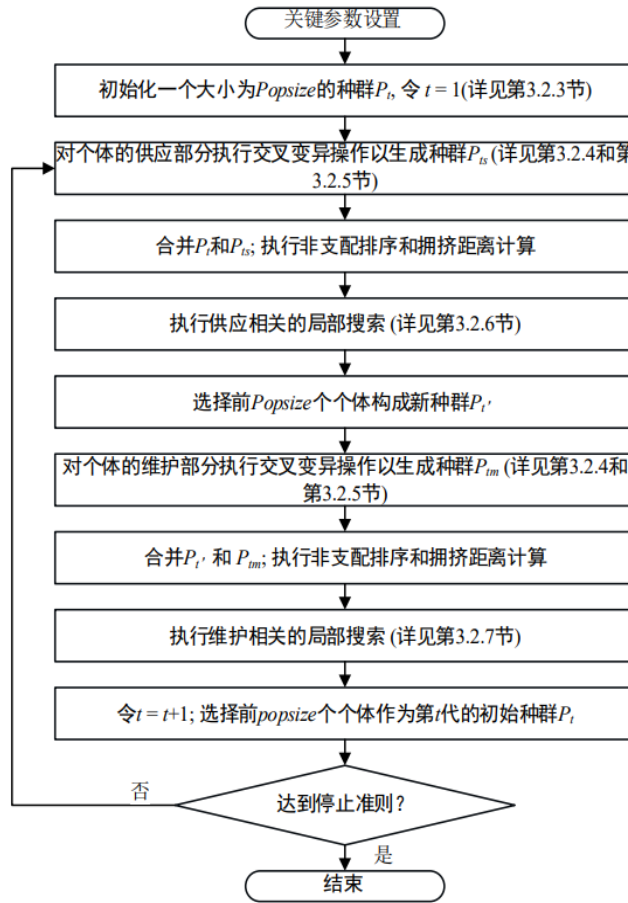


图 3 IACA 框架

3.2 染色体编码

根据第 2.1 节的问题描述, 有四个子问题需要解决: (1) 每个订单的工厂或仓库分配; (2) 各工厂的订单处理顺序; (3) 每台装备是否执行 IM 活动的决策以及进行 IM 活动的所有装备的工人分配; (4) 每个工人上的 IM 活动的执行先后顺序。为了解决这四个子问题, 本文设计了一种四层编码方法。它包括对每个订单生产模式的选择 (SMS), 即按工厂或按仓库、订单调度 (OS)、维

护工人分配（WA）和维护排序（MS）。

假设 PIM-DCSP 中有 n 个订单， F 个工厂， F 个仓库， W 个工人，OS 和 MS 分别由 n 个不重复的整数 1- n 组成。OS 中的每个基因代表一个订单，其处理顺序取决于其在 OS 中的位置。同样，MS 中的每个基因都表示一台装备，其 IM 活动的执行顺序由 MS 中的位置决定。SMS 和 WA 均表示为 $\{\pi_1, \pi_2, \dots, \pi_n\}$ 。对于 SMS， $\pi_j \in \{1, 2, \dots, 2 \times F\}$ 表示 J_j 选择的工厂还是仓库，其中 $\pi_j \leq F$ 表示选择工厂，否则表示选择仓库。对于 WA， $\pi_j \in \{0, 1, \dots, W\}$ 表示分配给 E_j 的工人，其中基因 0 表示相应装备上不执行 IM 活动。这种编码方法的一个示例如图 4 所示，在该示例中有 10 个订单、2 个工厂、2 个仓库和 2 个工人。图 4（b）为图 4（a）中染色体的基本安排和调度序列信息。

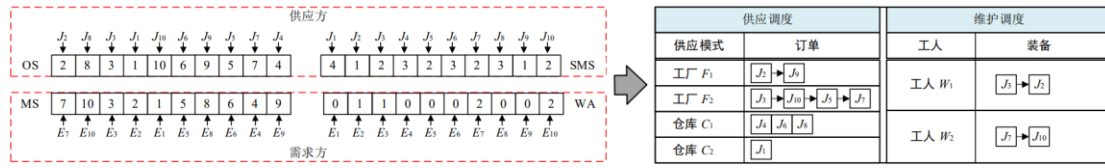


图 4 四层编码示例

3.3 初始化

初始化方法对最终解的质量起着至关重要的作用。为了同时保证解的多样性和高质量，提出了几种启发式初始化规则和随机初始化方法。另外，考虑到双方调度的独立性，生产部分和维护部分的初始化方法不同。完整的初始化方法如算法 1 所示。

表 2 算法 1 的伪代码

算法 1：初始化
1. For $i = 1: \text{Popsiz}$ do 2. 获取染色体供应部分的 SMS 序列 \leftarrow 算法 2 3. 生成一个 1 到 3 之间的随机整数 R_i 4. If $R_i = 1$ do OS \leftarrow 随机生成包含 n 个介于 1 和 n 之间的非重复整数的字符串 5. Else if $R_i = 2$ do OS \leftarrow 算法 3 中的 HA1_S 6. Else do OS \leftarrow 算法 3 中的 HA2_S 7. End if 8. WA \leftarrow 随机生成基因值属于 $\{0, 1, 2, \dots, W\}$ 的字符串 9. 和第四行一样随机生成一个字符串 MS 10. 获得个体 i : $Ch_i = [\text{OS SMS}; \text{MS WA}]$ 11. End for 12. 输出：初始种群 $P_t = \{Ch_1, Ch_2, \dots, Ch_{\text{popsiz}}\}$

对于染色体的生产部分，为了避免产生不可行的解决方案，应根据约束条件 (11) 和 (14) 为每个订单选择工厂或仓库。因此，优先生成生产模式的选择 (SMS)。相应的过程如算法 2 所示。考虑到交货期对生产调度的重要性，在该问题中，系

统采用了两个与交货期相关的启发式规则：基于流水车间中最后一台机器上的最小冗余（LSL），称为 HA1_S；考虑整个加工过程，基于第一台机器上的最小冗余（FSL），称为 HA2_S。算法 3.3 详细介绍了 HA1_S 和 HA2_S。值得注意的是，由于 LSL 和 FSL 都基于工厂生产订单已知的前提，所以应该首先生成 SMS。至于染色体的维护部分，MS 和 WA 都是随机生成的，因为工人没有负载限制，每个工人都有能力维护任何装备。

表 3 算法 2 的伪代码

算法 2: SMS 的生成
1. 从集合 $\{1,2\}$ 中随机选择一个数字 Rn 2. If $Rn=1$ do $J \leftarrow$ 随机打乱集合 $JJ=\{J_1, J_2, \dots, J_n\}$ 的顺序 3. Else do $J \leftarrow$ 根据订单到期日的非递减顺序将 JJ 中的订单排序 4. End if 5. For $i=1: n$ do 6. 找到 J 中的第 i 个订单，定义为 J_i ，并为其选择一种供应模式（ $sm=1$ 为由工厂供应； $sm=0$ 为由仓库供应） 7. If $sm=1$ do 从 $SF=\{1,2, \dots, F\}$ 中随机选择一个工厂 F_j 并将其填入 SMS 的第 j 个位置 8. Else do 找到仓库集合 WW，其包含的每个仓库都有和 J_i 相同类型的备件尚未被分配 9. If $WW \neq \emptyset$ do 从 WW 中随机选择一个仓库 C_j 并将 $(F + C_j)$ 填入 SMS 中 J_i 对应的位置 10. Else do 令 $sm=1$ 并返回第七行所示步骤 11. End if 12. End if 13. End for

表 4 算法 3 的伪代码

算法 3: HA1_S and HA2_S
1. 输入：算法 3.2 得到的 SMS 2. 确定分别由工厂和由仓库供应的订单集合 J_f 和 J_w 3. 对每个订单 $J_j \in J_f$ 执行：（1）HA1_S 中的 $Eq(1)$ ；（2）HA1_S 中的 $Eq(2)$ 4. $Eq(1)$ ：计算 $LSL_j = d_j - TF_{fj} - P_{rM}$ ，其中 f 表示为 J_j 选择的工厂而 r 表示 J_j 的类型 5. $Eq(2)$ ：计算 $FSL_j = d_j - TF_{fj} - \sum_{m=1}^M P_{rm}$ ，其中 f 表示为 J_j 选择的工厂而 r 表示 J_j 的类型 6. $OS_f \leftarrow$ 根据 LSL_j (HA1_S) 或 FSL_j (HA2_S) 的非递减顺序重新排列 J_f 7. $OS_w \leftarrow$ 随机打乱 J_w 的顺序 8. $OS = [OS_f, OS_w]$ 9. 输出：OS

