混合三维分布估计算法求解分布式加工装配和车辆配送集成调度问题

杨绍文[1,2],钱 斌[1,2*],胡 蓉[1,2],张梓琪[1,2]
(1. 昆明理工大学信息工程与自动化学院,云南昆明 650500;2. 云南省人工智能重点实验室,云南昆明 650500)

摘 要: 本文针对一类广泛存在的分布式加工装配和车辆配送集成调度问题(Integrated Scheduling Problem of Distributed Production Assembly and Vehicle Delivery,ISP_DPAVD),以最小化运输和延迟惩罚总成本为优化目标,提出一种混合三维分布估计算法(Hybrid three-Dimensional Estimation of Distribution Algorithm,H3DEDA)进行求解. ISP_DPAVD包含两个耦合的子问题,即加工装配阶段子问题(子问题1)和车辆配送阶段子问题(子问题2). 由于每个子问题1的解(部分解1)均会确定1个具体的子问题2,故ISP_DPAVD的解空间非常庞大. 根据这一特点,在H3DEDA中,先设计结合邻域变换的启发式规则来快速获取子问题2的优良解,以实现子问题间的部分解耦并明显缩减搜索空间,再设计三维EDA引导的全局搜索和变邻域驱动的局部搜索来获取ISP_DPAVD的高质量解. 通过在不同规模测试问题上的仿真实验和算法比较,验证了H3DEDA求解ISP_DPAVD的有效性.

关键词: 分布式加工装配流水车间;车辆配送;集成调度;三维分布估计算法;变邻域搜索
基金项目: 国家自然科学基金(No.62173169,No.61963022);云南省基础研究重点项目(No.202201AS070030)
中图分类号: TP273 **文献标识码:** A **文章编号:** 0372-2112(2024)03-0909-15
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220908

Hybrid Three-Dimensional Estimation of Distribution Algorithm for Integrated Scheduling Problem of Distributed Production Assembly and Vehicle Delivery

YANG Shao-wen[1,2], QIAN Bin[1,2*], HU Rong[1,2], ZHANG Zi-qi[1,2]
(1. *Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan* 650500, *China*;
2. *Yunnan Key Laboratory of Artificial Intelligence, Kunming, Yunnan* 650500, *China*)

Abstract: This paper proposes a hybrid three-dimensional distribution estimation algorithm (H3DEDA) to minimize the total cost of transportation and delay penalties for a kind of widely existed integrated scheduling problem (ISP_DPAVD), which includes two coupled subproblems, i.e., the subproblem in the production and assembly stage (subproblem 1) and the subproblem in the vehicle distribution stage (subproblem 2). Since each solution of subproblem 1 determines a specific subproblem 2, the solution space of the ISP_DPAVD is very large. According to this characteristic, in H3DEDA (Hybrid three-Dimensional Estimation of Distribution Algorithm), the heuristic rules combined with neighborhood transformation are designed to quickly obtain the excellent solution of subproblem 2, so as to achieve partial decoupling between subproblems and significantly reduce the search space. Then, the global search guided by three-dimensional EDA and the local search driven by variable neighborhood operations are devised to acquire high-quality solution for the ISP_DPAVD. Simulation experiments and comparisons on the test problems with different scales verify the effectiveness of H3DEDA in solving ISP_DPAVD.

Key words: distributed production and assembly flow shop; vehicle delivery; integrated scheduling; three-dimensional estimation of distribution algorithm; variable neighborhood search

Foundation Item(s): National Natural Science Foundation of China (No.62173169, No.61963022); Basic Research Key Project of Yunnan Province (No.202201AS070030)

收稿日期:2022-08-01;修回日期:2023-02-18;责任编辑:梅志强
*通讯作者:钱斌

1 引言

在经济全球化背景下,装配制造企业的生产模式逐渐从集中式向分布式转变,分布式加工装配模式能够有效降低生产风险、提高资源利用率[1]. 与此同时,在市场竞争加剧、客户需求多样等诸多因素影响下,企业为抢占市场、提高客户满意度,纷纷构建“工厂直发,送货到家”的产品配送模式,装配制造企业由传统的“加工-装配”式向新型的“加工-装配-配送”式转变. 如何在短时间完成产品加工、装配的同时,以较低的配送成本和延迟成本将其交付至客户成为当前新型装配制造企业发展的关键. 在上述背景下,分布式加工装配和车辆配送集成调度问题(Integrated Scheduling Problem of Distributed Production Assembly and Vehicle Delivery, ISP_DPAVD)的重要性日益凸显. 在计算复杂度上,分布式两阶段加工装配流水车间调度问题(Distributed Two stage Assembly Flow Shop scheduling Problem, DTAFSP)已被证明为非确定性多项式(Non-deterministic Polynomial Hard, NP-Hard)问题[1~5],而该问题又归约为ISP_DPAVD,故ISP_DPAVD亦为NP_hard问题. 因此,研究ISP_DPAVD建模和求解具有重要的实际意义和学术价值.

近年来,分布式生产和车辆配送的集成调度问题开始受到学术界的关注. 针对分布式单机车间和车辆配送集成调度问题,Gharaei等[6]以最小化总配送成本和总延迟时间为优化目标,提出一种带蜜蜂算法分解策略的多目标算法进行求解;Li等[7]以提高客户服务水平和降低运输成本为优化目标,采用动态规划算法进行求解;Yılmaz等[8]以最小化提前交货惩罚成本、延迟交货惩罚成本和运输成本之和为优化目标,提出一种结合模拟退火算法的改进人工蜂群算法(Hybrid Artificial Bee Colony and Simulated Annealing algorithm, HABCSA)进行求解. 针对分布式并行机车间和车辆配送集成问题,Hao等[9]以最大化每个工厂的权重利润和为优化目标,采用CPLEX进行求解;Abdollahzadeh等[10]以最小化生产成本、配送成本和延迟交货成本之和为优化目标,提出一种改进鲸鱼优化算法(Improved Whale Optimization Algorithm, IWOA)进行求解. 针对分布式混合流水车间和车辆配送集成问题,Fateme等[11]以最小化延迟成本和配送成本之和为优化目标,提出一种改进帝国竞争算法(Improved Imperialist Competitive Algorithm, IICA)进行求解;Qin等[12]以最小化提前交付惩罚成本、延迟交付惩罚成本和运输成本之和为优化目标,设计包含三种快速启发式策略和一种自适应基于人类学习的遗传算法(Adaptive Human-Learning-Based Genetic Algorithm, AHLBGA)进行求解. 针对分布式流水车间和车辆配送集成调度问题,Hou等[13]以最小化惩罚成本为优化目标,设计一种结合特定策略的头脑风暴算法(Brain Storm Optimization Algorithm with Some Particular Strategies, BSOASPS)进行求解;Fu等[14]以最小化最大完工时间为优化目标,提出一种增强黑寡妇算法(Enhanced Black Widow Optimization Algorithm, EBWOA)进行求解. 由文献调研可知,多数研究者仅考虑单机、并行机等简单分布式生产环境下的车辆配送集成调度问题. 而对于分布式加工装配流水车间(DTAFSP)和车辆配送集成调度问题(ISP_DPAVD)这类重要问题的研究还非常有限.

分布估计算法(Estimation of Distribution Algorithm, EDA)是一种基于统计学习的智能优化算法,EDA利用优质个体或解的信息构建概率模型并通过采样该模型生成新解来实现搜索. 可较好避免传统智能算法的核心操作容易破坏优良解内部模式的不足. 近年来EDA已被应用于求解车间调度问题. 例如,Wang等[15]针对以最小化完工时间为优化目标的分布式装配流水车间调度问题,设计一种基于模因算法的EDA(EDA-based Memetic Algorithm, EDA-MA)进行求解. Zhou等[16]针对分布式流水车间和车辆运输集成调度问题,设计一种超启发式三维EDA算法(Hyper-Heuristic three-Dimensional Estimation of Distribution Algorithm, HH3DEDA)进行求解. Guo等[17]针对以生产成本、拖期时间为优化目标的模糊分布式装配柔性车间调度问题,设计一种基于混合差分搜索和变邻域搜索的EDA(HEDA)进行求解. Zhang等[18]针对以最小化完工时间为优化目标的分布式装配流水车间调度问题,设计一种MCEDA(Matrix-Cube-based EDA, MCEDA)进行求解. 上述文献中的EDA在求解车间调度问题时,通常将相邻两个编号定义为块结构,采用二维概率模型或矩阵记录优质解的块结构信息[15~17]. 然而,二维概率模型受限于其结构,只能记录块结构的序列信息,无法记录块结构所处的具体位置. 相比而言,三维概率模型能同时记录块结构的序列信息和位置信息,可更合理地引导搜索方向[18]. 因此,本文研究如何设计基于三维概率模型的EDA来求解ISP_DPAVD.

综上,针对以最小化运输和延迟惩罚总成本为优化目标的ISP_DPAVD,本文提出一种混合三维分布估计算法(Hybrid Three-Dimensional Estimation of Distribution Algorithm, H3DEDA)进行求解. ISP_DPAVD包含加工装配阶段子问题(子问题1)和车辆配送阶段子问题(子问题2),且两个子问题间存在耦合. 首先,考虑ISP_DPAVD的特点,在解码部分(见3.1节),设计可由子问题1的解(子解1)来直接获取子问题2的解(子解2)的启发式规则,以实现子问题间的部分解耦并明显

缩减搜索空间. 其次,在全局搜索部分(见3.3节和3.4节),采用三维概率模型合理学习优质解中子解1对应的块结构模式信息,并设计自适应概率模型更新策略以避免算法过早收敛,从而可引导全局搜索较快到达解空间中的不同优质区域. 然后,在局部搜索部分(见3.5节),设计基于工厂内和工厂间邻域操作的两阶段变邻域搜索方法,用于对全局搜索发现的优质区域执行深入搜索,同时利用块结构性质进一步提升搜索效率. 仿真实验和算法对比验证了所提H3DEDA是求解ISP_DPAVD的有效算法.

2　问题描述

2.1　符号定义

文中符号定义见表1.

