

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

周丰顺^{1,2}, 胡 蓉^{1,2}, 钱 斌^{1,2}, 张长胜¹, 向凤红¹

(1. 昆明理工大学信息工程与自动化学院, 云南昆明 650500; 2. 昆明理工大学云南省人工智能重点实验室, 云南昆明 650500)

摘 要: 针对一类广泛存在的分布式流水线和车辆运输集成调度问题(Distributed Permutation Flow-shop and Vehicle Transportation Integrated Scheduling Problem, DPFVTISP), 本文建立问题模型, 并提出一种超启发式三维分布估计算法(Hyper-Heuristic three-Dimensional Estimation of Distribution Algorithm, HH3DEDA) 进行求解. 首先, 根据 DPFVTISP 的问题特性, 采用贪婪策略设计一种新颖的编解码规则. 其次, 为实现对 DPFVTISP 问题解空间中不同区域的深入搜索, 设计 10 种低层启发式操作 (即 10 种有效的邻域操作), 并将其所构成的排列作为高层个体; 同时在高层采用三维分布估计算法 (three-Dimensional Estimation of Distribution Algorithm, 3DEDA) 学习和积累优质高层个体中块结构及其位置信息, 再通过采样 3DEDA 中的概率模型生成新的高层个体或排列; 进而在低层将高层所生成的每个新个体所表征的一系列有序的启发式操作作为一种新的启发式算法, 对问题解空间执行较深入的邻域搜索. 最后, 通过仿真实验与算法对比验证 HH3DEDA 可有效求解 DPFVTISP.

关键词: 分布式流水线; 车辆运输; 集成调度; 超启发式算法; 三维分布估计算法; 块结构

中图分类号: TP273

文献标识码: A

文章编号: 0372-2112(2021)12-2419-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20201057

Hyper-Heuristic Three-Dimensional Estimation of Distribution Algorithm for Solving Distributed Permutation Flow-Shop and Vehicle Transportation Integrated Scheduling Problem

ZHOU Feng-shun^{1,2}, HU Rong^{1,2}, QIAN Bin^{1,2}, ZHANG Chang-sheng¹, XIANG Feng-hong¹

(1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan 650500, China;

2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, Yunnan 650500, China)

Abstract: Aiming at a kind of widely existing distributed permutation flow-shop and vehicle transportation integrated scheduling problem(DPFVTISP), this paper establishes the problem model and proposes a hyper-heuristic three-dimensional estimation of distribution algorithm(HH3DEDA) to solve it. Firstly, a novel coding and decoding rules adopting the greedy strategy is designed via analyzing the characteristics of DPFVTISP. Secondly, in order to search different regions in the solution space of DPFVTISP, ten kinds of low-layer heuristic operations, i.e., ten kinds of effective neighborhood operations, are designed, and their permutations are regarded as high-layer individuals. At the same time, the three-dimensional estimation of distribution algorithm (3DEDA) is used in the high layer to learn and accumulate the information of block structures and their positions in the high-quality high-layer individuals, and then new low-layer individuals or permutations are generated by sampling the probability model in 3DEDA. Furthermore, in the low layer, an ordered series of heuristic operations presenting by each new individual generated in the high layer is regarded as a new heuristic algorithm to execute further neighborhood search in the problem's solution space. Finally, simulations and comparisons demonstrate that HH3DEDA can effectively solve the DPFVTISP.

Key words: distributed permutation flow-shop; vehicle transportation; integrated scheduling; hyper-heuristic algorithm; three-dimensional estimation of distribution algorithm; block structure

1 引言

随着市场竞争的加剧以及经济全球化趋势的深入发展,生产制造活动由原来的集中式向分布式进行转变^[1].此外,由分布式工厂加工的工件往往需要通过车辆运输至指定集合点,以便日后统一分销.在上述背景下,分布式流水线和车辆运输集成调度问题(Distributed Permutation Flow-shop and Vehicle Transportation Integrated Scheduling Problem, DPFVTISP)的重要性日益凸显.目前,对于生产和运输集成调度问题的研究主要包括如下三类.第一类为单工厂生产与车辆配送集成调度问题. Moons等^[2]对2017年以前的集成生产调度和车辆配送问题的文献进行了详细综述,从中可知已有研究大都集中于单工厂生产与车辆配送集成调度问题.近三年来,该类问题仍得到较多研究^[3,4].第二类为多工厂并行生产与车辆配送集成调度问题^[5-7].第三类为多级多工厂生产与车辆配送集成调度问题^[8,9].从上述文献调研可知,对于实际生产中广泛存在的一类问题,即多工厂各自进行产品生产且各工厂产品由单个集合点安排车辆运输回集合点的集成调度问题(DPFVTISP),目前尚无相关研究.因此,本文建立 DPFVTISP 模型,并设计有效算法对其进行求解.

超启发式(Hyper-Heuristic, HH)算法是一类新型智能优化算法.该算法通过某种高层策略(High-Level Strategy, HLS)操纵或管理低层启发式算法(Low-Level Heuristics, LLHs),并以此实现对解空间不同区域的搜索^[10].近年来超启发式算法已经成功被应用于求解各类生产调度问题^[11,12].HH算法的高层解空间与大峡谷地形类似,其巨大而扁平的解空间内存在大量深浅不一的山谷(即局部最优解),同时最优解的附近分布着许多优质解,这表明各优质解之间存在相似的结构信息.显然,合理利用这些结构信息可引导算法快速到达山谷底部附近执行搜索,进而有望提高算法效率.从HH算法在求解调度问题上的研究现状来看,高层采用有效的策略作为HLS,同时底层结合问题特性构造丰富且合理的LLHs,是设计高效HH算法的关键.

分布估计算法(Estimation of Distribution Algorithm, EDA)是一种基于统计学习的智能算法^[13,14].近年来基于EDA的算法已经成功被应用于求解各类生产调度问题^[15-18].然而,现有基于EDA的算法均采用二维概率模型,这使其只能积累优质解中相似块的序信息,但无法积累相似块的位置信息.因此,本文设计基于三维概率模型的三维分布估计算法(three-Dimensional Estimation of Distribution Algorithm, 3DEDA)作为超启发式算法的高层策略,用于同时学习和积累高层优质解的相似块序信息和位置信息,继而在采样生成新

个体时能较准确地确定优质块结构在解中的具体位置,从而可更合理地引导算法搜索方向以提高搜索效率.