3.4 交叉

本部分针对四个子问题设计了三种交叉方法，即 OS 和 MS 的随机多订单交叉方法（RMOX）、SMS 的有限仓库能力的合理交叉方法（LRX）和 WA 的多订单间隔交叉方法（IMX）。交叉概率记为 P_c 。对于 RMOX，以 OS 为例（MS

和 OS 的交叉是一样的)。如图 5 所示, P1_OS 和 P2_OS 被定义为 OS 的两个父代, 而 C1_OS 和 C2_OS 被定义为其子代。这种交叉方法由以下步骤组成: (1) 通过克隆 P1_OS 和 P2_OS 生成 C1_OS 和 C2_OS; (2) 随机确定任意个非重复整数组成的位置集, 位置个数不超过 n , 记为 PS; (3) 分别在 P1_OS 和 P2_OS 上找到 PS 对应的订单集 Set1 和 Set2; (4) 找到 C1_OS/C2_OS 与 Set2/Set1 的交集, 将 C1_OS/C2_OS 中找到的交集按 Set2/Set1 的顺序重新排序。

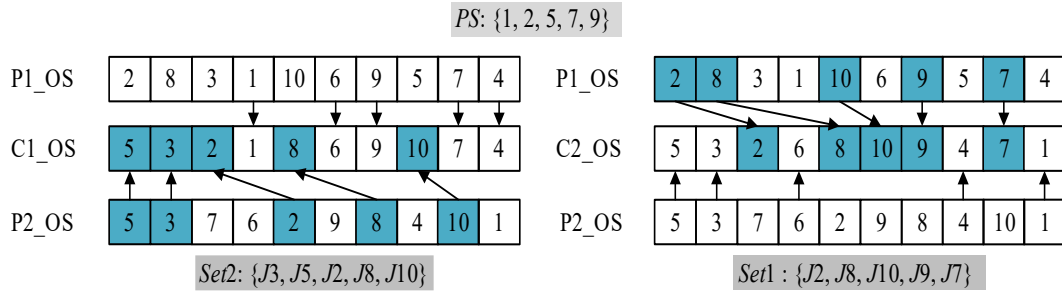


图 5 OS 上的 RMOX 交叉

对于 SMS, 使用 LRX 来避免由于仓库中备件库存不足而产生不可行的解决方案。如图 6 所示, P1_SMS 和 P2_SMS 代表 SMS 的父代, C1_SMS 和 C2_SMS 代表子代。子代通过以下步骤获得:

步骤 1: 随机选取一个大小为 n' 且满足 $n' \leq n$ 的订单集合 J' 。

步骤 2: 设置 $i=1$, 复制 P1_SMS/P2_SMS 为 C1_SMS/C2_SMS。

步骤 3: 确定 J' 中的第 i 个订单, 定义为 J_j 。

步骤 4: 在 P1_SMS 和 P2_SMS 中找到第 j 个位置的内容 $fw1$ 和 $fw2$ 。如果 $fw1=fw2$, 转至步骤 8。如果 $fw1 \neq fw2$, 讨论三种情况: (1) 如果满足 $fw1 \leq F$, $fw2 \leq F$ 的条件, 则转到步骤 4; (2) 如果 $fw1 \geq F$, $fw2 \geq F$, 则转步骤 5; (3) 否则请转步骤 6。

步骤 5: 通过交换 $fw1$ 和 $fw2$ 更新 C1_SMS 和 C2_SMS, 然后执行步骤 8。

步骤 6: 将 C1_SMS/C2_SMS 第 j 个位置的内容改为 $fw2/fw1$ 。检查 C1_SMS/C2_SMS 中从仓库 $(fw2-F)/(fw1-F)$ 生产的与 J_j 同类型备件数量是否少于仓库 $(fw2-F)/(fw1-F)$ 的原有仓储量。如果上述两个结果不全为真, 恢复 C1_SMS 和 C2_SMS 到步骤 4 中的样子, 然后转到步骤 8。

步骤 7: 将 C1_SMS/C2_SMS 第 j 个位置的内容改为 $fw2/fw1$ 。对于第 j 个位置的 fw 值大于 F 的序列, 检查该序列中从 $(fw-F)$ 仓库生产且与 J_j 同类型的备件数量是否大于 $(fw-F)$ 仓库中该类型的原有备件储存数量。如果为真, 保持 C1_SMS 和 C2_SMS 与步骤 4 中的一致, 然后执行步骤 8。

步骤 8: 用 C1_SMS/C2_SMS 替换 P1_SMS/P2_SMS。

步骤 9: 令 $i=i+1$, 并重复步骤 3-9 直至 $i > n'$ 。

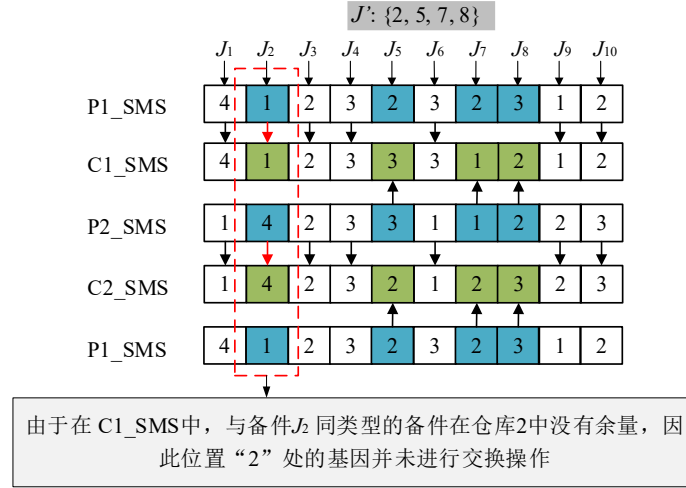


图 6 SMS 上的 LRX 交叉

对于 WA，提出了一种 IMX 交叉方法，如图 7 所示。父代分别用 P1_WA 和 P2_WA 表示，后代分别用 C1_WA 和 C2_WA 表示。具体操作过程如下图所示：

(1) 在 WA 长度范围内随机确定两个位置，标记为 p_1 和 p_2 ；(2) 得到 P1_WA/P2_WA 中 p_1 和 p_2 位置之间的基因串 WA1/WA2；(3) 将 P1_WA 和 P2_WA 上的 WA1 和 WA2 等位置交换得到 C2_WA 和 C1_WA。

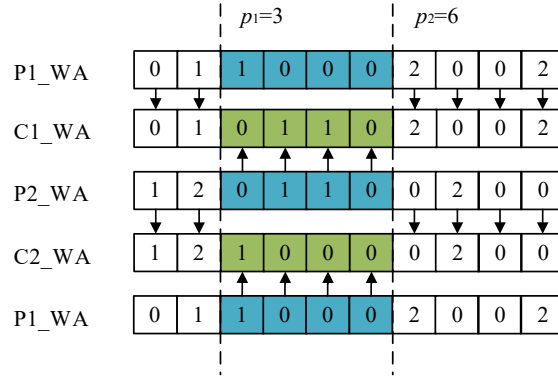


图 7 WA 上的 IMX 交叉

3.5 变异

为了避免陷入局部最优，需要在交叉后对种群进行变异。在突变过程中，和交叉一样，我们根据四个序列的不同性质采用不同的突变方法。OS 和 MS 采用“反转序列突变”（ISM），SMS 和 WS 采用“两点交换突变”（TEM），突变概率记为 P_m 。

以 OS 变异为例，ISM 如图 8 所示。PM_OS 为通过交叉得到的父代，CM_OS 为子代。首先，在 PM_OS 上随机生成两个位置，找到在这两个位置之间的基因串 J_M 。然后通过逆序 J_M 中的所有基因，得到一个新的基因串 J'_M 。最后，将 J'_M 填入 PM_OS 中原本 J_M 的位置生成 CM_OS，并将其他基因从 PM_OS 复制到 CM_OS 中。