表1　符号定义表

符号	说明	符号	说明
N	总工件数	N^f	分配至工厂f的工件数
i,j	工件或客户编号$i,j\in\{1,\cdots,N\}$	$\boldsymbol{\pi}^f$	工厂f内,工件的加工装配序列$\boldsymbol{\pi}^f=\{\pi^f(1),\cdots,\pi^f(l),\cdots,\pi^f(N^f)\}$
l	位置编号	$N^{f,h}$	工厂f内,车辆h装载的工件数
F	工厂总数	$\boldsymbol{\varpi}^{f,h}$	工厂f内,第h辆车的工件配送序列$\boldsymbol{\varpi}^{f,h}=\{\varpi^{f,h}(1),\cdots,\varpi^{f,h}(l),\cdots,\varpi^{f,h}(N^{f,h})\}$
f	工厂编号$f\in\{1,\cdots,F\}$	W_j	工件j的重量
M	机器总数	$V^{f,h}$	在工厂f内,第h辆车的出发时间
k	机器/部件编号	A_j	客户j工件的实际交付时间
H^f	工厂f所使用车辆总数	$R_{j,i}$	车辆在客户j与客户i之间的运输时间成本
h	车辆编号$h\in\{1,\cdots,H^f\}$	D_j	工件j的预定交付期
j_k	组成工件j的k部件	T_j	工件j的延迟交付时间
$t_{j,k}$	工件j的k部件在机器k上的加工时间	η	每辆车的发车成本
$s_{j,k}$	工件j的k部件在机器k上的准备时间	μ	每个工件的单位延迟成本
a_j	工件j在装配机上的装配时间	PC	车辆运输成本
b_j	工件j在装配机上的准备时间	DC	延迟惩罚成本
C_j^f	工厂f内工件j的装配完工时间	TC	总成本
σ_j^f	工厂f内工件j加工前装配机器的空闲时间	X_j^f	若工件j在工厂f内生产为1;否则为0
Δ_j^f	工厂f内工件j加工之前装配机器的最大空闲时间	$Y_{i,j}^f$	若在工厂f,工件j在工件i后直接生产为1;否则为0
n	$n=N+F-1$,表示编码个体π的长度	$G_{i,j}^{f,h}$	若在工厂f的车辆h上,工件j在工件i后直接交付为1;否则为0
$\boldsymbol{\pi}$	$\boldsymbol{\pi}=\{\pi(1),\pi(2),\cdots,\pi(l),\cdots,\pi(n)\}=\{\pi^1,0,\pi^2,0,\cdots,0,\pi^F\}$,表示编码个体		

2.2　问题描述

ISP_DPAVD可描述为:将N位客户的N个工件产品,分配至F个位于不同地理位置的同构工厂中以$\boldsymbol{\pi}^f$的顺序进行加工,加工完成后直接在该工厂的装配机器上组装成为工件产品. 待工件装配完成后,各个工厂将工件通过配送车辆以$\boldsymbol{\varpi}^{f,h}$的顺序配送至各个工件所属客户点. 整个过程主要分为加工阶段,装配阶段和配送阶段,其中加工阶段和装配阶段统称为生产阶段.

在加工阶段,每个工件的组成部件需要在各自的专用机器上进行加工,加工完成的部件将被输送至装配机组装成为工件产品. 在装配阶段,只有当属于该工件的所有组成部件都到达装配机且装配机处于空闲状态才会进行组装操作,完成的工件将被分批放入配送车辆中. 其中,每个工件及其部件只能被分配到一个工厂,每台机器同一时刻只能执行一个任务,且机器不存在故障或者被打断的情况. 在配送阶段,每个工厂都拥有足够数量物流车队,每辆车都具有相同的发车成本和车辆载重,所有工件在不违反车辆载重约束下被放入车内,按照配送路线依次配送给客户,在完成配送任务后返回各自的工厂内. ISP_DPAVD示意图如图1所示.

在生产阶段,工厂f的$\pi^f(l)$工件装配完工时间计算公式如下,其中,0和$N+1$表示虚拟工件:

$$\sum_{f=1}^{F}X_j^f=1,\forall j=1,2,\cdots,N. \tag{1}$$

$$Y_{i,j}^f+Y_{j,i}^f\le 1,\forall i,\ j\in\{0,1,\cdots,N+1\},i\ne j \tag{2}$$

$$\sum_{i=0}^{N}Y_{i,j}^f=X_j^f,\forall j=\{1,\cdots,N\},\forall f=\{1,\cdots,F\} \tag{3}$$

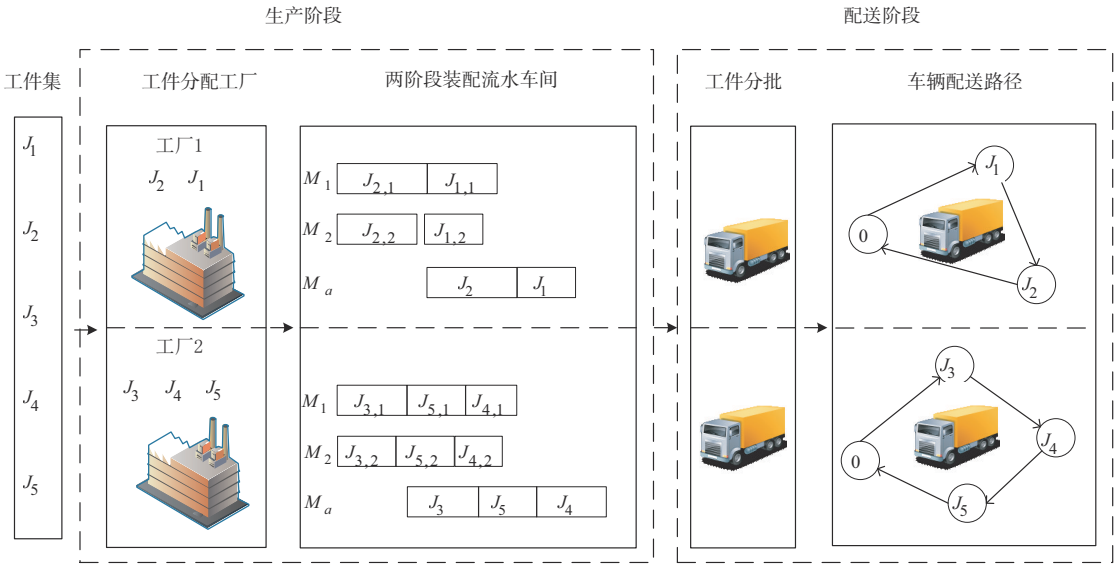


图1 ISP_DPAVD示意图

$$C_{\pi^f(l)}=\max\left\{\max_{k=1,2,\cdots,M}\sum_{t=1}^{l}\left(s_{\pi^f(t),k}+t_{\pi^f(t),k}\right),\ C_{\pi^f(l-1)}+b_{\pi^f(l)}\right\}+a_{\pi^f(l)}. \tag{4}$$

$$\sigma_{\pi^f(l)}=\max_{k=1,\cdots,M}\left\{\sum_{g=1}^{l}\left(t_{\pi^f(g),k}+s_{\pi^f(g),k}\right)\right\}-\sum_{g=1}^{l-1}\left(a_{\pi^f(g)}+b_{\pi^f(g)}\right)-b_{\pi^f(l)} \tag{5}$$

$$\Delta_{\pi^f(l)}=\max\left\{0,\sigma_{\pi^f(1)},\sigma_{\pi^f(2)},\cdots,\sigma_{\pi^f(l)}\right\} \tag{6}$$

$$C_{\pi^f(l)}=\Delta_{\pi^f(l)}+\sum_{g=1}^{l}\left(a_{\pi^f(g)}+b_{\pi^f(g)}\right) \tag{7}$$

配送阶段工件j延迟时间T_j计算公式如下：

$$\sum_{h=1}^{H}\sum_{i=0}^{N}G_{i,j}^{f,h}=X_j^f,\forall j=\{1,\cdots,N\},\ f=\{1,\cdots,F\} \tag{8}$$

$$\sum_{l=1}^{N^{f,h}}w_{\varpi^{f,h}(l)}\leqslant Q,\forall f\in\{1,\cdots,F\},\forall h\in\{1,\cdots,H^f\} \tag{9}$$

$$V^{f,h}=\max_{l=1,\cdots,N^{f,h}}\left\{C^f_{\varpi^{f,h}(l)}\right\},\forall f\in\{1,\cdots,F\},\forall h\in\{1,\cdots,H^f\} \tag{10}$$

$$A_{\varpi^{f,h}(1)}=V^{f,h}+R_{0,\varpi^{f,h}(1)} \tag{11}$$

$$A_{\varpi^{f,h}(l)}=A_{\varpi^{f,h}(l-1)}+R_{\varpi^{f,h}(l-1),\varpi^{f,h}(l)},\forall l\in\{2,\cdots,N^{f,h}\} \tag{12}$$

$$T_j=\max\left\{0,A_j-D_j\right\},\forall j\in\{1,\cdots,N\} \tag{13}$$

ISP_DPAVD优化目标计算公式如下：

$$\mathrm{DC}=\mu\sum_{j=1}^{N}T_j \tag{14}$$

$$\mathrm{PC}=\eta\sum_{f=1}^{F}\sum_{h=1}^{H}\sum_{i=1}^{N}G_{0,i}^{f,h}+\sum_{f=1}^{F}\sum_{h=1}^{H}\sum_{i=0}^{N}\sum_{j=0}^{N}R_{i,j}G_{i,j}^{f,h} \tag{15}$$

$$\mathrm{TC}=\mathrm{PC}+\mathrm{DC} \tag{16}$$

ISP_DPAVD的优化目标为找到一个最优排序$\boldsymbol{\pi}^*$，使得优化目标TC最小. 其中，式(1)表示每个工件都将被分配到一个工厂内；式(2)、式(3)表示工厂内的产品都有一个前置工件和后续工件被加工；式(4)用来计算生产阶段工件的完工时间，其中，$C_{\pi^f(0)}=0$；式(5)、式(6)用来计算装配机加工工件前的最大空闲时间；式(7)表示通过机器空闲时间计算工件完工时间；式(8)表示每个工件都要被配送至所属客户；式(9)为车辆载重约束；式(10)用于计算每辆车的发车时间；式(11)、式(12)用于计算车辆到达客户点的时间；式(13)用于计算每位客户工件的延迟时间；式(14)~式(16)分别计算车辆的延迟惩罚成本、运输成本以及总成本.

2.3 问题分析和算法设计

ISP_DPAVD中加工、装配和配送三个子问题彼此影响、相互耦合，其解空间近似为三个子问题解空间大小的乘积. 针对ISP_DPAVD这类多阶段强耦合的复杂调度问题，若直接对其进行求解难度较大. 因此需要设计可减小其解空间、降低多阶段耦合度的高效算法对其进行求解.