综上,本文研究 DPFVTISP 的建模与求解,在建模方面,针对实际生产活动中各分布式工厂内部存在复杂加工过程,且各工厂与统一集合点之间存在较复杂的运输过程,首次建立以最小化加工成本与运输成本的加权和为优化目标的 DPFVTISP 模型.在求解方面,考虑到 DPFVTISP 这类问题解空间巨大且复杂,常规智能算法难以实现有效搜索,故设计一种超启发式三维分布估计算法(Hyper-Heuristic three-Dimensional Estimation of Distribution Algorithm, HH3DEDA)进行求解.首先,在算法的编解码阶段,设计以车辆为基本单位对包含工厂信息的工件进行多段编码,同时根据问题结构特点,设计以最小化两阶段总完工时间为目标的解码方式.然后,HH3DEDA 高层采用基于三维概率模型的 3DEDA 学习和积累高层策略域中较优个体的排列信息,进而通过特定的采样方式对该模型进行采样以生成新的邻域操作排列,进而在低层把高层所生成的邻域操作排列作为一种启发式算法以实现对问题解空间进行搜索.最后,通过仿真实验和算法比较验证了 HH3DEDA 的有效性.

2 分布流水线和车辆运输集成调度问题

2.1 符号定义

本文所涉及的有关数学符号定义如表1所示.

表1 符号表

符号	说明
j	工件编号. $j=\{1, 2, \dots, N_j\}$.
i	机器编号. $i=\{1, 2, \dots, M\}$.
$p_{j,i}$	工件 j 在机器 i 上的加工时间.
$W(j)$	工件 j 的重量.
k, f	工厂和集合点编号 $k, f=\{0, 1, 2, \dots, F\}$, 集合点编号为0.
π^k	工厂 k 中工件加工序列. $\pi^k=\{\pi^k(l) l=1, 2, \dots, N^k\}$.
$C_{j,i}$	工件 j 在机器 i 上的完工时间.
h	车辆编号. $h=\{1, 2, \dots, H\}$.
Q	车辆的最大载重.
$D_{k,f}$	从工厂或集合点 k 到工厂或集合点 f 之间的距离.
V	车辆行驶的速度.
$S_{h,k}$	车辆 h 从工厂或集合点 k 出发时间.
$A_{h,k}$	车辆 h 到达工厂或集合点 k 的时间.
π_w^h	车辆 h 所运载的工件序列. $\pi_w^h=\{\pi_w^h(a) a=1, 2, \dots, N_w^h\}$.
π_t^h	车辆 h 行驶路径点序列. $\pi_t^h=\{\pi_t^h(b) b=1, 2, \dots, N_t^h\}$.
$C_{st1}(j)$	工件 j 在加工阶段的完工时间.
$C_{st2}(j)$	工件 j 运输至集合点的时间.

2.2 问题模型

DPFVTISP可描述为:将 N_j 个工件分配给 F 个具有相同配置但处于不同地理位置的工厂进行加工,工件加工完成后通过车辆运输至指定集合点. DPFVTISP示意图如图1所示,整个过程分为两阶段:生产阶段和运输阶段. 在生产阶段,各工厂的配置相同,均为流水线加工过程,所有工件均可安排在任意工厂进行加工,工件一旦被分配到某个工厂后就不能再被分配到其他工厂,每个工厂包括 M 台机器,工件需按照固定的机器加工顺序进行加工. 任意一台机器在同一时刻只能加工一个工件,不同工件之间相互独立互不影响. 在运输阶段,车辆行驶速度恒定, H 辆具有载重约束的车辆从集合点出发,每辆车按照计划将不同工厂中完工的工件运回集合点. 车辆在不违反载重约束前提下可访问多个工厂. 车辆运输路径示意图如图2所示,运输模式为带容量约束的车辆路径问题(Capacitated Vehicle Routing Problem, CVRP). 其中,车辆1的运输路径为:集合点→工厂2→工厂4→工厂5→集合点,车辆2的运输路径为:集合点→工厂1→工厂3→集合点,车辆3的运输路径为:集合点→工厂3→工厂5→工厂4→集合点.

生产阶段工件 $\pi^k(l)$ 的完工时间 $C_{st1}(\pi^k(l))$ 计算如下:

$$C_{\pi^k(0),1} = p_{\pi^k(0),1}, k = 1, 2, \dots, N_k \quad (1)$$

$$C_{\pi^k(l),1} = C_{\pi^k(l-1),1} + p_{\pi^k(l),1}, k = 1, 2, \dots, N_k, l = 1, 2, \dots, N^k \quad (2)$$

$$C_{\pi^k(0),i} = C_{\pi^k(0),i-1} + p_{\pi^k(0),i}, k = 1, 2, \dots, N_k, i = 1, 2, \dots, N_i \quad (3)$$

$$C_{\pi^k(l),i} = \max \{C_{\pi^k(l-1),i}, C_{\pi^k(l),i-1}\} + p_{\pi^k(l),i}, k = 1, 2, \dots, N_k, l = 1, 2, \dots, N^k, i = 1, 2, \dots, N_i \quad (4)$$

$$C_{st1}(\pi^k(l)) = C_{\pi^k(l),N_i}, k = 1, 2, \dots, N_k, l = 1, 2, \dots, N^k \quad (5)$$