对于 TEM 来说，SMS 和 WA 的过程基本相同。唯一的区别是在第一步中从

亲本中选择这两个基因的方式。WA 的方法是随机选择的，SMS 则是选择两个类型相同的备件基因。然后，交换这两个被选择的基因，得到后代。

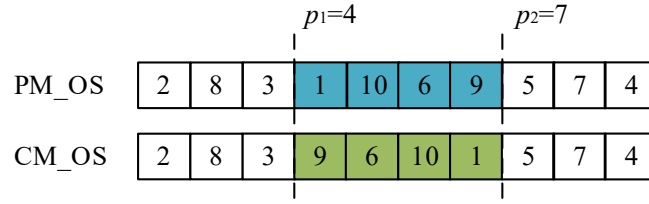


图 8 OS 上的 ISM 变异

3.6 生产相关的局部搜索

考虑到生产和维护调度之间反馈和相对独立的关系，这两个部分应该分别进行优化。因此，为每个部分设计了一系列问题导向的邻域搜索算子。本节介绍供应调度的搜索算子，而维护调度的搜索算子将在下一节提出并介绍。关于生产相关的局部搜索，为了进一步加快 IACA 算法的收敛速度，增强解的多样性，开发了 3 个降低目标 ob_1 的定向搜索算子 (LSO1_S、LSO2_S、LSO3_S) 以及 1 个随机搜索算子。详情按照他们在 IACA 中的执行顺序依次介绍如下。

局部搜索 LS1_P: 基于备件延误成本、库存和从仓库调拨的备件，对 ob_1 进行优化。由于仓库生产的订单没有延迟，对于延误成本最大的订单 J_j 来说，存在两种情况：（1）如果仓库里有与 J_j 相同类型的备件，直接从仓库调拨，代替 J_j 原本的生产方式，这样可以同时降低延误成本和库存成本；（2）否则，与另一个订单 $J_{j'}$ 交换生产模式，其中 $J_{j'}$ 与 J_j 类型相同且符合情境： $J_{j'}$ 来自仓库， $J_{j'}$ 的交货期大于 J_j 的（每个订单的交货期计算基于每一代的维护调度方案）以及 $DC_{j'} < DC_j$ 。显然，经过此操作，一方面，除了 J_j 和 $J_{j'}$ 之外的其他订单不会受到影响，因为它们属于同一类型，也就是说机器上的处理时间是相同的；另一面，当 $J_{j'}$ 的延误成本按预期降低时， J_j 的延误成本必然会增加，但总的来说，总延误成本会降低。伪代码在算法 4 中给出。

局部搜索 LS2_P: 基于工厂内部的作业交换操作，即对于延迟成本最大的 J_j ，将 J_j 和与 J_j 在同一工厂处理的另一个订单 $J_{j'}$ 进行交换。这两个订单除了满足 LS1_P 中的条件外，还需要满足一个额外的条件： J_j 是在 $J_{j'}$ 之后生产的。这个小扰动的结果是降低了整个工厂所有订单的总延误成本。算法 5 给出了具体的步骤。

局部搜索 LS3_P: 通过对随机两个从仓库调拨的相同类型的订单重新分配仓库，在不影响目标 ob_2 的情况下优化目标 ob_1 。即考虑到仓库与需求点之间的距离，为订单重新安排一个更合理的仓库，进一步降低运输成本。伪代码见算法 6。

随机搜索过程 LSS_S 是通过在两个不同工厂之间交换任意两个作业来实现的，在文献中得到了广泛的应用。注意，由于这四个搜索算子的角色不同，所以在每次迭代中都会执行它们。但是，每一代执行这些操作的个体 (SearchSize) 的数量是有限的，因为搜索过程相对耗时，后续小节将进行合理的设置。

表 5 算法 4 的伪代码

算法 4. 局部搜索 LSI_P
1. 输入: 个体 S
2. 找到延迟成本最大的订单, 记为 J_j
3. J_j 的类型、加工工厂和当前到期日分别记为 r, F_f 和 Id_j
4. 构造一个还有类型为 r 的备件剩余的仓库集合 RW
5. If $RW \neq \emptyset$ do
6. 生成一个新个体 $S' \leftarrow$ 从 RW 中随机选择一个仓库 $C_{f'}$ 用来替换 J_j 原来的供应模式 F_f
7. Else do 构造一个属于类型 r 且由仓库调拨的订单集合 J_w
8. If $J_w \neq \emptyset$ do
9. For $kk=1: J_w $ do
10. 提取 J_w 中的第 kk 个订单, 定义为 J_k , 并定义它的当前到期日为 Id_k
11. If $Id_j < Id_k$ and $DC_j > DC_k$ do
12. 获得一个新个体 $S' \leftarrow$ 交换 J_j and J_k 的供应模式
13. Break
14. End if
15. End for
16. End if
17. End if
18. If S' 未被 S 支配 do
19. 在当前种群中保存 S'
20. End if

表 6 算法 5 的伪代码

算法 5. 局部搜索 $LS2_P$
1. 输入: 个体 S
2. 找到延迟成本最大的订单, 记为 J_j
3. J_j 的类型、加工工厂和当前到期日分别记为 r, F_f 和 Id_i
4. 构造一个属于类型 r 且在工厂 F_f 中先于 J_j 加工的订单集合 J_f
5. If $J_f \neq \emptyset$ do
6. For $kk=1: J_f $ do
7. 提取 J_w 中的第 kk 个订单, 定义为 J_k , 并定义它的当前到期日为 Id_k
8. If $Id_j < Id_k$ and $DC_j > DC_k$ do
9. 在交换上述两个订单的加工位置后计算提前/延迟成本的变化, 并将该值保存在集合 TCC 中
10. End if
11. End for
12. 获得 TCC 中的最小值 TCC_{min} 和对应的订单 $J_{j'}$
13. If $TCC_{min} < 0$ do
14. 一个新个体 $S' \leftarrow$ 交换 J_j and $J_{j'}$ 的加工位置
15. End if
16. End if

续上表:

17. **If** S' 未被 S 支配 **do**
18. 在当前种群中保存 S'
19. **End if**

表 7 算法 6 的伪代码

算法 6. 局部搜索 $LS3_P$

1. 输入: 个体 S 和总类型数 R
2. **For** $r=1: R$ **do**
3. 构建一个类型为 r 且由仓库调拨的订单集合 SP
4. **If** $|SP| > 1$ **do**
5. 从 SP 中随机选择一个备件, 定义为 J_j , 并更新 $SP \leftarrow SP \setminus \{J_j\}$
6. 对于 $J_k \in SP$, 交换选定仓库后计算 J_j and J_k 的运输时间变化值 T_{jk}
7. 找到 T_{jk} 中的最小值 $T_{j'}$ 和其对应的订单 $J_{j'}$
8. **If** $T_{j'} < 0$ **do**
9. 更新个体 $S \leftarrow$ 交换 J_j 和 $J_{j'}$ 的初始选定仓库
10. **End if**
11. **End if**
12. **End for**

3.7 维护相关的局部搜索

秉承增强搜索能力和多样性的目的, 本文还针对维护部分设计了 3 个问题导向的搜索算子 $LSO1_M$ 、 $LSO2_M$ 和 $LSO3_M$, 以及随机搜索算子 LSS_M 。这三个特别设计的算子都以目标 ob_2 为指导。搜索算子的详细说明如下。

LSS_M 类似于 LSS_S 。将 LSS_S 中的工厂替换为工人, 即得到了 LSS_M 。局部搜索 $LS1_M$: 通过取消每个工人上不必要的 IM 活动来减少 ob_2 。具体来说就是找到所有具有 IM 活动且备件由仓库生产的装备, 取消其 IM 活动。取消后, ob_2 的组成成分维护成本将有效降低。伪代码在算法 7 中给出。

局部搜索 $LS2_M$: 它通过取消每个工人上的最后一个 IM 活动来起作用。与 $LS1_M$ 类似, $LS2_M$ 通过降低维护成本来达到降低 ob_2 的效果。算法 8 给出了相应的伪代码。

局部搜索 $LS3_M$: 考虑到备件提前交付造成的旧部件剩余价值浪费, 在满足算法 9 中条件的前提下, 我们将具有 IM 活动且备件提前到达时间最长的装备 E_j 和不执行 IM 活动的装备 E_j 进行交换, 从而减少了 E_j 上旧部件残值的浪费以及 E_j 的停机损失, 整体上对 ob_2 起到了有效地减少作用。算法 9 对 $LS3_M$ 进行了简要描述。