ISP_DPAVD包含加工装配阶段子问题(子问题1)和车辆配送阶段子问题(子问题2). 其中，子问题1需要将每个产品分配至工厂，以及确定相应工厂内的加工和装配序列. 子问题2需要将子问题1装配完成的工件进行分批(即将工件放入配送车辆内)，并确定每辆车的配送序列. 由于子问题1中产品的完工时间直接影响子问题2中车辆的最早出发时间和产品交付时间，工件完工时间越早，则在有限的交货时间内对于工件配送阶段的优化空间越大. 因此若在一个优质的子问

题1调度方案上对子问题2进行求解,能够有效提高算法搜索效率以及保证ISP_DPAVD求解质量. 故本文采用三维EDA首先对子问题1进行求解,并对加工和装配问题进行统一编解码以提高求解效率[19]. 在对子问题2求解时,大多研究者通常将车辆内工件的加工装配序列作为车辆的配送序列[8,10~14],然而最优加工序列并不一定是最优配送序列. 为尽可能减小配送成本,本文算法将采用结合启发式规则的解码策略,在确定子问题1加工序列的基础上,设计基于邻域变换的启发式规则对配送序列进行适当的调整和优化,最后将前后阶段子问题的解合并得到ISP_DPAVD的解. 显然,采用上述策略能够对ISP_DPAVD解空间大幅度进行削减,在保证求解质量的同时一定程度上实现两个子问题间的解耦.

3　混合三维分布估计算法

3.1　编码与解码

在编码中,每个个体由$[1,2,\cdots,N]$不重复数字和$F-1$个0组成. 其中,数字0代表工厂分隔符,剩余的数字代表对应的工件编号. 以规模为$N=8,F=3$的问题为例,具体的编码如图2所示.

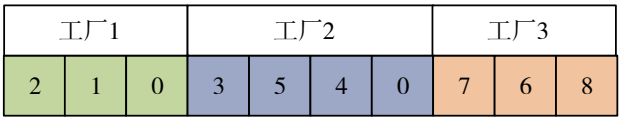


图2　编码示意图

在解码中,结合2.3节问题分析,设计一种分层解码策略. 第1层,将个体编码进行解码得到生产阶段的加工、装配序列信息;第2层,基于前一阶段的解码信息采用启发式规则进行配送阶段的解码,以此来得到工件分批情况和车辆配送序列信息. 具体如下.

(1)生产阶段解码:加工和装配序列:将个体编码按工厂分隔符划分为F段,从左往右每段对应为第1至第F个工厂的工件的加工和装配序列.

(2)配送阶段解码:配送阶段解码包含了对子问题2的求解. 首先采用启发式规则1对工件进行分批,然后采用结合Insert域操作的启发式规则2对车辆配送路径进行优化. 启发式规则描述如下.

启发式规则1:在对工件分批时,将工厂内所有工件以加工装配序列按照先完工先装车的规则分配给车辆,若车辆负荷超过最大载重约束,则投入新的车辆,然后按先装满先运输的规则进行发车.

启发式规则2:在确定车辆配送序列时,以车辆中装载工件加工装配序列执行以下操作:对序列每一个编号其他所有的位置都执行Insert作,得到一系列新邻域或新解,并输出其中最好的新邻域.

以图2所示编码为例,解码过程如图3所示. 令工厂2的生产和配送阶段数据如表2所示,车辆载重$Q=30$,车速为1,则工厂2各阶段数据计算过程如下,装配机空闲时间 $\sigma_{\pi^2(1)}=\max\{24,33\}-13=20$, $\sigma_{\pi^2(2)}=\max\{65,76\}-45=31$, $\sigma_{\pi^2(3)}=\max\{101,119\}-78=41$. 因此 $\Delta_{\pi^2(1)}=\max\{0,20\}=20$, $\Delta_{\pi^2(2)}=\max\{0,20,31\}=31$, $\Delta_{\pi^2(3)}=\max\{0,20,31,41\}=41$. 装配完工时间为 $C_{\pi^2(1)}=20+40=60$, $C_{\pi^2(2)}=31+70=101$, $C_{\pi^2(3)}=41+100=141$. 客户到达时间为 $A_{\varpi^{2,1}(1)}=141+149=290$, $A_{\varpi^{2,1}(2)}=290+112=402$, $A_{\varpi^{2,1}(3)}=402+100=502$, $A_{\varpi^{2,1}(4)}=502+49=551$. 则工厂2甘特图如图4所示.

表2　工厂2数据表

编号	重量	坐标	加工/准备时间	机器1	机器2	装配机
3	5	(102,175)	t	19	23	27
			S	5	10	13
4	6	(107,75)	t	22	27	22
			S	14	16	8
5	6	(12,108)	t	27	36	25
			S	14	7	5
0		(105,26)				

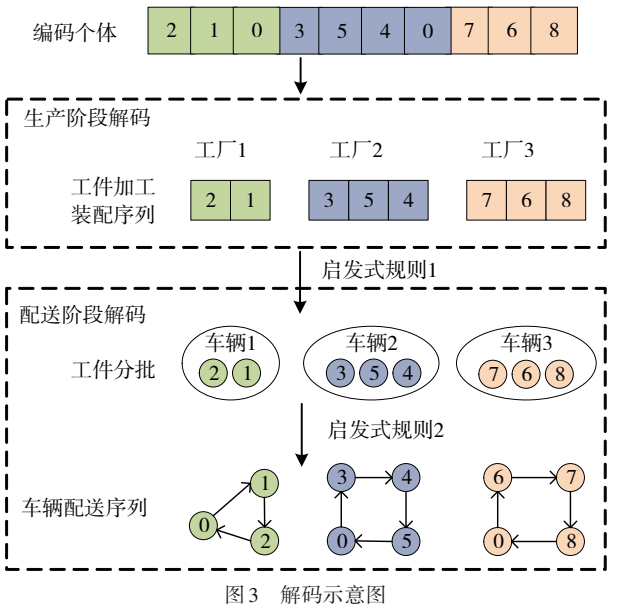


图3　解码示意图

3.2　初始化种群

针对以最小化运输和延迟惩罚的总成本为优化目标的ISP_DPAVD,提出一种策略用以提高初始解质量,具体描述如下:

步骤1　随机生成包含$[1,2,\cdots,N]$所有不重复数字

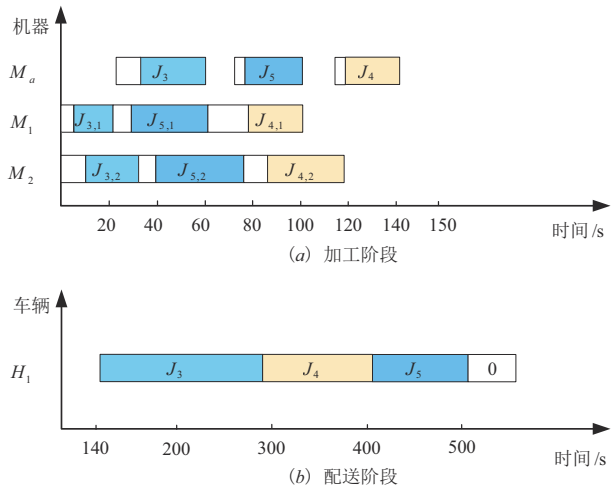


图4 工厂2甘特图

的序列,每个数字代表相应工件编号. 从序列第一个位置的工件起,将工件重量进行累加,以车辆载重约束进行分批,每个位置上的工件依次执行上述操作,直到最后一个位置上的工件分批完成.

步骤2 从左到右依次取出批次,将批次插入到各个工厂序列中,按式(1)~式(16)计算各个工厂的评价值,选择评价值最小的工厂作为其加工工厂. 每个批次依次执行上述操作,直到最后一个批次分配完成.

步骤3 在第一个工厂至倒数第二个工厂的加工序列末尾插入0,将每两个相邻工厂的加工序连接起来即可得到种群个体.

为尽可能兼顾解的质量和多样性,将初始种群一半个体按上述策略生成,其余个体采用随机方式生成.

3.3 三维概率模型

定义POP(G)为第G代种群,规模为ps,EPOP(G)为POP(G)的优质子种群,ρ为优质个体占比,EPOP(G)规模eps=ps×ρ. 令EPOP(G)的第e个个体为$\boldsymbol{\pi}_E^{G,e}=\left[\boldsymbol{\pi}_E^{G,e}(1),\cdots,\boldsymbol{\pi}_E^{G,e}(n)\right]$,三维矩阵$\boldsymbol{M}_{n\times n\times n}^G$和$\boldsymbol{P}_{n\times n\times n}^G$分别为统计模型和概率模型.

3.3.1 统计模型

三维矩阵$\boldsymbol{M}_{n\times n\times n}^G$用来记录EPOP($G$)不同位置上块结构数量的分布信息,$\boldsymbol{M}_{n\times n\times n}^G$为块结构$\left[y,z\right]$在$x$位置上出现的次数,具体描述如下:

$$\boldsymbol{M}_{n\times n\times n}^G(x)=\begin{bmatrix}\boldsymbol{M}_{n\times n\times n}^G(x,1)\\ \vdots\\ \boldsymbol{M}_{n\times n\times n}^G(x,n)\end{bmatrix}=\begin{bmatrix}M_{n\times n\times n}^G(x,1,1) & \cdots & M_{n\times n\times n}^G(x,1,n)\\ \vdots & \ddots & \vdots\\ M_{n\times n\times n}^G(x,n,I) & \cdots & M_{n\times n\times n}^G(x,n,n)\end{bmatrix},\quad x=1,\cdots,n-1 \tag{17}$$

$$B_{n\times n\times n}^{G,e}(x,y,z)=\begin{cases}1, & \text{若}y=\pi_E^{G,e}(x),z=\pi_E^{G,e}(x+1)\\ 0, & \text{否则}\end{cases},\quad x=1,\cdots,n-1;\ y,z=1,\cdots,n;\ e=1,\cdots,\text{eps} \tag{18}$$

其中,式(17)为三维模型的层次结构;式(18)为示性函数,用于记录EPOP(G)中第e个个体的块结构信息.

3.3.2 概率模型

三维矩阵$\boldsymbol{P}_{n\times n\times n}^G$用来存储EPOP($G$)块结构的概率分布信息,通过对$\boldsymbol{P}_{n\times n\times n}^G$采样和更新以实现对优质子种群的学习和积累. 令$\boldsymbol{P}_{n\times n\times n}^G$为块结构$\left[y,z\right]$出现在$x$位置上的概率值,具体描述如下:

$$\boldsymbol{P}_{n\times n\times n}^G(x)=\begin{bmatrix}\boldsymbol{P}_{n\times n\times n}^G(x,1)\\ \vdots\\ \boldsymbol{P}_{n\times n\times n}^G(x,n)\end{bmatrix}=\begin{bmatrix}P_{n\times n\times n}^G(x,1,1) & \cdots & P_{n\times n\times n}^G(x,1,n)\\ \vdots & \ddots & \vdots\\ P_{n\times n\times n}^G(x,n,1) & \cdots & P_{n\times n\times n}^G(x,n,n)\end{bmatrix} \tag{19}$$

其中,式(19)为在x位置上块结构的概率分布.