运输阶段车辆载重约束和工件 $\pi_w^h(a)$ 到达集合点的时间计算如下:

$$\sum_{a=1}^{N_w^h} W(\pi_w^h(a)) \leq Q, h = 1, 2, \dots, N_h \quad (6)$$

$$S_{h,0} = \max \{0, \max \{C_{st1}(\pi_w^{h(1)}(\pi_w^h(a))) - D_{\pi_r^h(0),\pi_r^h(1)} / V\}, h = 1, 2, \dots, N_h, a = 1, 2, \dots, N_w^h \quad (7)$$

$$A_{h,\pi_r^h(b)} = S_{h,\pi_r^h(b-1)} + D_{\pi_r^h(b-1),\pi_r^h(b)} / V, h = 1, 2, \dots, N_h, b = 1, 2, \dots, N_r^h \quad (8)$$

$$S_{h,\pi_r^h(b)} = \max \{A_{h,\pi_r^h(b)}, \max \{C_{st1}(\pi_w^{h(b)}(\pi_w^h(a)))\}\}, h = 1, 2, \dots, N_h, a = 1, 2, \dots, N_w^h, b = 1, 2, \dots, N_r^h \quad (9)$$

$$C_{st2}(\pi_w^h(a)) = A_{h,\pi_r^h(N_r^h)}, h = 1, 2, \dots, N_h, a = 1, 2, \dots, N_w^h \quad (10)$$

DPFVTISP的加工阶段成本 P ,运输阶段成本 D ,以及总成本 T 计算如下:

$$P = \alpha \times \max \{C_{\pi^k(N^k),N_i}, k = 1, 2, \dots, N_k \quad (11)$$

$$D = \beta \times N_h + \delta \times \sum_{h=1}^{N_h} \sum_{b=0}^{N_r^h-1} D_{b,b+1}, h = 1, 2, \dots, N_h \quad (12)$$

$$T = P + D \quad (13)$$

DPFVTISP的优化目标为在加工、运输两阶段的集成调度过程中找到一个最优排序 π^* 使得总成本 T 最小.

3 超启发式三维分布估计算法

3.1 编码与解码

对于高层策略域,种群中的每个个体 A 均由10种低层启发式操作 L_c 排列构成. 解码高层策略域个体时,对低层问题域中的解,从左到右依次执行高层策略域个体中的低层启发式操作. 每执行完一次启发式操作,就将得到的新解与旧解进行对比,若新解优于旧解,则用新解替换旧解,否则,保留旧解,然后继续执行剩余低层启发式操作. 执行完高层策略域个体中的所有启发式操作后,该高层个体的适应值即为其对应更新的低层问题域个体的适应值. 图3为一个高层策略域个体的示意图.

对于低层问题域,每个个体就是原问题的一个

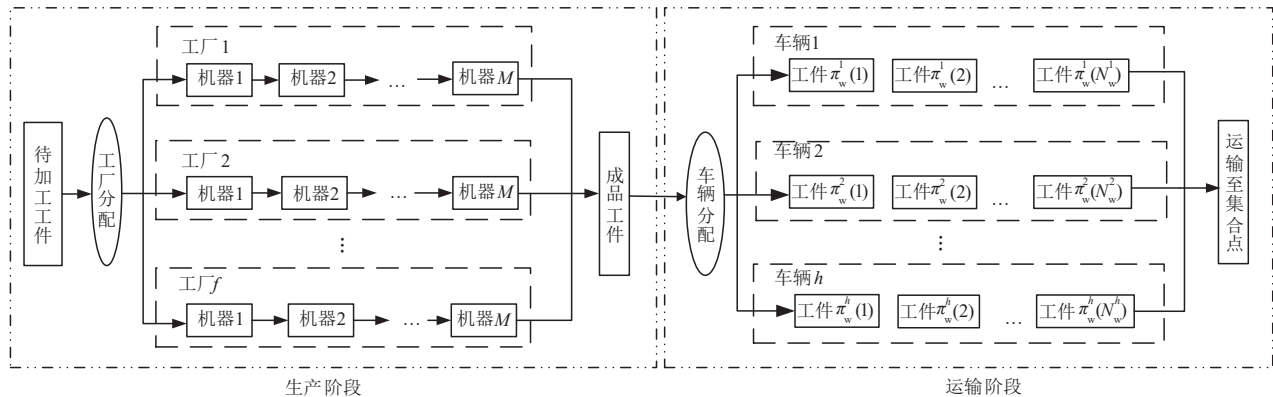


图1 DPFVTISP示意图

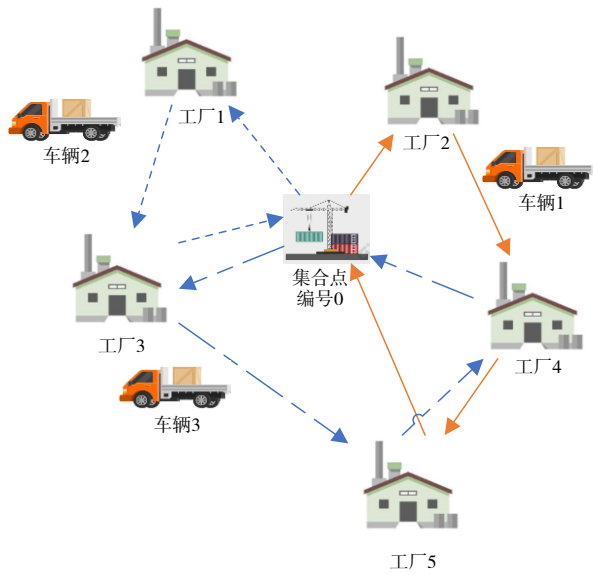


图2 运输阶段车辆路径示意图



图3 高层策略域个体示意图

解. 低层问题域编码个体 π 为工件加工序列. 此外, 本文针对 DPVFTISP 设计了一种新颖的解码方式, 该解码方式首先依据载重约束将工件分配给各车辆, 然后以车辆最早到达集合点 (即最小化两阶段总完工时间) 为目标对工件进行工厂分配. 解码后的个体由包含车辆装载信息的工件加工序列 π_j 和包含车辆路径信息的工厂分配序列 π_r 组成. 具体解码步骤如下:

步骤1 若当前已分配至车辆的工件数 $\zeta=0$, 则将 $\pi(1)$ 置于车辆编号 $h=1$ 运载序列 π_w^h 中的第 1 位, 令 $\zeta=\zeta+1$, 车辆 1 所运载工件数 $\theta_1=1$; 若 $\zeta \neq 0$ 且 $\zeta \neq N$ 时, 则转至步骤 2; 若 $\zeta=N$ 时, 则转至步骤 3.