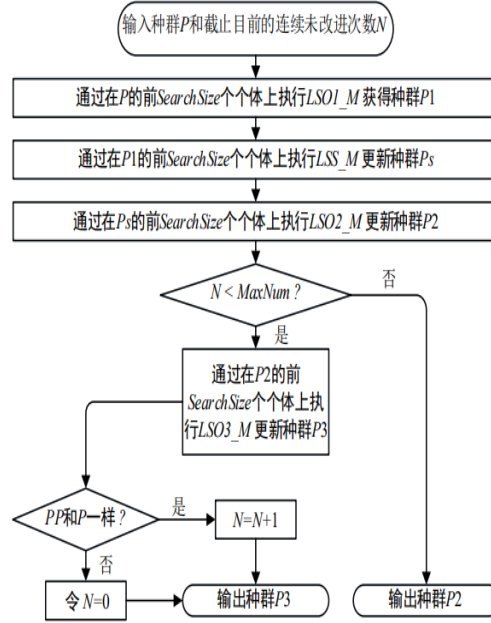


图 9 维护部分局部搜索的流程图

与生产相关的局部搜索类似， $SearchSize$ 数量的个体将被选择执行这些局部搜索算子。不同之处在于，并非本部分中提出的所有算子都在每次迭代中执行，具体的实现过程见图 9。在图 9 中，LS3_M 的相关参数 $MaxNum$ 意为“持续无改进的最大代数”，将在后面进行校准。

表 8 算法 7 的伪代码

算法 7. 局部搜索 $LS1_M$
1. 输入：个体 S
2. 构造一个装备集合 SE 其中每台装备安排了一个不完美维护活动且其备件由仓库调拨而来
3. If $SE \neq \emptyset$ do
4. $S' \leftarrow$ 对每个 $E_j \in SE$ ，删除其上的不完美维护活动
5. End if
6. If S' 支配 S do
7. 将 S 替换为 S'
8. End if

表.9 算法 8 的伪代码

算法 8. 局部搜索 $LS2_M$
1. 输入：个体 S 和工人集合 SW
2. $S' \leftarrow$ 复制 S ; 令 $flag=0$
3. For $k=1: SW $ do
4. 取消工人 $W_k \in SW$ 的最后一个不完美维护活动后计算目标值 ob_2 的变化 ΔC
5. If $\Delta C < 0$ do
6. 通过取消工人 W_k 的最后一个不完美维护来更新 S'

续上表:

7. $\text{flag} \leftarrow \text{flag}+1$
8. **End if**
9. **End for**
10. **If** $\text{flag} > 0$ 且 S' 未被 S 支配 **do**
11. 在当前种群中保留 S'
12. **End if**

表 10 算法 9 的伪代码

算法 9. Local search $LS3_M$

1. 输入: 个体 S 和工人集合 SW
2. $S' \leftarrow$ 复制 S ; 令 $\text{flag}=0$
3. 构造一个装备集合 EN , 其中每台装备都没安排不完美维护活动且其备件由工厂加工而来
4. **For** $k=1: |SW|$ **do**
5. 找到基于工人 $W_k \in SW$ 执行不完美维护后的交货期, 备件提前到达时间最长的装备 E_i
6. 从 EN 中选择那些和 E_i 的备件类型相同的装备组成集合 SEN
7. **For** $kk=1: |SEN|$ **do**
8. 找到 SEN 中第 kk 个装备, 定义为 E_j
9. **If** $d_j < d_i$ and $Rd_j > Rd_i$ and $WC_j < WC_i$ and $PC_j < PC_i$ **do**
10. $S' \leftarrow$ 删除 E_i 的不完美维护活动且在工人 W_k 上将其原本的维护位置安排给 E_j
11. **Break**
12. **End if**
13. **End for**
14. **End for**
15. **If** S' 支配 S **do**
16. 将 S 替换为 S'
17. **End if**

4. 实验分析

4.1 算例构造

为了证明所提算法对 PIM-DCSP 的有效性, 参考[109]构造了 96 个算例, 缩写为 PIM 01-96。每个算例被命名为 $n \times F \times m \times R \times P \times Ws$ 。96 个算例被分成三组, 每组代表一种规模。PIM 01-32 为小规模, 中等规模包括 PIM 33-64, 剩下的则属于大规模。各算例的具体构造方法如表 12 所示。

表 11 算例构造

参数	范围	构造方法
n	{100, 300, 500}	随机选择
F	{3, 5}	随机选择
m	{5, 8}	随机选择
R	{5, 15}	随机选择
ws	{3, 4}	随机选择
P	{3, 4}	随机选择
P_{rk}	[10, 30]	随机 (均匀分布的整数)
TM_e	[30, 50]	随机 (均匀分布的整数)
MC_e	[3, 5]	满足第 2.1 节中的关系 1 (均匀分布且保留一位小数)
SC_r	[1, 3]	随机 (均匀分布且保留一位小数)
TC	3	--
DC_j	[3, 6]	随机 (均匀分布且保留一位小数)
WC_j	[5, 8]	随机 (均匀分布且保留一位小数)
PC_j	[20, 25]	随机 (均匀分布且保留一位小数)
v_j	[1, 2]	随机 (均匀分布且保留两位小数)
LR_s	[0.5, 0.7]	随机 (均匀分布且保留两位小数)
TF_{fj}	[30, 150]	随机 (均匀分布的整数)
TR_{ij}	[30, 80]	随机 (均匀分布的整数)
d_j	--	参考公式 (29)
$[t_j^-, t_j^+]$	$[0.7d_j, 0.9d_j]$	--

对于每台装备, 其健康值 HV_j 是通过 v_j 和 d_j 获取的, 即 $HV_j = v_j \times d_j$ 。初始理想交货时间 d_j 由式 (29) 得到, 其中 C_{max} 为所有订单的最大加工完成时间, 根据 LPT 启发式规则计算而得。至于仓库库存量 N_{fr} , 只需要满足备品备件总数量在 $0.15n \sim 0.25n$ 范围内, 每个仓库的具体备件种类和数量是随机确定的。

$$d_j = \max \left\{ \max_{f \in SF, j \in N} TF_{fj}, U[0.1C_{\max}, 0.9C_{\max}] \right\} \quad (29)$$

4.2 参数校准

为了增强最佳参数组合确定的公正性, 采用了一种称为“全因子设计实验”的统计设计和实验分析方法来校准本文涉及的所有算法, 以确保算法的高性能和

公平比较。以上所有算法均由 MATLAB R2016a 编写，并在 3.60GHz 和 32GB RAM 的 Inter-Core i9-9900k CPU 上进行计算。为了避免单个算例实验结果的偶然性，从第 4.1 节中构建的 96 个测试算例中随机选择 15 个算例进行全因子实验。每次运行均将 0.5n 秒的最大 CPU 时间设置为停止标准。此外，所有实验结果都利用反向代际距离（IGD）进行评估。注：IGD 值越小越好。

4.2.1 IACA 的参数校准

对于本文首次提出的 IACA，需要校准的主要参数有 5 个，分别是 Popsiz、Pc、Pm、SearchSize 和 MaxNum。如表 12 所示，每个参数由四个水平组成，总共产生 $4 \times 4 \times 4 \times 4 \times 4 = 1024$ 个参数配置组合。此外，对于每个算例，每个配置组合重复且独立地执行 5 次，因而将一共产生 $1024 \times 15 \times 5 = 76800$ 个 IGD 计算结果并利用 Minitab 17 上的多元方差分析（MANOVA）对实验结果进行了分析。图 10 显示了每个参数对应的主效应图，直观地反映了每个参数水平对算法性能的影响程度。从 10 中，我们可以很容易地得出结论，IACA 在 Popsiz=25、Pc=0.9、Pm=0.1、SearchSize=25 和 MaxNum=15 时通常表现良好。因此，在后续实验中，参数将固定为上述值。

表 12 各参数各水平值

参数	Popsiz	p _c	p _m	SearchSize	MaxNum
水平 1	25	0.7	0.1	10	5
水平 2	50	0.8	0.15	15	10
水平 3	75	0.9	0.2	20	15
水平 4	100	1	0.25	25	20