3.3.3 自适应三维概率模型更新策略

在EDA中,概率模型通过统计EPOP(G)的块结构信息来完成概率值的更新,而概率值的大小又影响着新个体的生成,因此概率模型在算法中有着重要的作用. 以往的研究往往将更新概率模型相关参数优质子种群个数eps和学习效率r设定为恒值. 然而在算法搜索过程中,EPOP(G)个体之间相似度会由低到高变化,其有效信息也会随之减少,若概率模型按照固定的eps和r去积累、学习块结构信息,容易造成概率模型中同一位置上不同块结构概率差值持续变大,导致采样结果单一,全局搜索陷入局部最优,同时也会浪费大量时间在重复的工作上,影响算法搜索效率. 故设计一种自适应概率模型更新策略,能够针对不同搜索时期EPOP(G)个体特点,动态的选择相应eps和r的数值完成概率模型信息更新,降低算法陷入局部最优的可能性.

同时,在三维概率模型的更新上设置判断条件,定义P_{d}为优质子种群多样性指数,用于衡量优质子种群个体之间的相似度,κ为一个阈值. 令π^h为历史最优解,π^{c}为当前最优解. P_{d}计算步骤如下:

步骤1 统计所有优质子种群中第τ个位置上不同编号出现的个数$S_\tau,\tau=1,2,\cdots,n$. 若在第τ个位置上只出现同一个编号时,令$S_\tau=0$.

步骤2 按式(20)计算得到P_{d}:

$$P_{\mathrm{d}}=\frac{1}{n}\left(\sum_{\tau=1}^{n}\frac{S_\tau}{\text{eps}}\right) \tag{20}$$

P_{d}值越小则优质子种群之间的相似度越高,若$P_{\mathrm{d}}=0$,则表示优质子种群中所有个体都是相同的. 譬如,EPOP(G)中的个体分别为$\boldsymbol{\pi}_E^{G,1}=\left[3,4,1,2,5\right]$,$\boldsymbol{\pi}_E^{G,2}=$

$[1,2,3,4,5]$, $\boldsymbol{\pi}_E^{G,3}=[3,1,4,2,5]$, 则 $S_1=2$, $S_2=3$, $S_3=3$, $S_4=2$, $S_5=0$, $P_{\mathrm{d}}=(2/3+3/3+3/3+2/3+0)/5=0.64$.

概率模型 $\boldsymbol{P}_{n\times n\times n}^{G}$ 具体的更新步骤如下:

步骤 1　若 $G=0$, 则继续执行步骤 2; 若 $G\geqslant 1$, 则跳转至步骤 3.

步骤 2　按式(21)对 $\boldsymbol{P}_{n\times n\times n}^{0}$ 进行初始化.

$$P_{n\times n\times n}^{0}(x,y,z)=1/n^2,\quad x=1,\cdots,n-1;\ y,z=1,\cdots,n. \tag{21}$$

步骤 3　计算 P_{d}, 若 $P_{\mathrm{d}}\geqslant\kappa$, 则 $\mathrm{eps}=\rho\times\mathrm{ps}$, $r=\chi$, 跳转至步骤 5.

步骤 4　若 $\boldsymbol{\pi}^{\mathrm{c}}$ 优于 $\boldsymbol{\pi}^{\mathrm{h}}$, 则 $\boldsymbol{\pi}^{\mathrm{h}}=\boldsymbol{\pi}^{\mathrm{c}}$, $\mathrm{eps}=1$, $r=\chi$; 否则, $\mathrm{eps}=0$, $r=0$.

步骤 5　根据如下式(22)和式(23)分别计算 $\boldsymbol{M}_{n\times n\times n}^{G-1}(x)$ 和 $S^{G-1}(x)$, 并按照式(24)更新 $P_{n\times n\times n}^{G}(x,y,z)$.

$$M_{n\times n\times n}^{G}(x,y,z)=\sum_{e=1}^{\mathrm{eps}}B_{n\times n\times n}^{e,G}(x,y,z),\quad x=1,\cdots,n-1;\ y,z=1,\cdots,n \tag{22}$$

$$S^{G}(x)=\sum_{y=1}^{n}\sum_{z=1}^{n}M_{n\times n\times n}^{G}(x,y,z) \tag{23}$$

$$P_{n\times n\times n}^{G}(x,y,z)=(1-r)\times P_{n\times n\times n}^{G-1}(x,y,z)+r\times\boldsymbol{M}_{n\times n\times n}^{G-1}(x)/S^{G-1}(x),\quad x=1,\cdots,n-1;\ y,z=1,\cdots,n \tag{24}$$

3.4　新种群生成方法

H3DEAD 通过对 $\boldsymbol{P}_{n\times n\times n}^{G}$ 采样生成新种群, 以执行对问题解空间的全局搜索. 令 $[\pi^{G,e}(\tau-1),\pi^{G,e}(\tau)]$ 为 POP(G) 第 e 个个体在 $\tau-1$ 位置上的块结构, 由于 τ 位置编号被选中的概率存储在 $\boldsymbol{P}_{n\times n\times n}^{G-1}(\tau-1)$ 中, 故需要对 $\boldsymbol{P}_{n\times n\times n}^{G-1}$ 的第 $\tau-1$ 层进行采样. 此外, 当 $\tau=1$ 时, 上述的方法无法对 $\tau-1$ 层进行采样, 需要设计一种首位置采样方法. 令 e 为个体序号, τ 为位置序号, J_{s} 为选中编号, 完整的 POP(G) 采样更新步骤如下:

步骤 1　令 $\tau=1$, $e=1$.

步骤 2　若 $\tau=1$, 则继续执行步骤 3; 否则跳转至步骤 4.

步骤 3　先按式(25)计算 $S^{G-1}(y)$, 再随机产生一个数 $p_{\mathrm{s}}\in\left[0,\sum_{t=1}^{n}S^{G-1}(t)\right]$. 若 $p_{\mathrm{s}}\in[0,S^{G-1}(1)]$, 则 $J_{\mathrm{s}}=1$; 若 $p_{\mathrm{s}}\in(S^{G-1}(t),S^{G-1}(t+1)]$, $(t=1,\cdots,n-1)$, 则 $J_{\mathrm{s}}=t+1$.

$$S^{G-1}(y)=\sum_{z=1}^{n}P_{n\times n\times n}^{G-1}(1,y,z),y=1,\cdots,n \tag{25}$$

步骤 4　随机产生一个数 $p_{\mathrm{s}}\in\left[0,\sum_{d=1}^{n}P_{n\times n\times n}^{G-1}(\tau-1,\pi^{G,e}(\tau-1),d)\right)$. 若满足 $p_{\mathrm{s}}\in[0,P_{n\times n\times n}^{G-1}(\tau-1,\pi^{G,e}(\tau-1),1)]$, 则 $J_{\mathrm{s}}=1$; 若 $p_{\mathrm{s}}\in\left(\sum_{d=1}^{t}P_{n\times n\times n}^{G-1}(\tau-1,\pi^{G,e}(\tau-1),d),\sum_{d=1}^{t+1}P_{n\times n\times n}^{G-1}(\tau-1,\pi^{G,e}(\tau-1),d)\right]$, $(t=1,\cdots,n-1)$, 则 $J_{\mathrm{s}}=t+1$.

步骤 5　令 $\tau=\tau+1$, 若 $\tau\leqslant n$, 则跳转至步骤 2; 否则跳转至步骤 6.

步骤 6　令 $e=e+1$, 若 $e\leqslant\mathrm{ps}$, 则跳转至步骤 2; 否则跳转至步骤 7.

步骤 7　输出 POP(G), 并更新 EPOP(G).

3.5　局部搜索

为增强 H3DEDA 局部搜索能力, 本文基于 Swap 和 Insert 种邻域操作设计两阶段变邻域搜索机制, 对优质子种群 EPOP(G) 中的个体 $\boldsymbol{\pi}^{G,e}$ (对应优质区域)执行细致且高效的搜索.

3.5.1　扰动策略

扰动的目的是为了避免搜索陷入局部最优, 但扰动过于频繁容易增大搜索的随机性, 会降低搜索的引导性. 因此, 采用优质子种群多样性指数 P_{d} 来选择扰动方式. 当每次执行局部搜索时, 按如下两种情况进行扰动:

(1) 如果 $P_{\mathrm{d}}<\kappa$ (κ 为阈值), 则表明当前优质子种群个体间相似度过高(即各个体对应的搜索区域较为集中), 且各个体较大概率已接近或到达局部最优. 这时对 $\boldsymbol{\pi}^{G,e}$ 继续执行局部搜索, 已难以获得有效改进. 在此情况下, 先对 $\boldsymbol{\pi}^{G,e}$ 执行 20 次扰动操作 Interchange 得到新邻域或新解, 并以此直接更新 $\boldsymbol{\pi}^{G,e}$, 再对更新后的 $\boldsymbol{\pi}^{G,e}$ 执行局部搜索.

(2) 如果 $P_{\mathrm{d}}\geqslant\kappa$, 则表明当前优质种群个体仍分布在较多不同区域. 这时可对 $\boldsymbol{\pi}^{G,e}$ 继续执行局部搜索. 在此情况下, 先对 $\boldsymbol{\pi}^{G,e}$ 执行 20 次扰动操作 Interchange 得到新邻域或新解, 并取 $\boldsymbol{\pi}^{G,e}$ 和新解中较优者作为 $\boldsymbol{\pi}^{G,e}$, 再对当前 $\boldsymbol{\pi}^{G,e}$ 执行局部搜索.

情况(1)和(2)中扰动操作 Interchange 的具体定义如下:

Interchange($\boldsymbol{\pi}$): 在 $\boldsymbol{\pi}$ 中, 随机生成 τ 和 τ', 将 τ 位置和 τ' 位置的编号互换位置.

3.5.2　两阶段变邻域搜索

为确保局部搜索具有一定深度, 设计基于 2 种厂间邻域操作和 2 种厂内邻域操作的两阶段变邻域局部搜索. 每种邻域操作实际对应 1 种邻域结构. 相关的邻域操作设计如下.

(1)　第一阶段工厂间操作

针对工厂间的工件, 基于 F_Swap 和 F_Insert 邻域操作, 设计带"改进跳出"策略的变邻域搜索(见局部搜索的步骤 3 和步骤 4), 用以确保为所有工件找到较优的工厂分配方案. 令 $\boldsymbol{J}_{\theta}$ 为未调度工件序列, F_Swap 和

F_Insert的具体定义如下：

F_Swap$(\boldsymbol{\pi},\boldsymbol{J}_\theta)$：从工件序列$\boldsymbol{J}_\theta$中取出工件$J_l$，将其与各工厂$\boldsymbol{\pi}^f$中所有工件互换，得到一系列新邻域或新解，并输出其中最好的新邻域.