步骤2 若 $\sum_{d=1}^{\theta_h} W(\pi(d)) < Q$ 且 $\sum_{d=1}^{\theta_h+1} W(\pi(d)) < Q$, 则将 $\pi(\zeta+1)$ 置于 π_w^h 中的第 θ_h+1 位, 令 $\zeta=\zeta+1$, $\theta_h=\theta_h+1$; 若 $\sum_{d=1}^{\theta_h} W(\pi(d)) < Q$ 且 $\sum_{d=1}^{\theta_h+1} W(\pi(d)) > Q$, 则将 $\pi(\zeta+1)$ 置于车辆序列 π_w^{h+1} 中的第 1 位, 令 $h=h+1$, $\zeta=\zeta+1$, $\theta_h=1$. 转至步骤 1.

步骤3 若车辆 $h=1$ 时, 车辆 h 已分配至工厂的工件数 ω_h , 工厂 k (当前 $k=1$) 已加工工件数为 α_k , 则将 $\pi_w^h(\omega_h+1)$ 置于工厂 k 加工序列 π^k 中的第 α_k+1 位, 按照式 (1)~(10) 计算得到 $\pi_w^h(\omega_h+1)$ 到达集合点的时间 $C_{st2}(\pi_w^h(\omega_h+1))_k$ 后, 转至步骤 4. 若车辆 $h \neq 1$, 且当前车辆所运载工件数 $\theta_h \neq 0$, 则将 $\pi_w^h(\omega_h+1)$ 置于工厂 k

(当前 $k=1$) 的加工序列 π^k 中的第 α_k+1 位, 按照式 (1)~(10) 计算得到 $\pi_w^h(\omega_h+1)$ 到达集合点的时间 $C_{st2}(\pi_w^h(\omega_h+1))_k$ 后, 转至步骤 4; 若车辆 $h \neq 1$ 且 $\theta_h=0$ (即所有车辆内工件已分配完毕), 则转至步骤 6.

步骤4 令 $k=k+1$, 将 $\pi_w^h(\omega_h+1)$ 置于工厂 k 加工序列 π^k 中的第 α_k+1 位, 按照式 (1)~(10) 计算得到 $\pi_w^h(\omega_h+1)$ 到达集合点的时间 $C_{st2}(\pi_w^h(\omega_h+1))_k$. 若 $k < F+1$ 则转至步骤 4; 否则, 转至步骤 5.

步骤5 选择工厂 k 能使 $\arg \{C_{st2}(\pi_w^h(\omega_h))_k\} \rightarrow \min$, 令 $\alpha_k=\alpha_k+1$, $\omega_h=\omega_h+1$, 若 $\omega_h \geq \theta_h$, 则 $h=h+1$, 转至步骤 3; 若 $\omega_h < \theta_h$, 则直接转至步骤 3.

步骤6 在相邻两车辆运载工件序列之间插入 0 得到解码后的工件序列 $\pi_j=[0, \pi_w^1, 0, \pi_w^2, 0, \dots, 0, \pi_w^h, 0]$, 在相邻两车辆运输路径点之间插入 0 得到解码后的工厂分配序列 $\pi_r=[0, \pi_r^1, 0, \pi_r^2, 0, \dots, 0, \pi_r^h, 0]$.

以规模为 $N=8, F=5$ 的问题为例, 若工件序列 $\pi_j=[2, 1, 5, 3, 7, 4, 6, 8]$, 经由上述步骤解码完成所得低层个体的示意图如图 4 所示. 由图 4 可知, 车辆 1 所运载工件序列为 $\pi_w^1=[2, 1, 5]$, 车辆 2 所运载的工件序列为 $\pi_w^2=[3, 7]$, 车辆 3 所运载的工件序列为 $\pi_w^3=[4, 6, 8]$; 车辆 1 的运输路径 $\pi_r^1=[0, 2, 4, 5, 0]$, 车辆 2 的运输路径 $\pi_r^2=[0, 1, 3, 0]$, 车辆 3 的运输路径 $\pi_r^3=[0, 3, 5, 4, 0]$. 其中 0 为集合点的编号; 工厂 1 中工件加工序 $\pi^1=[3]$, 工厂 2 中工件加工序 $\pi^2=[2]$, 工厂 3 中工件加工序 $\pi^3=[7, 4]$, 工厂 4 中工件加工序 $\pi^4=[1, 8]$, 工厂 5 中加工序 $\pi^5=[6, 5]$.

0	2	1	5	0	3	7	0	4	6	8	0
$\pi_r(1)$	$\pi_r(2)$	$\pi_r(3)$	$\pi_r(4)$	$\pi_r(5)$	$\pi_r(6)$	$\pi_r(7)$	$\pi_r(8)$	$\pi_r(9)$	$\pi_r(10)$	$\pi_r(11)$	$\pi_r(12)$
0	2	4	5	0	1	3	0	3	5	4	0
$\pi_r(1)$	$\pi_r(2)$	$\pi_r(3)$	$\pi_r(4)$	$\pi_r(5)$	$\pi_r(6)$	$\pi_r(7)$	$\pi_r(8)$	$\pi_r(9)$	$\pi_r(10)$	$\pi_r(11)$	$\pi_r(12)$

图4 解码后低层问题域个体示意图

3.2 三维分布估计算法

3.2.1 块结构与相似块

对于高层策略域而言, 将整个序列中连续相邻的两个启发式操作视为一个块结构. 不同位置的块结构被定义为相似块. 显然, 高层策略域中个体由一系列相互作用的相似块或位于不同位置的块结构构成.