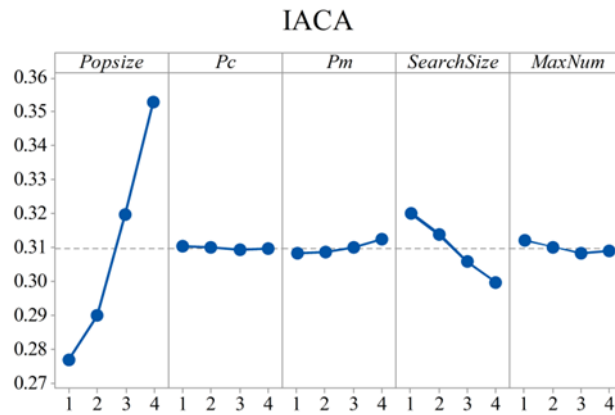


图 10 IACA 各参数水平下的效应图

4.2.2 对比算法的参数设定

为了更好地说明 IACA 在解决 PIM-DCSP 上的出色表现，选择了三种著名算法在相同的时间和算例下与 IACA 进行比较，即非支配排序遗传算法(NSGA-II)、

非支配邻居免疫算法 (NNIA) 和多目标粒子群优化算法 (MOPSO)，这三个算法在解决多目标问题方面的有效性在许多当前文献中得到了证明。为了确保对比效果的可靠性，对比较算法也进行了参数校准。具体校准方法参见第 4.1 节。与 IACA 的参数设置类似，每个参数也有四个水平。对于 NSGA-II，它包括三个参数 $Popsiz$ 、 P_c 和 P_m ，它们的水平与 IACA 相同，因为 IACA 是在 NSGA-II 的基础上形成的。在 MOPSO 中有两个参数 ($Popsiz$ 和 P_m)，两个参数的四个水平设计为 $Popsiz$: [30, 40, 50, 60] 和 P_m : [0.1, 0.2, 0.3, 0.4]。对于 NNIA，它有四个参数 (NA 、 NC 、 P_c 和 P_m ，其中 NA 和 NC 分别代表活跃种群和支配种群的最大规模)，其中 P_c 和 P_m 的四个水平与 IACA 一致，另外两个参数设计为 NA : [20, 30, 40, 50]， NC : [70, 80, 90, 100]。上述三种算法的每个参数对应的主效应图分别如图 11 (a~c) 所示。最后，得出结论，所有比较算法的参数设置为 NSGA-II: $Popsiz=100$, $P_c=1$, $P_m=0.25$; MOPSO: $Popsiz=60$, $P_m=0.4$; NNIA: $NA=50$, $NC=70$, $P_c=1$, $P_m=0.25$ 。

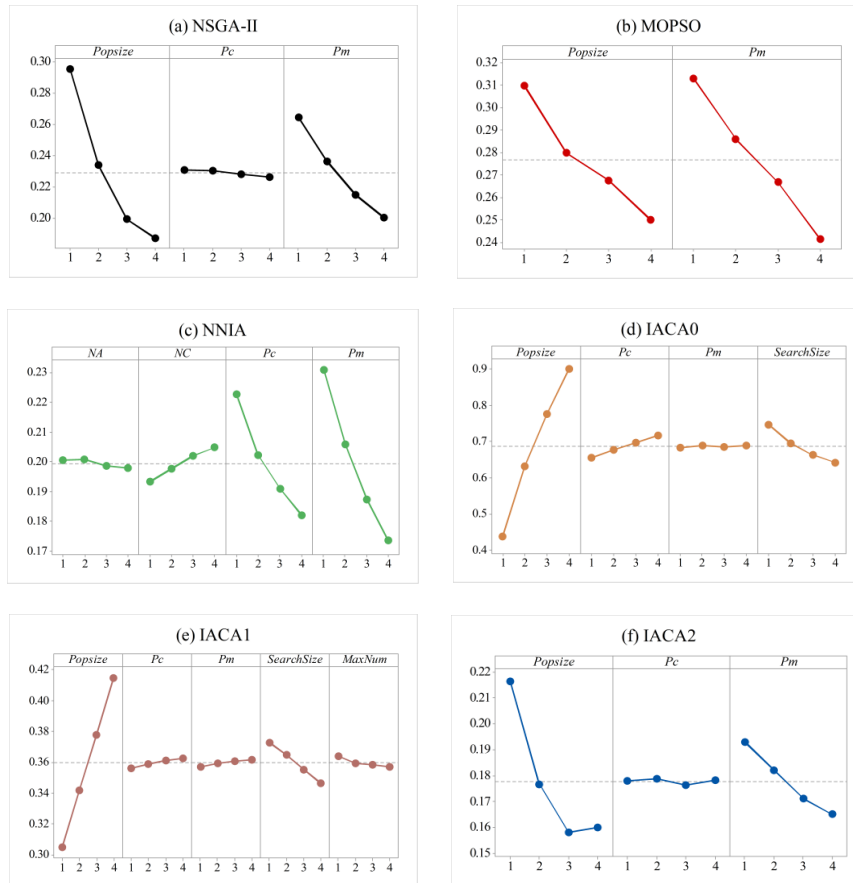


图 11 对比算法各参数水平影响效应图

4.3 PIM-DCSP 模型的有效性

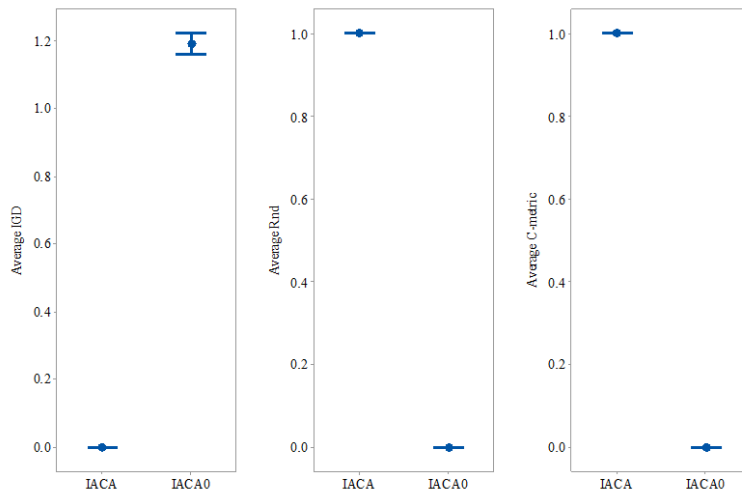
毫无疑问，将维护调度添加到算法中非常耗时。之所以在本文中仍被考虑是

因为备件交付前装备上的 IM 活动对降低供需双方的成本起着重要作用。因此，为了证明考虑维护的重要性，我们做了大量有维护和无维护的对比实验。删除维护部分的算法称为 IACA0。由于维护的取消对 IACA0 的结构有很大的影响，所以 IACA0 的参数需要重新设置，具体的方法和过程与 4.1 节相同，所以这里不再重复。设置结果是 $Popsiz=25$, $Pc=0.8$, $Pm=0.2$, $SearchSize=25$ 。

实验结果表明，就 IGD、 R_{nd} 和 C 而言，纵观全部算例，IACA 均绝对优于 IACA0。这个结论来自于在所有算例中，IACA 的 IGD 均等于 0， R_{nd} 和 C 均等于 1。由于空间限制，本节未列出每个评估指标的实验数据结果。为了更好地反映各算法的比较结果之间的差异，采用了 Wilcoxon 符号秩检验进行辅助分析，结果列于表 13，可以得到 IACA 和 IACA0 的性能具有显著性差异的结论。图 12 为 IACA 和 IACA0 关于 IGD、 R_{nd} 和 C 的 95%置信区间的统计结果，直观清晰地显示了 IACA 与 IACA0 的显著差异。因此，有必要将维护考虑在内。

表 13 Wilcoxon 符号秩检验结果

	P 值 (IGD)	P 值 (R_{nd})	P 值 (C-metric)
(IACA, IACA0)	1.78E-17	1.15E-22	1.15E-22
(IACA, IACA1)	3.01E-06	2.14E-08	1.86E-10
(IACA, IACA2)	1.61E-16	2.09E-21	1.80E-21
(IACA, NSGA-II)	1.78E-17	1.86E-22	1.86E-22
(IACA, NNIA)	1.78E-17	1.15E-22	1.15E-22
(IACA, MOPSO)	1.78E-17	1.15E-22	1.15E-22