F_Insert$(\boldsymbol{\pi},\boldsymbol{J}_\theta)$：从工件序列$\boldsymbol{J}_\theta$中取出工件$J_l$，将其插入到各工厂$\boldsymbol{\pi}^f$中所有可能的位置，得到一系列新邻域或新解，并输出其中最好的新邻域.

（2） 第二阶段工厂内操作

针对工厂内的工件，基于Job_Swap和Job_Insert邻域操作，设计变邻域下降搜索（见局部搜索的步骤5和步骤6），用于确定厂内较优的工件加工装配序列. Job_Swap和Job_Insert的具体定义如下：

Job_Swap($\boldsymbol{\pi}^f$)：从$\boldsymbol{\pi}^f$中取出工件$J_{\pi^f(l)}$，将其与工厂内的所有工件互换，得到一系列新邻域或新解，并输出其中最好的新邻域.

Job_Insert($\boldsymbol{\pi}^f$)：从$\boldsymbol{\pi}^f$中取出工件$J_{\pi^f(l)}$，将其插入到工厂内所有可能的位置，得到一系列新邻域或新解，并输出其中最好的新邻域.

基于上述定义，局部搜索设计如下.

步骤1 对于优质子种群EPOP(G)中的每个解$\boldsymbol{\pi}^{G,e}$，令$\boldsymbol{\pi}=\boldsymbol{\pi}^{G,e}$.

步骤2 执行20次Interchange($\boldsymbol{\pi}$)，若$P_\mathrm{d}\geqslant\kappa$，除非扰动后的解优于$\boldsymbol{\pi}$，否则不更新$\boldsymbol{\pi}$；若$P_\mathrm{d}<\kappa$，以扰动后的解更新$\boldsymbol{\pi}$.

步骤3 令$\boldsymbol{J}_\theta=\boldsymbol{\pi}$，对$\boldsymbol{\pi}$执行工厂间操作F_Swap$(\boldsymbol{\pi},\boldsymbol{J}_\theta)$若获得更优解，则将$\boldsymbol{\pi}$更新为更优解，并跳转至步骤5.

步骤4 对$\boldsymbol{\pi}$执行工厂间操作F_Insert$(\boldsymbol{\pi},\boldsymbol{J}_\theta)$. 若获得更优解，则将$\boldsymbol{\pi}$更新为更优解；否则，跳转至步骤7.

步骤5 对$\boldsymbol{\pi}$执行工厂内操作Job_Swap($\boldsymbol{\pi}$). 若获得更优解，则将π更新为更优解，并重新执行步骤5.

步骤6 对$\boldsymbol{\pi}$执行工厂内操作Job_Insert($\boldsymbol{\pi}$). 若获得更优解，则将$\boldsymbol{\pi}$更新为更优解，并跳转至步骤5.

步骤7 令$\boldsymbol{\pi}^{G,e}=\boldsymbol{\pi}$.

在上述局部搜索中，步骤2为扰动环节（见3.5.1节），执行此扰动有利于帮助算法跳出局部最优，并可持续推动全局和局部搜索到达解空间中不同区域. 步骤3至步骤7为局部搜索执行环节，其中各步骤均采用改进更新策略（即发现更优解则更新$\boldsymbol{\pi}$），同时步骤5和步骤6采取改进重启策略（即发现更优解则重启步骤5）. 由于采用4种邻域操作来执行变邻域迭代局部搜索，可驱动算法在到达各邻域结构共同的局部最优解前一直向下挖掘，有助于实现深度搜索并发现高质量解.

3.5.3 判断厂内Swap邻域操作有效性的块结构性质

基于ISP_DPAVD中解（即个体）的块结构特性，提出以下定理，用于快速判断无效的厂内Swap邻域操作，可提高局部搜索效率.

令个体$\boldsymbol{\pi}_1$对应的工厂f加工序列为$\boldsymbol{\pi}_1^f$，将$\boldsymbol{\pi}_1^f$块结构τ位置的工件i和$\tau+1$位置的工件j互换，得到个体$\boldsymbol{\pi}_2$及其对应工厂f的加工序列$\boldsymbol{\pi}_2^f$.

引理1 在工件序列$\boldsymbol{\pi}_1^f$和$\boldsymbol{\pi}_2^f$中，对于$l\in\{\tau+2,\cdots,N^f\}$，若满足$\max\left\{\sigma_{\pi_1^f(\tau)},\sigma_{\pi_1^f(\tau+1)}\right\}\geqslant\max\left\{\sigma_{\pi_2^f(\tau)},\sigma_{\pi_2^f(\tau+1)}\right\}$，则有$C_{\pi_2^f(l)}\leqslant C_{\pi_1^f(l)}$.

证明 对于工件序列$\boldsymbol{\pi}_1^f$和$\boldsymbol{\pi}_2^f$，除τ和$\tau+1$位置上的工件外，其余位置的工件排序相同.

这时，由式（1）可得$C_{\pi_1^f(l),k}=C_{\pi_2^f(l),k}$，$(\forall l=1,\cdots,\tau-1;\forall k=1,\cdots,M)$；同时由式（5）可得$\sigma_{\pi_1^f(l)}=\sigma_{\pi_2^f(l)},(\forall l\neq\tau,\tau+1)$.

显然，当$l\in\{\tau+2,\cdots,N^f\}$时，结合引理中的前提条件$\max\left\{\sigma_{\pi_1^f(\tau)},\sigma_{\pi_1^f(\tau+1)}\right\}\geqslant\max\left\{\sigma_{\pi_2^f(\tau)},\sigma_{\pi_2^f(\tau+1)}\right\}$，以及式（6）和式（7），可得$C_{\pi_1^f(l)}\geqslant C_{\pi_2^f(l)}$成立.

证毕.

定理1 若在同一辆内，在τ位置的块结构满足$\max\left\{\sigma_{\pi_1^f(\tau)},\sigma_{\pi_1^f(\tau+1)}\right\}\geqslant\max\left\{\sigma_{\pi_2^f(\tau)},\sigma_{\pi_2^f(\tau+1)}\right\}$时，则对于任意$l\in\{\tau,\cdots,N^f\}$，有$\mathrm{TC}_{\pi_2}\leqslant\mathrm{TC}_{\pi_1}$.

证明 对于ISP_DPAVD的任意解π，存在如下性质：

性质1 任何车辆的发车时间提前，会使得客户交付时间提前或不变，进而车辆延迟惩罚成本至少不会增大.

性质2 在工厂间不发生工件转移的情况下，工厂成本只会受该工厂内工件加工序列和配送序列变换的影响.

当满足定理1的条件时，根据引理1可得$C_{\pi_2^f(l)}\leqslant C_{\pi_1^f(l)}$，从而使得对应解中部分车辆的发车时间可能提前，即有$V_{\pi_2}^{f,h}\leqslant V_{\pi 1}^{f,h}(h=1,\cdots,H^f)$.

这时，对于任意$l\in\{\tau,\cdots,N^f\}$，对应解的加工装配时间可能减小，且解中部分车辆的延迟惩罚成本至少不会增大（根据性质1），其余工厂成本不变（根据性质2），即$\mathrm{TC}_{\pi_2}\leqslant\mathrm{TC}_{\pi_1}$成立.

证毕.

定理2 若在同一辆车内，当块结构满足以下条件：（1）$a_j\geqslant a_i$，$b_j\leqslant b_i$；（2）$t_{j,k}+s_{j,k}\leqslant t_{i,k}+s_{i,k},\forall k=1,\cdots,M$；（3）$\min\limits_{k=1,\cdots,M}\left\{t_{i,k}+s_{i,k}\right\}\geqslant t_{i,k}+s_{i,k}$.；则有$\mathrm{TC}_{\pi_2}\leqslant\mathrm{TC}_{\pi_1}$.

证明 当满足条件式（1）~式（3）时，则有$C_{\pi_2^f(l)}\leqslant$

$C_{\pi_1^f(l)},(\forall l=1,\cdots,N^f)$[20]. 类似定理1的证明,可得$\mathrm{TC}_{\pi_2}\leqslant\mathrm{TC}_{\pi_1}$成立.

证毕.

在3.5.2节局部搜索的步骤5中,进一步利用定理1和定理2,先对Job_Swap($\boldsymbol{\pi}^f$)中由Swap邻域操作产生的新邻域或新解的质量进行快速判断,若优于$\boldsymbol{\pi}$则计算其目标函数值,否则就直接排除该邻域,从而可避免无效计算,提升局部搜索效率.

3.6 H3DEDA算法流程

根据第3.1节至第3.5节对H3DEDA各部分的描述,H3DEDA的整体流程图如图5所示,具体步骤如下:

步骤1 初始化概率模型$\boldsymbol{P}^G_{n\times n\times n}$、算法参数种群规模ps、优质个体占比$\rho$、学习参数$\chi$.

步骤2 初始化种群,并评价种群个体.

步骤3 按式(22)计算种群多样性指数P_{d},并选择相应的更新参数.

步骤4 按式(24)自适应更新概率模型.

步骤5 对概率模型采样生成新种群并评价种群个体,计算种群多样性指数P_{d}.

步骤6 对优质子种群各个体进行局部搜索.

步骤7 判断是否满足终止条件,若不满足,则跳转至步骤3.

步骤8 输出当前优质子种群中的最好解.

在上述H3DEDA中,步骤3至步骤5为全局搜索部分,步骤6为局部搜索部分. 其中,步骤3和步骤4采用三维概率模型和自适应更新策略来有效积累优质个体或解的块结构信息;步骤5通过采样概率模型生成新种群来引导搜索尽快到达解空间中的优质区域;步骤6利用变邻域迭代局部搜索在优质区域执行深入挖掘. 由于H3DEDA的全局和局部搜索部分均得以加强,故有望成为求解ISP_DPAVD的有效算法.

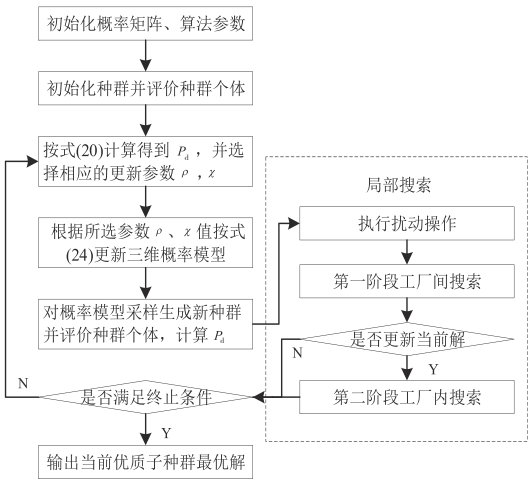


图5 算法流程图

3.7 算法复杂度分析

本节将所提H3DEDA和4.3.2节实验涉及的3种对比算法的复杂度进行分析.