3.2.2 三维矩阵模型

定义 P_G 为 3DEDA 中第 G 代的高层策略域种群, P_G^H 为 P_G 中的优质解, S_p 为 P_G 的规模, S_{HP} 为 P_G^H 的规模, $A_H^{G,k}$ 为 P_G^H 的第 k 个个体. 定义三维矩阵 $M_{n \times n \times n}^G$ 用于储存第 G 代优秀个体中相似块及其分布等信息, 令 $M_{n \times n \times n}^G(x, y, z)$ 为 $M_{n \times n \times n}^G$ 中的元素, $M_{n \times n \times n}^G(x, y, z)$ 的具体描述如下所示:

$$F_{n \times n \times n}^{G,k}(x,y,z) = \begin{cases} 1, & \text{若 } y=A_H^{G,k}(x) \text{ 和 } z=A_H^{G,k}(x+1), \\ 0, & \text{否则,} \end{cases} \quad (14)$$

$$M_{n \times n \times n}^G(x,y,z) = \sum_{k=1}^{sps} F_{n \times n \times n}^{G,k}(x,y,z), \quad (15)$$

$$x=1, \dots, n-1, y, z=1, \dots, n, k=1, \dots, S_{HP}$$

其中,式(14)中 $F_{n \times n \times n}^{G,k}(x,y,z)$ 用于记录 P_G^H 中第 k 个个体中启发式操作的顺序和相似块的位置;式(15)用于记录高层策略域中所有优质解的启发式操作的顺序和相似块的位置信息, $M_{n \times n \times n}^G(x,y,z)$ 为在 P_G^H 中块结构 $[y,z]$ 在 x 位置上出现的次数。

3.2.3 三维概率模型

HH3DEDA 使用基于 $M_{n \times n \times n}^G$ 的三维概率模型 $N_{n \times n \times n}^G$ 来学习和积累高层策略域优质解中相似块的位置信息,令 $N_{n \times n \times n}^G(x,y,z)$ 为 $N_{n \times n \times n}^G$ 中的元素。

高层策略域中所有优质解中相似块分布在 x 位置上的概率和为:

$$N_{Sum}^G(x) = \sum_{y=1}^n \sum_{z=1}^n N_{n \times n \times n}^G(x,y,z) \quad (16)$$

高层策略域中所有优质解中相似块分布在 x 位置上的次数为:

$$M_{Sum}^G(x) = \sum_{y=1}^n \sum_{z=1}^n M_{n \times n \times n}^G(x,y,z) \quad (17)$$

概率模型的更新过程如下:

步骤1 按式(18)对概率模型进行初始化。

$$N_{n \times n \times n}^0(x,y,z) = \begin{cases} 1/n, & x=1, y, z=1, \dots, n \\ 1/n^2, & x=2, 3, \dots, n-1, y, z=1, \dots, n \end{cases} \quad (18)$$

步骤2 按式(19)计算三维矩阵 $M_{n \times n \times n}^0$ 并更新概率模型 $N_{n \times n \times n}^1$ 。

$$N_{n \times n \times n}^1(x,y,z) = \begin{cases} M_{n \times n \times n}^0(x,y,z)/M_{Sum}^0(x), & x=1, y, z=1, \dots, n, \\ (N_{n \times n \times n}^0(x,y,z) + M_{n \times n \times n}^0(x,y,z)) / (N_{Sum}^0(x) + M_{Sum}^0(x)), & x=2, 3, \dots, n-1, y, z=1, \dots, n. \end{cases} \quad (19)$$

步骤3 将 G 设为1。

步骤4 按式(20)计算 $M_{n \times n \times n}^G$ 并更新概率模型 $N_{n \times n \times n}^G$

$$N_{n \times n \times n}^G(x,y,z) = (1-r) \times N_{n \times n \times n}^{G-1}(x,y,z) + r \times M_{n \times n \times n}^{G-1}(x,y,z) / M_{Sum}^{G-1}(x), \quad (20)$$

$$x=1, 2, \dots, n-1, y, z=1, \dots, n$$

步骤5 将 $G=G+1$,若 $G < \max G$ 则转到步骤4;否则,终止循环。

3.3 种群初始化

本文按式(18)对初始三维概率模型采样形成高层策略域的初始种群,而对于低层问题域,为了避免解的分布过于集中,导致后续操作无法对解空间进行

充分搜索,采用随机初始化的方式生成低层问题域种群。

3.4 高层策略域种群采样更新

对于高层策略域种群,定义 $A^{G,k}$ 为 P_G 中的第 k 个个体。 $C_{opt}(A^{G,k},i)$ 为操作选择函数,用于确定在 $A^{G,k}$ 中第 i 个位置出现的低层启发式操作。由于相似块 $[A^{G,k}(i-1), A^{G,k}(i)]$ 被选中的概率存储在三维概率模型 $N_{n \times n \times n}^{G-1}(i-1)$ 中,故 $C_{opt}(A^{G,k},i)$ 依据三维概率模型 $N_{n \times n \times n}^{G-1}(i-1)$ 进行采样, $C_{opt}(A^{G,k},i)$ 的具体过程如下:

步骤1 随机生成概率数 $r, r \in [0, \sum_{h=1}^n N_{n \times n \times n}^{G-1}(i-1, A^{G,k}(i-1), h)]$ 。

步骤2 用轮盘赌的方法选择 L_c ,若 $r \in [0, N_{n \times n \times n}^{G-1}(i-1, A^{G,k}(i-1), 1)]$ 则 $c=1$,转至步骤3,若 $r \in [\sum_{h=1}^{pos} N_{n \times n \times n}^{G-1}(i-1, A^{G,k}(i-1), h),$

$\sum_{h=1}^{pos+1} N_{n \times n \times n}^{G-1}(i-1, A^{G,k}(i-1), h)]$ 则 $c=pos+1$ 。

$pos \in \{1, \dots, n-1\}$

步骤3 返回 L_c 。

对于高层策略域个体在位置1处,不存在位置0与位置1组成的相似块,进而 $C_{opt}(A^{G,k},i)$ 在初始位置1无法进行采样,因此,本文针对初始位置提出了一种初始采样策略,具体描述如下:

步骤1 按式(21)计算 $N_{init}^{G-1}(y)$ 。

$$N_{init}^{G-1}(y) = \sum_{z=1}^n N_{n \times n \times n}^{G-1}(1, y, z), y=1, \dots, n \quad (21)$$