 图 12 IACA 和 IACA0 平均 IGD、 R_{nd} 和 C 的 95%置信区间

4.4 IACA 算法特殊策略的有效性

在我们的算法中，初始化方法 (LSL 和 FSL) 和局部搜索是两种特殊的策略，本节将对其有效性进行检验。在实验之前，生成了两种算法：一种是 IACA1，

没有 IACA 上的初始化方法；另一种是 IACA2，去掉 IACA 上所有的生产部分和维护部分的局部搜索算子。为了确保实验的公平性，还进行了第 4.1 节中介绍的参数校准实验。从图 13(e~f) 观察，确定了上述两个算法的最佳参数组合：IACA1 的为 $Popsize=25$, $Pc=0.7$, $Pm=0.1$, $SearchSize=25$ 和 $MaxNum=20$ ，IACA2 的为 $Popsize=75$, $Pc=0.9$, $Pm=0.25$ 。在最佳参数水平下，对所有算例进行了比较实验。图 13 给出了包含所有三个指标的统计结果。观察图 13 所示的比较结果可以看出，与 IACA2 相比，无论是哪个指标，IACA 的性能都明显优于 IACA2。特别是从 C 的角度来看，在大多数情况下，IACA 产生的前沿个体可以完全支配 IACA2 获得的前沿个体。对于 IACA1，虽然对比效果不如 IACA2 显著，但也足以显示出和 IACA1 比较，IACA 具有更好的表现。如第 3.4.2 节所述，上述算法之间是否存在显著差异也通过 Wilcoxon 符号秩检验进行测试。测试结果位于表 13 的第 2 行和第 3 行。显然，再次验证了它们之间的差异是显著的。总之，我们可以得出结论，初始化方法和局部搜索对于 IACA 在有限的时间内找到高质量的解非常有帮助。

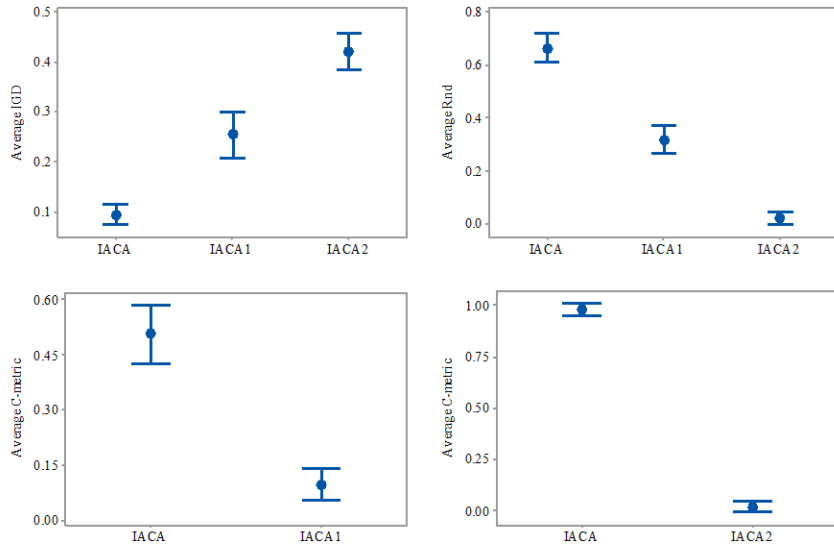


图 13 IACA, IACA1 和 IACA2 平均 IGD、 R_{nd} 和 C 的 95% 置信区间

4.5 和其他算法比较

为了体现 IACA 在处理 PIM-DCSP 方面的强大优势，本部分列出并分析了 IACA、NSGA-II、NNIA 和 MOPSO 四种算法的计算结果。同时得到了 IGD、 R_{nd} 和 C 三个评价指标的比较结果。对于 R_{nd} 和 C，IACA 的值为 1，其他三种算法的值几乎在所有实例上都为 0。因此，为了使本文更简洁，这里仅在表 14 中给出 IGD 的结果。为了直观地体现所设计的算法（IACA）相对于所有其他算法的优越性，我们选择了四个不同规模的算例来绘制帕累托前沿图，如图 3.14 所示。如表 14 所示，IACA 的 IGD 在所有实例上都比其他三种算法的 IGD 小很多。更

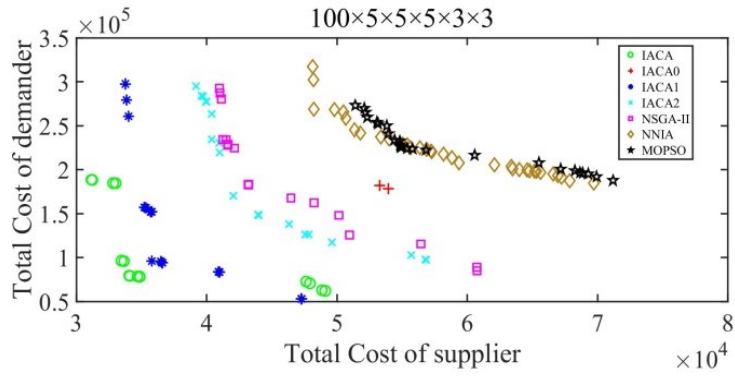
具体地说, 对于 IACA 来说, 除了 PIM 09、PIM 16、PIM 20 和 PIM 33 之外, 它的 IGD 在其余所有算例上都等于零。对于 Rnd 和 C, 我们可以发现 NNIA 和 MOPS 的前沿个体在所有情况下都完全由 IACA 获得的前沿个体支配, 而 NSGA-II 的前沿个体在绝大多数情况下完全由 IACA 获得的前沿个体支配。通过分析, 我们得出了 IACA 的性能优于 NSGA-II、NNIA 和 MOPSO 的结论。此外, 基于表 14 中的最后三行, IACA 与其竞争算法之间的显著差异是明显的, 这种对比效果在图 14 中很好地呈现了。

表 14 IACA、NSGA-II、NNIA 和 MOPSO 的 IGD 对比结果

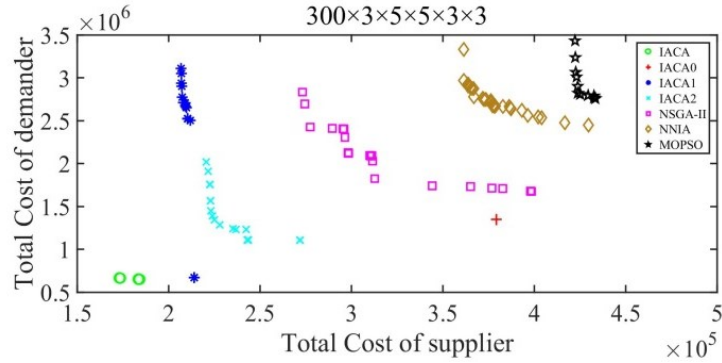
算例	IACA	NSGAII	NNIA	MOPSO	算例	IACA	NSGAII	NNIA	MOPSO
PIM 01	0.0000	0.4082	0.6709	0.7697	PIM 49	0.0000	0.5601	1.1355	0.8692
PIM 02	0.0000	0.2930	0.6659	0.6061	PIM 50	0.0000	0.5962	1.1856	0.9209
PIM 03	0.0000	0.4009	0.7381	0.7047	PIM 51	0.0000	0.5571	1.0950	1.0171
PIM 04	0.0250	0.2017	0.5796	0.6795	PIM 52	0.0000	0.6705	1.0324	1.0474
PIM 05	0.0000	0.1969	0.5659	0.5475	PIM 53	0.0000	0.5734	1.1854	0.8204
PIM 06	0.0000	0.2741	0.5958	0.5511	PIM 54	0.0000	0.5699	1.1004	0.8521
PIM 07	0.0000	0.3940	0.7291	0.8326	PIM 55	0.0000	0.6379	1.1682	1.0530
PIM 08	0.0000	0.1742	0.5669	0.5936	PIM 56	0.0000	0.5696	0.8714	0.8417
PIM 09	0.0000	0.2769	0.5767	0.5560	PIM 57	0.0000	0.6115	1.1266	0.9361
PIM 10	0.0000	0.4652	0.8268	0.8523	PIM 58	0.0000	0.6244	1.0347	0.9116
PIM 11	0.0000	0.3664	0.6877	0.7578	PIM 59	0.0000	0.7189	1.1221	0.9633
PIM 12	0.0000	0.4272	0.7906	0.8070	PIM 60	0.0000	0.5781	1.0165	1.0081
PIM 13	0.0000	0.2069	0.5938	0.4655	PIM 61	0.0000	0.7284	1.3349	1.0011
PIM 14	0.0000	0.2133	0.6534	0.6350	PIM 62	0.0000	0.3783	0.8799	0.8219
PIM 15	0.0000	0.1965	0.5135	0.5337	PIM 63	0.0000	0.6145	1.1425	0.9533
PIM 16	0.0000	0.2131	0.4759	0.5062	PIM 64	0.0000	0.4768	0.9461	0.8488
PIM 17	0.0000	0.3438	0.6810	0.6569	PIM 65	0.0000	0.5492	1.0609	0.8937
PIM 18	0.0000	0.2869	0.6033	0.5927	PIM 66	0.0000	0.4678	0.9508	0.6889
PIM 19	0.0000	0.3745	0.6737	0.7187	PIM 67	0.0000	0.6530	1.0880	0.9575
PIM 20	0.0000	0.3388	0.6236	0.5562	PIM 68	0.0000	0.4728	0.9425	0.7728
PIM 21	0.0000	0.2156	0.5909	0.6136	PIM 69	0.0000	0.5377	1.0277	0.8663
PIM 22	0.0000	0.2510	0.6755	0.6449	PIM 70	0.0000	0.4822	1.0683	0.8421
PIM 23	0.0000	0.3891	0.7089	0.6826	PIM 71	0.0000	0.5384	1.0477	0.7969
PIM 24	0.0000	0.2914	0.5782	0.6113	PIM 72	0.0000	0.5464	1.0331	0.8374
PIM 25	0.0000	0.5357	0.7459	0.8026	PIM 73	0.0000	0.5870	1.1339	0.4376
PIM 26	0.0000	0.4525	0.7102	0.7457	PIM 74	0.0000	0.6765	1.1581	0.6111
PIM 27	0.0000	0.3527	0.6534	0.6590	PIM 75	0.0000	0.6829	1.1322	0.5847
PIM 28	0.0000	0.6463	0.8841	0.8465	PIM 76	0.0000	0.6252	1.0252	0.6377
PIM 29	0.0000	0.2470	0.5867	0.6317	PIM 77	0.0000	0.6048	1.1218	0.5637
PIM 30	0.0000	0.2732	0.6078	0.4996	PIM 78	0.0000	0.5506	1.0087	0.4702