3.7.1 H3DEDA复杂度

令$\max G$为算法最大迭代次数. H3DEDA的复杂度主要包含了评价个体的计算复杂度$\Delta=F\times\left(M\times N^f+H^f\times N^{f,h}\times\left(N^{f,h}-1\right)\right)$,$P_{\mathrm{d}}$计算复杂度$O\left(\mathrm{ps}\times\rho\times n\right)$,更新三维概率矩阵计算复杂度$O\left(\mathrm{ps}\times\rho\times n^3\right)$,采样概率模型生成种群计算复杂度$O\left(\mathrm{ps}\times n^2\right)$,种群进行快速排序计算复杂度为$O\left(\mathrm{ps}\times\log_{\mathrm{ps}}\right)$. 局部搜索计算复杂度$O\left(\mathrm{ps}\times\rho\times n^2\times\Delta\right)$. 故H3DEDA的整体复杂度为:

$$T_{\mathrm{H3DEDA}}=\max G\times\left[O\left(\mathrm{ps}\times n^2\right)+O\left(\mathrm{ps}\times\rho\times n\right)+O\left(\mathrm{ps}\times\log_{\mathrm{ps}}\right)+O\left(\mathrm{ps}\times\rho\times n^2\times\Delta\right)+O\left(\mathrm{ps}\times\rho\times n^3\right)\right]$$

3.7.2 对比算法复杂度

令HABCSA[8]局部搜索的最大迭代次数为λ,个体连续未改进的最大次数为ε. HABCSA复杂度主要包含雇佣蜂阶段计算复杂度$O\left(\mathrm{ps}\times n\times\lambda\times\Delta\right)$、观察蜂阶段计算复杂度$O\left(\mathrm{ps}^2\times n\times\varepsilon\times\Delta\right)$、侦查蜂阶段计算复杂度$O\left(\mathrm{ps}\times n\times\varepsilon\times\Delta\right)$和快速排序阶段计算复杂度$O\left(\mathrm{ps}\times\log_{\mathrm{ps}}\right)$,故HABCSA的整体复杂度为:

$$T_{\mathrm{HABCSA}}=\max G\left[O\left(\mathrm{ps}\times n\times\lambda\times\Delta\right)+O\left(\mathrm{ps}^2\times n\times\varepsilon\times\Delta\right)+O\left(\mathrm{ps}\times n\times\varepsilon\times\Delta\right)+O\left(\mathrm{ps}\times\log_{\mathrm{ps}}\right)\right]$$

AHLBGA[12]复杂度主要包含遗传算法复杂度$O\left(\mathrm{ps}\times n\times\Delta\right)$,更新信息矩阵复杂度$O\left(\mathrm{ps}\times\rho\times n\right)$,选择学习操作改进个体复杂度$O\left(\mathrm{ps}\times 3\times n\times\Delta\right)$,快速排序阶段复杂度$O\left(2\times\mathrm{ps}\times\log_{2\times\mathrm{ps}}\right)$. 故AHLBGA的整体复杂度为:

$$T_{\mathrm{AHLBGA}}=\max G\left[O\left(\mathrm{ps}\times n\times\Delta\right)+O\left(\mathrm{ps}\times\rho\times n\right)+O\left(\mathrm{ps}\times 3\times n\times\Delta\right)+O\left(2\times\mathrm{ps}\times\log_{2\times\mathrm{ps}}\right)\right]$$

令EBSO[13]聚类的分组数为ϑ. EBSO复杂度主要包含聚类复杂度$O\left(\mathrm{ps}\times\vartheta\right)$,生成新个体复杂度$O\left(\mathrm{ps}\times n\times\lambda\times\Delta\right)$,局部搜索复杂度$O\left(n^2\times\Delta\right)$,快速排序阶段复杂度$O\left(2\times\mathrm{ps}\times\log_{2\times\mathrm{ps}}\right)$,故EBSO的整体复杂度为:

$$T_{\mathrm{EBSO}}=\max G\left[O\left(\mathrm{ps}\times\vartheta\right)+O\left(\mathrm{ps}\times n\times\lambda\times\Delta\right)+O\left(n^2\times\Delta\right)+O\left(2\times\mathrm{ps}\times\log_{2\times\mathrm{ps}}\right)\right]$$

实际上,在各算法的整体复杂度中,除n和Δ之外的其他变量均只与算法相关,且在实际使用中被设置为常数,故H3DEDA、HABCSA、AHLBGA和EBSO的计算复杂度可分别简化为$O\left(n^2\times\Delta\right)+O\left(n^3\right)$、$O\left(n\times\Delta\right)$、

$O(n\times\Delta)$和$O(n^2\times\Delta)$. 这表明H3DEDA的复杂度相对较高. 但从第4节的实验结果可知,H3DEDA的性能最好. 这表明复杂度和算法性能并不是正相关关系(见4.3.2节分析).

4 实验结果及分析

4.1 算例生成

由于目前尚无合适的ISP_DPAVD算例,本文所有的算例的数据均在Basir[21]为解决两阶段装配车间和车辆配送集成调度问题所提供的数据分布区间上生成,具体如表3所示,工厂和客户的坐标在[0,400]区间内随机生成. 本文共有24个算例,其中$N=\{20,50,100\}$, $M=\{5,10,20\}$, $F=\{2,4,6\}$. 按照$N\times M\times F$形式组合. 所有的算法和实验均通过delphi2010编程和实现,操作系统为Windows10,CPU为英特尔I5处理器,频率为3.30 GHz,8 GB内存. 为确保比较的公平性,将各算法在相同时间内运行20次,每次运行时间均设定为$(N\times M\times F\times 20)$ms.

表3 数据分布区间表

问题参数	数据
$t_{j,k}$	[1~100]
$s_{j,k}$	[1~10]
a_j	[1~100]
b_j	[1~10]
Q	30
W_j	[1~10]
η	200
μ	1

4.2 参数设置

在H3DEDA中,关键的参数主要有种群规模ps、优质个体占比ρ、学习参数χ以及种群多样性阈值κ. 为确定4个参数的取值,本文选取规模为$[50\times 10\times 4]$的测试算例进行实验设计(Design Of Experiment,DOE)[22]分析,每个参数选取4个水平值,具体的参数水平表如表4所示.

根据表4参数水平的设置,按照规模为$L16(4^3)$的正交实验,将每组参数组合下的测试问题进行20次独立实验,运行时间均为$(N\times M\times F\times 20)$ms,取20次实验结果的平均值作为响应值,响应值越小则代表在相应的参数设置下算法性能越强. 参数正交表如表5所示.

表4 参数水平表

参数	水平设置			
	1	2	3	4
ps	20	30	40	50
ρ	0.1	0.2	0.3	0.4
χ	0.4	0.5	0.6	0.7
κ	0.1	0.2	0.3	0.4

表5 参数正交表和响应值

参数组合	参数水平				AVG
	ps	ρ	χ	κ	
1	1	1	1	1	5 560.8
2	1	2	2	2	5 191.2
3	1	3	3	3	5 179.0
4	1	4	4	4	5 179.2
5	2	1	2	3	5 220.3
6	2	2	1	4	5 171.1
7	2	3	4	1	5 156.5
8	2	4	3	2	5 141.4
9	3	1	3	4	5 295.9
10	3	2	4	3	5 219.8
11	3	3	1	2	5 144.9
12	3	4	2	1	5 175.0
13	4	1	4	2	5 380.5
14	4	2	3	1	5 134.9
15	4	3	2	4	5 164.2
16	4	4	1	3	5 179.0

将表5中每个参数在同一参数水平$L(L=1,\cdots,4)$下平均值AVG作为该参数在此L下的响应值,可得表6中的响应值,进而可得相应的极差和影响等级,其中等级越小代表该参数影响力越大. 在表6的基础上,根据各参数在4个水平下的响应值可得到对应的变化趋势图(见图6).

表6 参数组合响应值

水平	参数			
	ps	ρ	χ	κ
1	5 278	5 364	5 264	5 257
2	5 172	5 179	5 188	5 214
3	5 209	5 161	5 188	5 200
4	5 215	5 169	5 234	5 203
极差	105	203	76	57
等级	2	1	3	4

由图6可知,当H3DEDA的参数设置为ps=20,ρ=0.3,χ=0.4,κ=0.3时,各参数响应值最小,算法性能表现良好. 参数变化对算法的影响分析如下.

在H3DEDA中,ps越大,生成优质解的可能性越

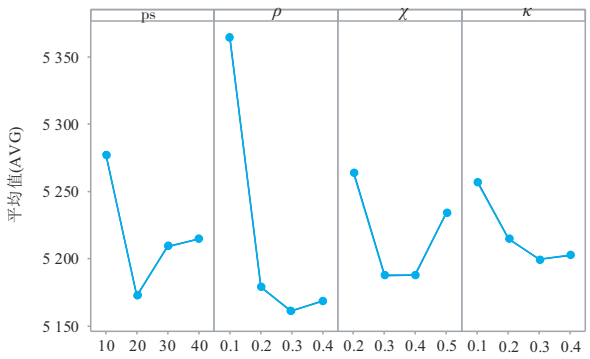


图6　各参数响应趋势图

大. 但当ps过大时,随着计算量增加,单次迭代消耗的时间也越多,而搜索次数会随之减少,找到优质解的可能性就会降低. 当ps过小时,则会导致种群多样性过低,算法容易过早陷入局部最优,影响搜索效率.

优质个体占比ρ在三维概率模型对优质种群的优质信息积累和更新过程中起到了重要作用. 当ρ值过大时,优质种群个体随之增加,会造成概率模型积累的块结构信息过于分散,导致无法将搜索放在有效区域内. 当ρ值过小时,会致使当前最优个体在概率模型更新中影响力过大,导致搜索过于集中,容易陷入局部最优.

学习参数χ主要影响概率模型的收敛速度. 当χ值过大,会导致概率模型过早收敛. 当χ值过小,会导致概率模型无法有效的积累优质个体的块结构信息.

κ为种群多样性阈值,是更新三维概率模型和启动扰动操作的重要因素,一个合理的κ值能够有效保持全局和局部搜索的搜索活力,降低陷入局部最优的可能性.