步骤2 采用轮盘赌方法确定 $A^{G,k}, k=1, \dots, S_p$ 中初始位置1的低层启发式操作 L_c ,随机生成概率数 r' , $r' \in [0, \sum_{h=1}^n N_{init}^{G-1}(h)]$,若 $r' \in [0, N_{init}^{G-1}(1)]$ 则 $c=1$,转至步骤3,若 $r' \in [\sum_{h=1}^{pos} N_{init}^{G-1}(h), \sum_{h=1}^{pos+1} N_{init}^{G-1}(h)]$, $pos \in \{1, \dots, n-1\}$ 则 $c=pos+1$ 。

步骤3 令 $A^{G,k}(1)$ (其中 $k=1, \dots, S_p$)为 L_c 。

基于上述描述, P_G 更新过程如下:

步骤1 令 $k=1$ 。

步骤2 根据初始采样策略确定 $A^{G,k}$ 初始位置1的低层启发式操作 L_c 。

步骤3 令 $i=2$ 。

步骤4 令 $A^{G,k}(i) = C_{opt}(A^{G,k}, i), i=i+1$ 。

步骤5 令 $i=i+1$,若 $i \leq n$ 则转至步骤4。

步骤6 令 $k=k+1$,若 $k \leq S_p$ 则转至步骤2。

步骤7 输出 P_G 。

3.5 低层启发式操作

为能对HH3DEDA全局搜索发现优质解区域进行有效搜索,本文在算法低层设计如下10种有效的邻域的操作,并通过不同的排列构成多种启发式算法,采用多种启发式算法进行搜索可在搜索到达多种邻域结

构下的局部最优解前一直持续有效深入搜索。

L_1 : 同一辆车内, 加工序列交叉操作. 随机选择一辆车, 再从该车运载工件的加工序列中随机选择两个工件进行交换。

L_2 : 同一辆车内, 加工序列前项插入操作. 随机选择一辆车, 再从该车运载工件的加工序列中随机选择两位, 将位置靠后的工件插入到位置靠前的工件之前。

L_3 : 同一辆车内, 加工序列相邻工件交叉操作. 随机选择一辆车, 再从该车运载工件的加工序列中随机选择一个工件, 将它与向前一位或向后一位工件交换位置。

L_4 : 同一辆车内, 工厂分配序列交叉操作. 随机选择一辆车, 再从该车运载工件的工厂分配序列中随机选择两位, 交换它们的工厂分配。

L_5 : 同一辆车内, 工厂分配序列突变操作. 随机选择一辆车, 再从该车运载工件的工厂分配序列中随机选择两位, 并将它们随机突变成其他工厂。

L_6 : 加工序列交叉操作. 从加工序列中随机选择两位工件, 交换它们的位置。

L_7 : 加工序列前项插入操作. 从加工序列中随机选择两位工件, 将位置靠后的工件插入到位置靠前的工件之前。

L_8 : 加工序列相邻工件交叉操作. 从加工序列中随机选择一个工件, 将它与向前一位或向后一位工件交换位置。

L_9 : 工厂分配序列突变操作. 从工厂分配序列中随机选择两位, 并将它们随机突变成另一工厂。

L_{10} : 学习优质序列操作. 从加工序列中截取一段, 将其替换为优质解中同样位置的加工序列。

3.6 算法流程

根据上述算法描述, 整个算法具体流程如下:

步骤 1 初始化低层问题域种群, 种群大小为 S_p ; 初始化三维概率模型。

步骤 2 采样产生高层策略域种群。

步骤 3 评价低层问题域种群。

步骤 4 解码高层策略域种群中所有个体。

步骤 5 选择高层策略域当前种群 $S_p \times \rho$ 个精英个体更新三维分布估计算法概率模型, 并通过采样生成下一代新种群。

步骤 6 判断是否满足终止条件. 若不满足则转到步骤 3; 否则, 终止循环。

4 实验设计与分析

由于目前没有合适的 DPFVTISP 的标准算例, 本文所有测试算例均在 DPFSP 标准测试算例的基础上生

成. 其所有算法和实验均由 Delphi2010 编程实现, 操作系统为 Window10, CPU 为 2.90GHz, 内存为 16GB. 每种算法对每个测试问题均在相同时间下独立运行 31 次. 其中, AVG 为算法独立运行 31 次输出最优结果的平均值。

4.1 参数设置

HH3DEDA 的关键参数为种群规模 S_p 、学习率 γ 、精英个体所占比例 ρ . 本节对问题规模为 $100 \times 10 \times 4$ 的算例采用实验设计方法 (Design Of Experiment, DOE)^[19] 进行参数实验分析*, 进而确定 HH3DEDA 的参数取值. 经实验分析, 最佳的参数组合为 $S_p = 20$ 、 $\gamma = 0.35$ 、 $\rho = 0.55$ 。

4.2 仿真结果与比较

对于不同规模问题, 设定每种算法的运行时间均为 $20 \times N \times M \times Fms$. 所有测试问题均进行 31 次独立实验. 其中, Average 为所有规模问题通过相关算法获得的每个性能指标输出结果的平均值, NB 为所有规模问题通过相关算法获得的每个性能指标最优值的总数, IMP 为以 HHGA 的 AVG 为基准, 当前算法的 AVG 改进的百分比值. 以 AVG 为性能指标的实验结果如表 2、表 3 所示, 其余指标的完整实验结果请见上传的网络附件^①。

4.2.1 验证高层策略的有效性

为验证 HH3DEDA 中选择 3DEDA 作为高层策略的有效性, 将 HH3DEDA 与分别采用 GA 与 EDA 作为高层策略 HHGA、HHEDA 进行比较. 这三种算法除高层策略不同外, 其他部分均相同, 比较结果如表 2 所示. 由表 2 可知, HH3DEDA 在大部分问题上的测试结果都明显优于 HHGA 和 HHEDA, 这验证了 3DEDA 作为高层策略的有效性。