续上表:

算例	IACA	NSGAII	NNIA	MOPSO	算例	IACA	NSGAII	NNIA	MOPSO
PIM 31	0.0000	0.3076	0.5952	0.5806	PIM 79	0.0000	0.6603	1.1123	0.4939
PIM 32	0.0000	0.3328	0.6545	0.6002	PIM 80	0.0000	0.5830	0.9293	0.9220
PIM 33	0.0000	0.6024	1.1082	0.9671	PIM 81	0.0000	0.5078	1.1980	0.6996
PIM 34	0.0000	0.5517	0.9536	0.8616	PIM 82	0.0000	0.7202	1.2908	1.1383
PIM 35	0.0000	0.5327	1.0524	0.9582	PIM 83	0.0000	0.7160	1.1580	0.8380
PIM 36	0.0000	0.5495	1.0751	1.0030	PIM 84	0.0000	0.6788	1.1287	0.9560
PIM 37	0.0000	0.6004	1.0900	0.9383	PIM 85	0.0000	0.6063	1.2621	0.8307
PIM 38	0.0000	0.3990	0.9723	0.7006	PIM 86	0.0000	0.4789	1.0618	0.9570
PIM 39	0.0000	0.4533	1.0623	0.7783	PIM 87	0.0000	0.5705	1.0931	0.8970
PIM 40	0.0000	0.3282	0.7533	0.6801	PIM 88	0.0000	0.5917	1.0534	0.9781
PIM 41	0.0000	0.6821	1.1594	1.0075	PIM 89	0.0000	0.6029	1.0996	1.2632
PIM 42	0.0000	0.4298	0.9452	0.8488	PIM 90	0.0000	0.5043	0.9199	1.2878
PIM 43	0.0000	0.5725	0.9123	0.8422	PIM 91	0.0000	0.5548	0.9389	1.2360
PIM 44	0.0000	0.5031	0.9109	0.7982	PIM 92	0.0000	0.4878	0.8121	1.1484
PIM 45	0.0000	0.6144	1.1421	1.0169	PIM 93	0.0000	0.5176	0.9319	1.0150
PIM 46	0.0000	0.4793	0.9113	0.7958	PIM 94	0.0000	0.4218	0.7334	1.1501
PIM 47	0.0000	0.4231	0.8685	0.7570	PIM 95	0.0000	0.5506	0.9215	1.2617
PIM 48	0.0000	0.5371	0.9534	0.8655	PIM 96	0.0000	0.4775	0.7357	1.1654



(a) PIM-17



(b) PIM-33

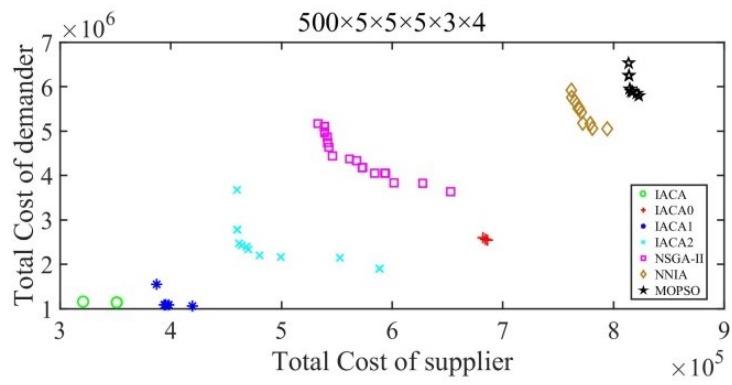
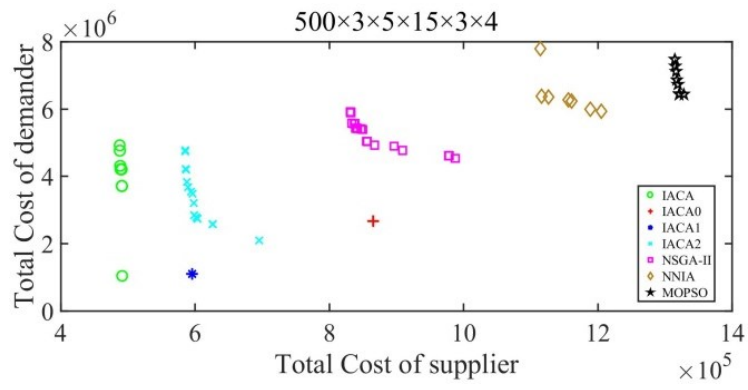


图 14 四种选择算例的帕累托前沿图

5. 结论

为了弥补现有供需信息互馈研究的不足,本章研究了供给侧和需求侧相互反馈的生产、库存和不完全维护一体化的最优调度问题,建立了以供需双方总成本分别最小为优化目标的数学模型。为了有效处理 PIM-DCSP,设计了一种具有有效启发式初始化方法和局部搜索算子的协作算法,验证了启发式初始化方法和局部搜索算子在 IACA 中的优势,以及增加维护以降低成本的重要意义。最后,通过大量的实验,与其他三种著名算法进行比较,充分证明了 IACA 算法的显著性能。

本研究工作的实践和理论价值体现在:(1)通过信息交换的方法,为供需双方提供了一种双赢的合作机制。该机制具有双向指导作用,具体体现为:a)对于需求方而言,在收到生产计划后,可以指导其对装备进行预防性维护管理,从而适度延缓装备故障的发生,从而减少工程损失;b)对于生产商而言,需求方反馈的延迟的备件交货期可以缓解产压力,实现生产资源和库存的协同优化,制定客户满意的生产调度计划,从有效增强其市场竞争力。(2)提出了一种新的跨领域调度研究,它打破了传统的以生产为中心的究层次,将研究维度扩展到制造和运维集成,为跨实体业务协作的研究奠定了基础。