4.3　仿真结果比较和分析

4.3.1　验证算法改进环节有效性

H3DEDA主要的改进在于:(1)设计两阶段变邻域搜索机制;(2)提出三维概率模型自适应更新策略. 本文为验证以上改进的有效性,设计了2种改进算法:(1)H3DEDA_V1:在3DEDA基础上引入常规的变邻域搜索机制;(2)H3DEDA_V2:在3DEDA基础上引入两阶段变邻域搜索机制. 在相同的时间内对H3DEDA_V1、H3DEDA_V2和H3DEDA独立运行20次,比较结果如表7所示. 其中,比较占优的结果用粗体表示,AVG、BST和WST分别为算法独立运行20次输出的平均值、最优值和最差值. 由表7可知,H3DEDA_V2的性能相较于H3DEDA_V1有明显提升. 这表明两阶段变邻域搜索从工厂间和工厂内分别构造相应的邻域操作可引导算法在优质区域执行更为深入的局部搜索. 同时,H3DEDA在大部分算例上的测试结果均明显优于H3DEDA_V2. 这说明在H3DEDA中采用P_d来自适应控制概率模型的更新,可引导全局搜索到达解空间中更多的优质区域.

4.3.2　H3DEDA与其他算法的比较

为进一步验证H3DEDA的性能,将H3DEDA与HABCSA[8]、AHLBGA[12]以及EBSO[13]进行对比. 比较结果如表8所示,横向均值图如图7所示. 由图7和表8可知,H3DEDA明显优于其余三种算法. 这表明H3DEDA是求解ISP_DPAVD的有效算法. 其中,HABCSA分别针对雇佣蜂和观察蜂阶段设计结合模拟退火机制的变邻域搜索,有效的改进了初始解质量、拓宽搜索范围,但由于其种群个体间缺乏联系,导致优质结构信息无法有效的在种群中传播,收敛速度较慢. 同时对个体执行的邻域操作次数较少,存在较大的随机性,可能造成旧解还未在局部搜索中获得改进就被抛弃产生新解的情况,缺乏对解的深入搜索. AHLBGA将传统遗传算法和自适应学习机制相结合,利用交叉、变异和替换的方式生成新解,具有较好的全局搜索能力. 设计的自适应学习机制虽然能够提高算法对于优质结构信息的学习能力,但未能进一步改善遗传算法局部搜索能力较弱的缺点. 在EBSO中,新解是基于旧解而产生的,求解质量依赖于初始种群,优化结果稳定性较差,且局部搜索的邻域操作数量和搜索方式较为单一,对于解的改进有限. H3DEDA采用三维概率模型合理学习优质个体或解的块结构信息,可更好地引导全局搜索,同时利用多种邻域协同执行更加深入的局部搜索,故具有更强的搜索引擎,从而能获取更优的解.

同时,为能够直观比较4种算法在搜索过程中评价值的动态变化,将各算法在大规模算例[100×20×6]和小规模算例[50×10×4]上分别进行20次实验后作出图8中相应的曲线. 图8中的横坐标为算法运行时间(单位为秒),纵坐标为平均优化目标函数值. 每条曲线由对应算法在每隔2 s时的平均历史最优值(算法重复运行20次)连线形成[23]. 由图8可知,H3DEDA对应的曲线明显处于最下方. 这表明H3DEDA具有较强的搜索能力和较好的收敛性. 在其他算例上的结果与图8类似.

此外,为进一步验证H3DEDA与AHLBGA、HABCSA、EBSO间是否存在显著性差异,以表8实验结果的均值和最优值作为测试样本,按95%置信度对算法运行结果进行非参数配对样本检验[24]. 实验结果如表9所示. 若p值大于0.05,则差异不显著;若p值小于0.05,则差异显著. 由表9可知,H3DEDA与其余三种算

表 7 H3DEDA_V1、H3DEDA_V2、H3DEDA 对比结果

问题规模	H3DEDA_V1			H3DEDA_V2			H3DEDA		
	BST	WST	AVG	BST	WST	AVG	BST	WST	AVG
20×5×2	2 875	3 347	3 078	2 728	3 192	2 859	**2 728**	**2 970**	**2 791**
20×5×4	2 705	2 967	2 853	2 411	2 663	2 644	**2 361**	**2 663**	**2 485**
20×10×2	3 080	3 333	3 201	3 041	3 183	3 122	**3 020**	**3 120**	**3 074**
20×10×4	2 864	3 171	3 026	2 584	3 172	2 817	**2 584**	**2 668**	**2 623**
20×20×2	3 884	4 215	4 036	3 861	3 933	3 876	**3 837**	**3 888**	**3 867**
20×20×4	3 045	3 579	3 294	3 068	3 077	3 075	**2 754**	**2 882**	**2 825**
50×5×2	7 562	8 155	7 868	7 239	7 239	7 239	**6 980**	**7 239**	**7 155**
50×5×4	7 079	7 515	7 334	6 538	6 777	6 745	**6 404**	**6 777**	**6 636**
50×5×6	7 104	7 661	7 363	6 034	6 113	6 098	**6 026**	**6 113**	**6 098**
50×10×2	6 361	6 978	6 791	6 048	6 170	6 143	**5 775**	**6 170**	**6 060**
50×10×4	6 088	6 667	6 425	4 882	5 179	**5 054**	**4 850**	**5 179**	5 137
50×10×6	5 560	6 000	5 853	**4 747**	5 142	5 028	**4 816**	**5 142**	**5 021**
50×20×2	7 362	7 798	7 552	6 886	7 441	7 063	**6 544**	**7 195**	**6 929**
50×20×4	6 143	6 549	6 361	5 251	5 333	5 299	**5 210**	**5 333**	**5 295**
50×20×6	5 772	6 271	6 051	4 610	**4 684**	4 667	**4 595**	**4 729**	**4 716**
100×5×2	12 894	14 533	13 526	12 518	12 518	12 518	**11 827**	**12 518**	**12 299**
100×5×4	13 955	14 659	14 301	13 483	13 483	13 483	**12 002**	**13 196**	**12 704**
100×5×6	11 393	13 052	12 502	10 339	10 339	10 339	**10 271**	**10 339**	**10 334**
100×10×2	12 343	13 574	12 820	12 087	12 236	12 221	**11 014**	**11 879**	**11 506**
100×10×4	13 389	14 601	14 161	**12 304**	12 606	12 591	12 339	**12 606**	**12 560**
100×10×6	12 482	14 313	13 415	11 629	11 629	11 629	**11 384**	**11 629**	**11 611**
100×20×2	12 181	13 114	12 768	11 966	12 018	11 987	**11 186**	**11 795**	**11 470**
100×20×4	12 265	13 146	12 683	10 840	11 076	10 964	**10 670**	**11 076**	**10 950**
100×20×6	11 537	12 592	12 107	**9 640**	9 733	**9 681**	9 645	**9 733**	9 717
Average	7 913	8 658	8 307	7 281	7 456	7 381	**7 034**	**7 368**	**7 244**
NB	0	0	0	3	1	2	**22**	**24**	**23**

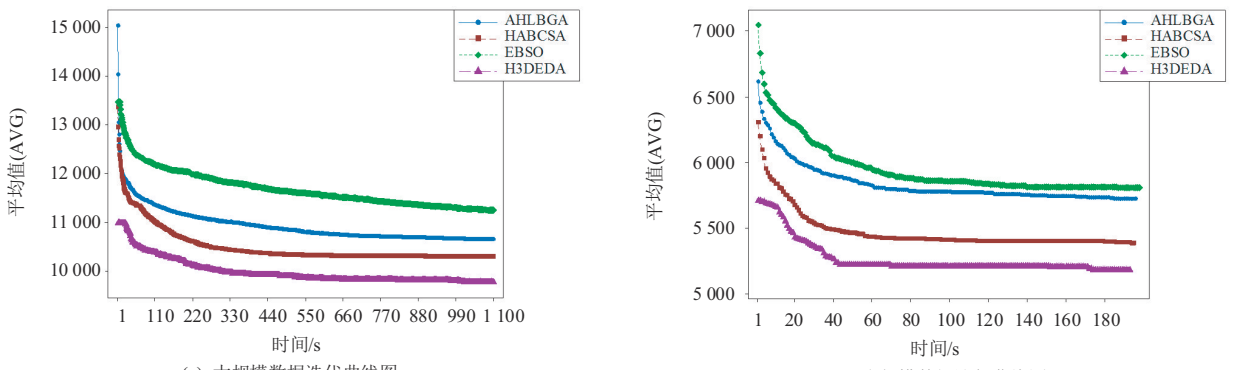
图 7 横向均值图

法的 p 值都小于 0.05. 这表明 H3DEDA 在统计意义下显著优于其余三种算法.

最后,值得指出的是,在各比较算法中,H3DEDA 能在复杂度较高(见 3.7 节)的情况下取得最好的结果. 这说明在相同时间下,虽然 H3DEDA 在解空间中搜索的总区域相对较少(即评价解的次数少),但实际搜索的优质区域更多,且在优质区域内搜索的更深. 因此,对于智能算法设计,如何结合问题特点,合理确定搜索方向来发现解空间中的优质区域,并在优质区域中尽量执行深度搜索,比简单增加搜索区域更为重要.