实际上, 将 3DEDA 和 EDA 这类基于概率模型的算法作为高层策略, 可一定程度上避免 GA 这类传统进化方法中普遍存在的对较优解模式破坏的问题, 由表 2 可知, HHEDA 与 HH3DEDA 在大部分问题规模上的测试结果都明显优于 HHGA, 这验证了基于概率模型的算法作为高层策略的有效性. 同时, 相对于常规 EDA, 3DEDA 可更合理地学习和利用高层优质解信息引导搜索方向, 从而获得了更优的性能。

4.2.2 验证本文算法的有效性

为验证 HH3DEDA 的有效性, 将 HH3DEDA 与 DGA^[4]、IWO^[7] 和 IICA^[9], 进行对比. 比较结果如表 3 所示. 由表 3 可知, HH3DEDA 在绝大部分问题上的测试结果明显优于对比算法, 表明 HH3DEDA 是求解 DP-FVTISP 的有效算法。

①* 链接: <https://pan.baidu.com/s/1WTUxI9zT0WJAVD68yf0oA>. 提取码: doo6, 若链接失效请联系通信作者。

表 2 验证 HH3DEDA 高层策略的有效性

问题规模	HHGA	HHEDA		HH3DEDA	
	AVG	AVG	IMP	AVG	IMP
20×5×2	1324.8	1326.6	-0.14%	1322.7	0.15%
20×10×2	1742.0	1745.5	-0.20%	1738.9	0.18%
20×20×2	2406.4	2406.2	0.01%	2403.6	0.12%
50×5×2	2612.4	2544.4	2.60%	2475.0	5.26%
50×10×2	3227.7	3202.9	0.77%	3181.1	1.44%
50×20×2	4135.3	4056.4	1.91%	4039.5	2.32%
100×5×2	4714.7	4633.1	1.73%	4581.9	2.82%
100×10×2	6405.4	6301.0	1.63%	6257.4	2.31%
100×20×2	7229.9	6982.1	3.43%	6913.8	4.37%
20×5×4	1122.7	1083.4	3.50%	1065.0	5.14%
20×10×4	1486.0	1450.7	2.37%	1450.3	2.40%
20×20×4	2035.7	1999.6	1.77%	1999.5	1.78%
50×5×4	2429.5	2355.2	3.06%	2301.1	5.28%
50×10×4	2472.2	2275.7	7.95%	2265.1	8.38%
50×20×4	3603.4	3496.1	2.98%	3390.5	5.91%
100×5×4	4563.9	4393.1	3.74%	4359.2	4.49%
100×10×4	5192.3	4896.4	5.70%	4847.4	6.64%
100×20×4	6218.5	6056.2	2.61%	6009.9	3.35%
20×5×6	1207.1	1183.7	1.94%	1176.2	2.56%
20×10×6	1468.0	1462.8	0.35%	1461.3	0.46%
20×20×6	2066.1	2062.1	0.19%	2063.6	0.12%
50×5×6	2117.5	2084.2	1.57%	2033.7	3.96%
50×10×6	2711.7	2612.7	3.65%	2574.3	5.07%
50×20×6	3384.2	3200.8	5.42%	3123.5	7.70%
100×5×6	4628.5	4426.3	4.37%	4406.5	4.80%
100×10×6	5065.1	4801.5	5.20%	4845.1	4.34%
100×20×6	5643.9	5298.7	6.12%	5423.3	3.91%
Average	3378.3	3271.8	3.15%	3248.5	3.84%
NB	0	3	3	24	24

实际上,如果算法中用于搜索的邻域操作(如交叉、变异、插入和交换)数量越多,则各邻域的共同局部最优值越小,算法可在到达此共同局部最优值之前一直持续向下搜索.现有的群智能算法(包括本文比较的 DGA、IICA 和 IWO)本质上都是采用数个邻域操作,按某种基本固定的顺序反复执行(执行次数为算法进化代数),以实现了对问题解空间的搜索.由于这类算法搜索邻域数量有限且搜索方式单一,导致其实际搜索深度有限. HH3DEDA 高层利用三维概率模型动态控制低层个体执行包含多种邻域操作组合的启发式搜索,有利于引导算法在多种不同邻域区域进行更为深入且有效的搜索,进而能发现存在于复杂解空间中的优质解.因此, HH3DEDA 能在上述实验中表现出更好的

性能.

表 3 HH3DEDA 与 3 种算法的对比结果

问题规模	IWOA	IICA	DGA	HH3DEDA
	AVG	AVG	AVG	AVG
20×5×2	1345.8	1332.0	1340.6	1322.7
20×10×2	1805.1	1758.4	1784.7	1738.9
20×20×2	2452.3	2423.1	2432.3	2403.6
50×5×2	2555.2	2509.8	2528.1	2475.0
50×10×2	3279.7	3230.0	3220.1	3181.1
50×20×2	4176.2	4107.8	4066.5	4039.5
100×5×2	4785.4	4651.5	4552.2	4581.9
100×10×2	6564.6	6381.9	6220.0	6257.4
100×20×2	7288.7	7123.7	6882.9	6913.8
20×5×4	1120.8	1090.3	1109.3	1065.0
20×10×4	1508.3	1471.8	1489.9	1450.3
20×20×4	2046.5	2020.9	2028.6	1999.5
50×5×4	2468.1	2391.9	2380.6	2301.1
50×10×4	2393.2	2305.9	2279.8	2265.1
50×20×4	3589.5	3513.2	3501.6	3390.5
100×5×4	4705.5	4501.0	4366.4	4359.2
100×10×4	5408.1	5076.9	4909.3	4847.4
100×20×4	6463.1	6184.9	5965.0	6009.9
20×5×6	1218.6	1207.0	1231.6	1176.2
20×10×6	1524.4	1492.2	1505.2	1461.3
20×20×6	2115.7	2084.8	2100.1	2063.6
50×5×6	2210.1	2111.3	2095.7	2033.7
50×10×6	2721.8	2695.8	2659.7	2574.3
50×20×6	3297.1	3263.5	3260.8	3123.5
100×5×6	4866.9	4622.1	4465.5	4406.5
100×10×6	5275.3	5038.7	4904.1	4845.1
100×20×6	5935.6	5661.7	5549.1	5423.3
Average	3449.0	3342.7	3290.0	3248.5
NB	0	0	4	23