参考文献

- [1] 国务院. 国务院关于积极推进“互联网+”行动的指导意见. 中华人民共和国国务院公报, 2015, (20): 11-23.
- [2] 国务院. 国务院关于印发《中国制造 2025》的通知. 中华人民共和国国务院公报, 2015, (16): 10-26.
- [3] 刘献礼, 李雪冰, 丁明娜, 岳彩旭, 王力羣, 梁月昇, 张博闻. 面向智能制造的刀具全生命周期智能管控技术. 机械工程学报, 2021, 57(10): 196-219.
- [4] 汪俊亮, 高鹏捷, 张洁, 王力羣. 制造大数据分析综述: 内涵、方法、应用和趋势. 机械工程学报, 2023, 59(12): 1-16.
- [5] 蔡静雯, 马玉敏, 黎声益, 刘鹃. 基于 Q 学习的智能车间自适应调度方法. 计算机集成制造系统, 2023, 29(11): 3727-3737.
- [6] Zhang B H, Che A, Wang Y S. Grid-based artificial bee colony algorithm for multi-objective job shop scheduling with manual loading and unloading tasks. Expert Systems with Applications, 2024, 245: 123011.
- [7] 梁望, 钱斌, 胡蓉, 张梓琪, 张长胜. 两阶段智能优化算法求解紧凑型带钢生产热轧调度问题. 计算机集成制造系统, 2023. 1-21. (网络首发时间: 2023-08-02)
<https://kns.cnki.net/kcms2022/detail/2011.5946.TP.20230801.20231648.20230007.html>.
- [8] Huang J P, Gao L, Li X Y, Zhang C J. A novel priority dispatch rule generation method based on graph neural network and reinforcement learning for distributed job-shop scheduling. Journal of Manufacturing Systems, 2023, 69: 119-134.
- [9] Soares L C R, Carvalho M a M. Biased random-key genetic algorithm for the job sequencing and tool switching problem with non-identical parallel machines. Computers & Operations Research, 2024, 163: 106509.
- [10] 胡一凡, 张利平, 白雪, 唐秋华. 深度强化学习求解柔性装配作业车间调度问题. 华中科技大学学报(自然科学版), 2023, 51(02): 153-160.
- [11] Lian X Y, Zheng Z, Zhu M M, Gao X Q. Proactive scheduling for steel plants with unrelated parallel machines and time uncertainty. Computers & Industrial Engineering, 2024, 188: 109890.
- [12] Pan Y X, Gao K Z, Li Z W, Wu N Q. Solving Biobjective Distributed Flow-Shop Scheduling Problems With Lot-Streaming Using an Improved Jaya Algorithm. IEEE Transactions on Cybernetics, 2023, 53(6): 3818-3828.
- [13] Zhao F Q, Di S L, Wang L. A Hyperheuristic With Q-Learning for the Multiobjective Energy-Efficient Distributed Blocking Flow Shop Scheduling Problem. IEEE Transactions on Cybernetics, 2023, 53(5): 3337-3350.
- [14] Wu X Q, Yan X F, Guan D H, Wei M Q. A deep reinforcement learning model for dynamic job-shop scheduling problem with uncertain processing time. Engineering Applications of Artificial Intelligence, 2024, 131: 107790.
- [15] Meng L L, Duan P, Gao K Z, Zhang B, Zou W Q, Han Y Y, Zhang C Y. MIP modeling of energy-conscious FJSP and its extended problems: From simplicity to complexity. Expert Systems with Applications, 2024, 241: 122594.
- [16] Pan Z X, Wang L, Zheng J, Chen J F, Wang X. A Learning-Based Multipopulation

- Evolutionary Optimization for Flexible Job Shop Scheduling Problem With Finite Transportation Resources. *IEEE Transactions on Evolutionary Computation*, 2023, 27(6): 1590-1603.
- [17] Li D, Ruan X, Yue Q. Optimization study of three-stage assembly flowshop problem in pharmacy automation dispensing systems. *Computers & Operations Research*, 2022, 144: 105810.
- [18] Liu Q H, Wang C Y, Li X Y, Gao L. An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task. *Journal of Manufacturing Systems*, 2023, 70: 127-136.
- [19] Chen S Y, Wang X W, Wang Y, Gu X S. A modified adaptive switching-based many-objective evolutionary algorithm for distributed heterogeneous flowshop scheduling with lot-streaming. *Swarm and Evolutionary Computation*, 2023, 81: 101353.
- [20] Li H T, Liu J H, Wang Y, Zhuang C B. A multi-objective complex product assembly scheduling problem considering transport time and worker competencies. *Advanced Engineering Informatics*, 2023, 58: 102233.
- [21] Yu F, Lu C, Zhou J J, Yin L. Mathematical model and knowledge-based iterated greedy algorithm for distributed assembly hybrid flow shop scheduling problem with dual-resource constraints. *Expert Systems with Applications*, 2024, 239: 122434.
- [22] Li R, Gong W Y, Wang L, Lu C, Dong C X. Co-Evolution With Deep Reinforcement Learning for Energy-Aware Distributed Heterogeneous Flexible Job Shop Scheduling. *IEEE Transactions on Systems Man Cybernetics-Systems*, 2023, 54(1): 201-211.
- [23] Zhang L, Feng Y, Xiao Q E, Xu Y L, Li D, Yang D S, Yang Z L. Deep reinforcement learning for dynamic flexible job shop scheduling problem considering variable processing times. *Journal of Manufacturing Systems*, 2023, 71: 257-273.
- [24] Zhao F, Ma R, Wang L. A Self-Learning Discrete Jaya Algorithm for Multiobjective Energy-Efficient Distributed No-Idle Flow-Shop Scheduling Problem in Heterogeneous Factory System. *IEEE Transactions on Cybernetics*, 2022, 52(12): 12675-12686.
- [25] 蔡磊, 李文锋, 罗云. 个性化定制车间生产-物流协同调度框架与算法研究. *机械工程学报*, 2022, 58(07): 214-226.
- [26] Wang H, Yan Q, Wang J. Blockchain-secured multi-factory production with collaborative maintenance using Q learning-based optimisation approach. *International Journal of Production Research*, 2023, 61(11): 3685-3702.
- [27] Cai L, Guo W J, He L J, Li W F. Port integrated scheduling under uncertain operation time and cascade effects: A complex network structure entropy solution. *Computers & Industrial Engineering*, 2023, 182: 109435.
- [28] Huang R J, Ma Y S, Li H, Sun C Y, Liu J, Zhang S, Wang H Q, Hao B. Operation parameters multi-objective optimization method of large vertical mill based on CFD-DPM. *Advanced Powder Technology*, 2023, 34(6): 104014.
- [29] Xia X X, Liu Z P, Wang Z Q, Sun T, Zhang H L. Multi-layer performance optimization based on operation parameter-working fluid-heat source for the ORC-VCR system. *Energy*, 2023, 272: 127103.

- [30] 姜维达, 秦国华, 王华敏, 吴竹溪. 基于复化 Cotes 积分和神经网络的稳定铣削工艺参数多目标优化方法. 机械工程学报, 2023, 59(17): 279-290.
- [31] Guo C H, Liang Z L. A predictive Markov decision process for optimizing inspection and maintenance strategies of partially observable multi-state systems. Reliability Engineering & System Safety, 2022, 226: 108683.
- [32] 葛亚雄, 周晓军. 考虑修复非新的多组件 DTS 设备的状态维护建模. 计算机集成制造系统, 2021, 27(03): 739-746.
- [33] Han X, Wang Z L, Xie M, He Y H, Li Y, Wang W Z. Remaining useful life prediction and predictive maintenance strategies for multi-state manufacturing systems considering functional dependence. Reliability Engineering & System Safety, 2021, 210: 107560.
- [34] Babai M Z, Chen H, Syntetos A A, Lengu D. A compound-Poisson Bayesian approach for spare parts inventory forecasting. International Journal of Production Economics, 2021, 232: 107954.
- [35] Lv L L, Shen W M. An improved NSGA-II with local search for multi-objective integrated production and inventory scheduling problem. Journal of Manufacturing Systems, 2023, 68: 99-116.
- [36] Li H, Mi S H, Li Q F, Wen X Y, Qiao D P, Luo G F. A scheduling optimization method for maintenance, repair and operations service resources of complex products. Journal of Intelligent Manufacturing, 2020, 31(7): 1673-1691.
- [37] Rahimi-Ghahroodi S, Al Hanbali A, Zijm W H M, Timmer J B. Multi-resource emergency supply contracts with asymmetric information in the after-sales services. International Journal of Production Economics, 2020, 229: 107761.
- [38] Rahimi-Ghahroodi S, Al Hanbali A, Vliegen I M H, Cohen M A. Joint optimization of spare parts inventory and service engineers staffing with full backlogging. International Journal of Production Economics, 2019, 212: 39-50.
- [39] Turan H H, Atmis M, Kosanoglu F, Elsayah S, Ryan M J. A risk-averse simulation-based approach for a joint optimization of workforce capacity, spare part stocks and scheduling priorities in maintenance planning. Reliability Engineering & System Safety, 2020, 204: 107199.