表 8　4 种算法求解 ISP_DPAVD 结果

问题规模	AHLBGA			HABCSA			EBSO			H3DEDA		
	BST	WST	AVG	BST	WST	AVG	BST	WST	AVG	BST	WST	AVG
20×5×2	2 853	3 026	2 935	2 813	3 411	3 048	2 809	5 370	3 314	**2 728**	**2 970**	**2 791**
20×5×4	2 370	2 834	2 626	2 435	2 857	2 720	2 362	2 994	2 631	**2 361**	**2 663**	**2 485**
20×10×2	3 140	3 449	3 242	3 064	3 508	3 177	3 080	3 650	3 248	**3 020**	**3 120**	**3 074**
20×10×4	2 606	3 053	2 869	2 735	3 222	2 926	2 711	3 516	3 003	**2 584**	**2 668**	**2 623**
20×20×2	4 494	4 711	4 594	3 875	5 310	4 234	3 884	5 849	4 087	**3 837**	**3 888**	**3 867**
20×20×4	2 944	3 260	3 138	2 913	3 350	3 125	2 807	3 077	2 877	**2 754**	**2 882**	**2 825**
50×5×2	**6 938**	7 600	7 332	8 042	8 985	8 423	7 248	8 789	7 853	**6 980**	**7 239**	**7 155**
50×5×4	6 374	6 865	6 669	6 936	7 800	7 324	6 387	7 465	6 766	**6 404**	**6 777**	**6 636**
50×5×6	5 985	6 798	6 409	6 638	7 714	7 255	5 978	7 268	6 386	**6 026**	**6 113**	**6 098**
50×10×2	5 890	6 454	6 207	6 365	6 994	6 663	5 886	7 484	6 208	**5 775**	**6 170**	**6 060**
50×10×4	5 255	5 899	5 636	5 740	6 631	6 050	5 159	6 677	5 648	**4 850**	**5 179**	**5 137**
50×10×6	4 945	5 310	5 132	5 303	5 800	5 510	4 852	5 787	5 151	**4 816**	**5 142**	**5 021**
50×20×2	6 771	7 332	7 153	7 074	8 014	7 444	6 551	7 844	7 121	**6 544**	**7 195**	**6 929**
50×20×4	5 404	5 785	5 559	5 630	6 394	5 938	5 094	6 295	5 362	**5 210**	**5 333**	**5 295**
50×20×6	4 886	5 458	5 144	5 056	5 731	5 398	4 666	5 540	5 021	**4 595**	**4 729**	**4 716**
100×5×2	**11 456**	**12 286**	**11 874**	13 872	15 739	14 949	12 883	14 536	13 893	11 827	12 518	12 299
100×5×4	**12 311**	**13 046**	**12 634**	14 929	15 811	15 264	14 340	16 607	15 289	12 002	13 196	12 704
100×5×6	10 608	11 238	10 945	12 361	13 634	13 103	11 632	13 225	12 405	**10 271**	**10 339**	**10 334**
100×10×2	11 052	11 903	**11 443**	12 860	13 925	13 377	12 097	14 512	13 254	**11 014**	**11 879**	11 506
100×10×4	12 387	13 035	12 675	14 197	15 122	14 571	13 466	15 028	14 124	**12 339**	**12 606**	**12 560**
100×10×6	11 604	12 206	11 926	13 403	14 284	13 782	11 919	13 632	12 580	**11 384**	**11 629**	**11 611**
100×20×2	**11 135**	11 846	11 537	12 452	13 955	13 056	12 035	14 804	12 890	**11 186**	11 795	11 470
100×20×4	11 098	11 993	11 507	12 255	13 466	12 838	11 163	12 637	11 695	**10 670**	**11 076**	**10 950**
100×20×6	10 118	11 178	10 592	11 159	12 326	11 662	9 923	11 848	10 753	**9 645**	**9 733**	**9 717**
Average	7 193	7 774	7 491	8 004	8 916	8 410	7456	8 935	7 982	**7 034**	**7 368**	**7 244**
NB	4	2	3	0	0	0	0	0	0	**21**	**23**	**22**



(*a*) 大规模数据迭代曲线图　　(*b*) 小规模数据迭代曲线图

图 8　AHLBGA、HABCSA、EBSO、H3DEDA 迭代过程曲线图

表9 统计结果分析

平均值配对检验	p值	最优值配对检验	p值
AHLBGA, H3DEDA	0.000 233	AHLBGA, H3DEDA	0.002
HABCSA, H3DEDA	0.000 001	HABCSA, H3DEDA	0.000 008
EBSO,H3DEDA	0.000 049	EBSO,H3DEDA	0.002

5 结论

本文在分布式装配流水车间调度问题的基础上，进一步考虑产品配送这一重要过程. 首次建立以最小化延迟惩罚和运输总成本为优化目标的分布式加工装配和车辆配送集成调度问题模型，并提出一种混合三维分布估计算法(H3DEDA)进行求解. H3DEDA具有以下创新：(1)针对ISP_DPAVD的特点，设计结合邻域变换的启发式规则来快速确定子问题2的解，可合理缩小搜索空间，从而能提高搜索效率；(2)为克服传统概率模型更新策略容易导致全局搜索过早收敛的不足，设计自适应更新策略来动态调控优质解信息的积累程度，有助于引导全局搜索到达解空间中更多的优质区域；(3)根据问题解的编码特征，设计基于厂内和厂外邻域操作的两阶段变邻域搜索，可驱动算法在优质区域内执行更深入的挖掘，同时利用块结构性质快速排除无效的厂内Swap邻域操作，能进一步提高挖掘效率. 仿真实验验证了H3DEDA能有效求解ISP_DPAVD. 后续的研究将把ISP_DPAVD扩展用于求解带装箱约束的分布式生成和车辆配送集成调度问题.

参考文献

[1] XIONG F L, XING K Y. Meta-heuristics for the distributed two-stage assembly scheduling problem with bi-criteria of makespan and mean completion time[J]. International Journal of Production Research, 2014, 52(9): 2743-2766.

[2] XIONG F L, XING K Y, WANG F, et al. Minimizing the total completion time in a distributed two stage assembly system with setup times[J]. Computers & Operations Research, 2014, 47: 92-105.

[3] ZHANG G H, XING K Y. Memetic social spider optimization algorithm for scheduling two-stage assembly flowshop in a distributed environment[J]. Computers & Industrial Engineering, 2018, 125: 423-433.

[4] LEI D M, SU B, LI M. Cooperated teaching-learning-based optimisation for distributed two-stage assembly flow shop scheduling[J]. International Journal of Production Research, 2021, 59(23): 7232-7245.

[5] 陈雅玲, 雷德明. 求解分布式两阶段装配流水车间调度的帝国竞争协作算法[J]. 控制理论与应用, 2021, 38(12): 1957-1967.
CHEN Y L, LEI D M. An imperialist competition and cooperation algorithm for distributed two-stage assembly flow shop scheduling[J]. Control Theory & Applications, 2021, 38(12): 1957-1967. (in Chinese)

[6] GHARAEI A, JOLAI F. A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain[J]. Applied Soft Computing, 2018, 65: 577-589.

[7] LI S, ZHONG X L, LI H, et al. Batch delivery scheduling with multiple decentralized manufacturers[J]. Mathematical Problems in Engineering, 2014, 2014: 321513.

[8] YıLMAZ Ö F, PARDALOS P M. Minimizing average lead time for the coordinated scheduling problem in a two-stage supply chain with multiple customers and multiple manufacturers[J]. Computers & Industrial Engineering, 2017, 114: 244-257.

[9] HAO J H, CAO L S, JIANG D K. Integrated production-distribution scheduling problem with multiple independent manufacturers[J]. Mathematical Problems in Engineering, 2015, 2015: 579893.

[10] ABDOLLAHZADEH V, NAKHAIKAMALABADI I, HAJIMOLANA S M, et al. A multifactory integrated production and distribution scheduling problem with parallel machines and immediate shipments solved by improved whale optimization algorithm[J]. Complexity, 2018, 2018: 5120640.

[11] MARANDI F, FATEMI GHOMI S M T. Integrated multi-factory production and distribution scheduling applying vehicle routing approach[J]. International Journal of Production Research, 2019, 57(3): 722-748.

[12] QIN H, LI T, TENG Y, et al. Integrated production and distribution scheduling in distributed hybrid flow shops [J]. Memetic Computing, 2021, 13(2): 185-202.

[13] HOU Y S, FU Y P, GAO K Z, et al. Modelling and optimization of integrated distributed flow shop scheduling and distribution problems with time windows[J]. Expert Systems with Applications, 2022, 187: 115827.

[14] FU Y P, HOU Y S, CHEN Z H, et al. Modelling and scheduling integration of distributed production and distribution problems via black widow optimization[J]. Swarm and Evolutionary Computation, 2022, 68: 101015.

[15] WANG S Y, WANG L. An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems,

2016, 46(1): 139-149.

[16] 周丰顺, 胡蓉, 钱斌, 等. 超启发式三维分布估计算法求解分布式流水线和车辆运输集成调度问题[J]. 电子学报, 2021, 49(12): 2419-2427.
ZHOU F S, HU R, QIAN B, et al. Hyper-heuristic three-dimensional estimation of distribution algorithm for solving distributed permutation flow-shop and vehicle transportation integrated scheduling problem[J]. Acta Electronica Sinica, 2021, 49(12): 2419-2427. (in Chinese)

[17] 郭晨, 曾思豪, 郭钧, 等. 混合分布估计算法求解模糊分布式装配柔性车间调度问题[J]. 系统工程理论与实践, 2021, 41(4): 1037-1048.
GUO C, ZENG S H, GUO J, et al. Hybrid estimation of distribution algorithm for distributed assembly flexible job shop scheduling problem with fuzzy processing time[J]. Systems Engineering-Theory & Practice, 2021, 41(4): 1037-1048. (in Chinese)

[18] ZHANG Z Q, QIAN B, HU R, et al. A matrix-cube-based estimation of distribution algorithm for the distributed assembly permutation flow-shop scheduling problem[J]. Swarm and Evolutionary Computation, 2021, 60: 100785.

[19] POTTS C N, SEVAST'JANOV S V, STRUSEVICH V A, et al. The two-stage assembly scheduling problem: Complexity and approximation[J]. Operations Research, 1995, 43(2): 346-355.

[20] AL-ANZI F S, ALLAHVERDI A. A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times[J]. European Journal of Operational Research, 2007, 182(1): 80-94.

[21] BASIR S A, MAZDEH M M, NAMAKSHENAS M. Bi-level genetic algorithms for a two-stage assembly flow-shop scheduling problem with batch delivery system[J]. Computers & Industrial Engineering, 2018, 126: 217-231.

[22] MONTGOMERY D C. Design and Analysis of Experiments[M]. 6th ed. Hoboken: John Wiley & Sons, 2005.

[23] 胡蓉, 陈文博, 钱斌, 等. 学习型蚁群算法求解绿色多车场车辆路径问题[J]. 系统仿真学报, 2021, 33(9): 2095-2108.
HU R, CHEN W B, QIAN B, et al. Learning ant colony algorithm for green multi-depot vehicle routing problem[J]. Journal of System Simulation, 2021, 33(9): 2095-2108. (in Chinese)

[24] 李正雯, 胡蓉, 钱斌, 金怀平, 吕阳. 学习型离散排超联赛算法求解带时间窗的绿色多车型两级车辆路径问题[J]. 控制理论与应用, 2023, 40(3): 549-557.
LI Z W, HU R, QIAN B, et al. A learning discrete volleyball premier league algorithm for solving green two-echelon heterogeneous-fleet vehicle routing problem with time windows[J] . Control Theory and Technology, 2023, 40 (3): 549-557. (in Chinese)

作者简介

杨绍文　男, 1995 年出生, 云南大理人, 硕士研究生, 目前研究方向为复杂系统智能优化.
E-mail: yswkust@163.com

钱　斌　男, 1976 年出生, 云南曲靖人, 教授, 博士生导师, 目前研究方向为智能调度理论与方法.
E-mail: bin.qian@vip.163.com

胡　蓉　女, 1973 年出生, 贵州安顺人, 副教授, 硕士生导师, 目前研究方向为优化方法与决策支持系统.
E-mail: ronghu@vip.163.com

张梓琪　男, 1989 年出生, 云南曲靖人, 博士研究生, 目前研究方向为智能算法与优化调度.
E-mail: 768894018@qq.com