5 结论

本文首次建立分布式流水线和车辆运输集成调度问题(DPFVTISP)模型,并提出一种超启发式三维分布估计算法(HH3DEDA)进行求解. HH3DEDA 具有如下优点:(1)为克服常用编解码规则直接应用于 DPFVTISP 所存在的不足,设计更为合理的编解码规则,可提高解的质量.(2)结合 DPFVTISP 的特点,在算法低层设计 10 种启发式操作,可增强 HH3DEDA 执行较深入局部搜索的能力.(3)在 HH3DEDA 高层采用 3DEDA 持续更新种群,进而动态确定低层每个个体所执行的启发式算法,可引导算法较快到达解空间

中多个优质区域执行搜索。后续研究将把 HH3DEDA 扩展用于求解绿色分布式生产和车辆运输集成调度问题。

参考文献

- [1] 戚远航, 蔡延光, 蔡颢, 等. 带容量约束的供应链物流运输调度问题的双层变邻域蝙蝠算法[J]. 电子学报, 2019, 47(7): 1434 – 1442.
Qi Y H, Cai Y G, Cai H, et al. Two-level bat algorithm with variable neighborhood search for capacitated vehicle routing problem in supply chain[J]. Acta Electronica Sinica, 2019, 47(7): 1434 – 1442. (in Chinese)
- [2] Moons S, Ramaekers K, Caris A, et al. Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion[J]. Computers & Industrial Engineering, 2017, 104: 224 – 245.
- [3] Mohammadi S, Al-E-hashem S M J M, Rekik Y. An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company[J]. International Journal of Production Economics, 2020, 219: 347 – 359.
- [4] Feng X, Xu Z Y. Integrated production and transportation scheduling on parallel batch-processing machines [J]. IEEE Access, 2019, 7: 148393 – 148400.
- [5] Gharaei A, Jolai F. A branch and price approach to the two-agent integrated production and distribution scheduling [J]. Computers & Industrial Engineering, 2019, 136: 504 – 515.
- [6] Gharaei A, Jolai F. Two heuristic methods based on decomposition to the integrated multi-agent supply chain scheduling and distribution problem[J]. Optimization Methods and Software, 2020. DOI:10.1080/10556788.202.1714615.
- [7] Abdollahzadeh V, Nakhakamalabadi 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: 1 – 21.
- [8] Karimi N, Davoudpour H. A branch and bound method for solving multi-factory supply chain scheduling with batch delivery [J]. Expert Systems with Applications, 2015, 42(1): 238 – 245.
- [9] 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.
- [10] 李尚函, 胡蓉, 钱斌, 等. 超启发式遗传算法求解模糊柔性作业车间调度 [J]. 控制理论与应用, 2020, 37(2): 316 – 330.
- [11] Li S H, Hu R, Qian B, et al. Hyper-heuristic genetic algorithm for solving fuzzy flexible job shop scheduling problem[J]. Control Theory & Applications, 2020, 37(2): 316 – 330.(in Chinese)
- [12] 田云娜, 李冬妮, 刘兆赫, 等. 一种基于动态决策块的超启发式跨单元调度方法[J]. 自动化学报, 2016, 42(4): 524 – 534.
Tian Y N, Li D N, Liu Z H, et al. A hyper-heuristic approach with dynamic decision blocks for inter-cell scheduling[J]. Acta Automatica Sinica, 2016, 42(4): 524 – 534. (in Chinese)
- [13] 周雅兰, 王甲海, 黄聪. 求解排列问题的分布估计离散粒子群优化算法[J]. 电子学报, 2014, 42(3): 561 – 571.
Zhou Y L, Wang J H, Huang C. Estimation of distribution-discrete particle swarm optimization algorithm for permutation-based problems[J]. Acta Electronica Sinica, 2014, 42(3): 561 – 571.(in Chinese)
- [14] 程玉虎, 王雪松, 郝名林. 一种多样性保持的分布估计算法[J]. 电子学报, 2010, 38(3): 591 – 597.
Cheng Y H, Wang X S, Hao M L. An estimation of distribution algorithm with diversity preservation[J]. Acta Electronica Sinica, 2010, 38(3): 591 – 597.(in Chinese)
- [15] Faraji Amiri M, Behnamian J. Multi-objective green flow-shop scheduling problem under uncertainty: Estimation of distribution algorithm[J]. Journal of Cleaner Production, 2020, 251: 119734.
- [16] 曹政才, 林诚然, 黄冉. 带等待时间约束并行机调度问题的 Copula 分布估计算法 [J]. 电子学报, 2017, 45(12): 2949 – 2956.
Cao Z C, Lin C R, Huang R. An estimation of distribution algorithm based on copula for parallel machine scheduling with constrained waiting time[J]. Acta Electronica Sinica, 2017, 45(12): 2949 – 2956. (in Chinese)
- [17] 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.
- [18] Shao W S, Pi D C, Shao Z S. A Pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with se-

quence-dependent setup time[J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(3): 1344 - 1360.

[19] Montgomery D C. Design and Analysis of Experiments [M]. Hoboken: John Wiley and Sons, 2005.

作者简介



周丰顺 男,1998年出生,湖南永州人.硕士研究生,研究方向为优化调度与智能算法.
E-mail: 1445155907@qq.com



张长胜 男,1970年出生,云南昆明人.副教授,硕士生导师,研究方向为光纤传感器技术与应用.
E-mail: 1368037752@qq.com



胡蓉(通讯作者) 女,1974年出生,云南曲靖人.副教授,硕士生导师,研究方向为优化方法与决策支持系统.
E-mail: ronghu@vip.163.com



向凤红 男,1964年出生,四川省盐亭人.教授,硕士生导师,研究方向为智能优化方法.
E-mail: xiangfh5447@sina.com



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