Journal of System Simulation

Volume 36 | Issue 4 Article 13

4-15-2024

Hyper-heuristic Approach with K-means Clustering for Inter-cell Scheduling

Yanlin Zhao
School of Mathematics and Computer Science, Yan'an University, Yan'an 716000, China,
756749010@qq.com

Yunna Tian
School of Mathematics and Computer Science, Yan'an University, Yan'an 716000, China

Hyper-heuristic Approach with K-means Clustering for Inter-cell Scheduling

Abstract

Abstract: According to the actual production situation of China's manufacturing industry, a hyperheuristic algorithm based on K-means clustering is proposed for inter-cell scheduling problem of flexible job-shop. K-means clustering is applied to group entities with similar attributes into the corresponding work cluster decision blocks, and the ant colony algorithm is used to select heuristic rules for each decision block. The optimal scheduling solutions are generated by using corresponding heuristic rules for scheduling of entities in each decision block. Computational results show that, the computational granularity is properly increased by the form of decision blocks, and the computational efficiency of the optimal algorithm is improved. The clustering algorithm could group the processed entities with similar attributes and the suitable rules for entities with different attributes are easy to be chosen. The proposed approach not only improves computational efficiency but also exhibits good optimization performance, and provides a scientific optimization solution for inter-cell scheduling problems.

Keywords

inter-cell scheduling, hyper-heuristic algorithm, decision block, clustering, ant colony optimization

第 36 卷第 4 期 系统仿真学报© Vol. 36 No. 4
2024 年 4 月 Journal of System Simulation Apr. 2024

基于K-means聚类的超启发式跨单元调度方法

赵彦霖，田云娜
(延安大学 数学与计算机科学学院，陕西 延安 716000)

摘要：结合我国制造业实际生产状况，针对柔性作业车间跨单元调度问题，提出一种基于K-means聚类的超启发式算法。*应用K-means聚类算法将相近属性的实体划入相应“工件簇”决策块中，采用蚁群算法为每个决策块选择启发式规则；对每个决策块内的实体运用相应的启发式规则产生调度解。*仿真结果表明：该算法以决策块的形式适度增大了计算粒度，有效降低了算法时间复杂度，以聚类的方式将具有相近属性的被加工实体进行聚集，有利于为不同属性的实体选择合适的规则。该算法提高了计算效率，具有较好的优化性能，是解决柔性跨单元调度的一种有效算法。

关键词：跨单元调度；超启发式算法；决策块；聚类；蚁群算法
中图分类号：TP18；TP301.6 文献标志码：A 文章编号：1004-731X(2024)04-0941-16
DOI: 10.16182/j.issn1004731x.joss.22-1541
引用格式: 赵彦霖, 田云娜. 基于K-means聚类的超启发式跨单元调度方法[J]. 系统仿真学报, 2024, 36(4): 941-956.
Reference format: Zhao Yanlin, Tian Yunna. Hyper-heuristic Approach with K-means Clustering for Inter-cell Scheduling [J]. Journal of System Simulation, 2024, 36(4): 941-956.

Hyper-heuristic Approach with K-means Clustering for Inter-cell Scheduling

Zhao Yanlin, *Tian Yunna*
(School of Mathematics and Computer Science, Yan'an University, Yan'an 716000, China)

Abstract: According to the actual production situation of China's manufacturing industry, a hyper-heuristic algorithm based on K-means clustering is proposed for inter-cell scheduling problem of flexible job-shop. *K-means clustering is applied to group entities with similar attributes into the corresponding work cluster decision blocks, and the ant colony algorithm is used to select heuristic rules for each decision block. The optimal scheduling solutions are generated by using corresponding heuristic rules for scheduling of entities in each decision block.* Computational results show that, the computational granularity is properly increased by the form of decision blocks, and the computational efficiency of the optimal algorithm is improved. The clustering algorithm could group the processed entities with similar attributes and the suitable rules for entities with different attributes are easy to be chosen. The proposed approach not only improves computational efficiency but also exhibits good optimization performance, and provides a scientific optimization solution for inter-cell scheduling problems.

Keywords: inter-cell scheduling; hyper-heuristic algorithm; decision block; clustering; ant colony optimization

收稿日期：2022-12-23 修回日期：2023-04-11
基金项目：国家自然科学基金(61763046，62041212)
第一作者：赵彦霖(1998-)，男，硕士生，研究方向为最优化方法、理论与应用。E-mail：756749010@qq.com

0 引言

随着社会进步和科技发展，消费者需求逐渐趋向于多样化，生产制造产业面临多品种、小批量、生产周期短等生产需求的挑战。为了适应市场变化需求，单元制造系统(cellular manufacturing systems，CMS)随之诞生。CMS 具有降低库存和材料处理成本、加大生产控制的灵活性、节省生产和设置时间等优势。然而在实际生产中，由于一些特殊工件(exceptional parts，EP)无法在同一单元下完成所有的加工，单个单元生产能力有限，从实际成本和加工效率上考虑，需要单元间相互协作，工件需要频繁依靠运载能力有限的小车进行转移。文献[1]指出单元间转移不可避免，理想情况的CMS 很难实现，跨单元调度问题是实际生产系统中必须要考虑的问题。

在针对跨单元调度问题的研究中，学者们起初倾向于优化能力较强的元启发式算法。文献[2]提出一种采用专门修复策略的遗传算法(GA)，以最小化最大时间为目标，考虑工序排序和单元间运输时间，显著减少了计算时间。文献[3]采用元启发式改进的三层遗传算法(three-layer chromosome GA，TCGA)，以经典的GA 算法和选择具有加工时间最短的部件(shortest processing time，SPT)规则进行了比较，缩短了单元间移动时间。文献[4]提出了一种具有较好收敛速度的交替迭代遗传算法(an alternant iterative method by hybrid GA，AIHGA)，考虑了 3 种生产计划约束，提高了在大规模调度问题中生产成本解的质量。文献[5-6]提出了两阶段启发式算法和嵌套禁忌搜索(nested tabu search，NTS)算法，通过实验测试了这 2 种启发式算法在跨作业单元调度问题对总生产时间、最大完工时间为目标的有效性和高效性。文献[7]提出一种改进的和声搜索算法(improved harmony search，IHS)，考虑单元调度顺序，并对比 3 种不同的改进自适应神经模糊推理系统(adaptive neural fuzzy inference system，ANFIS)结构，在小规模调度中提高了 makespan 解的质量。文献[8]采用了一种分散搜索方法来解决CMS 系统中异常零件的调度问题，并将它看作整体，解决了中小型多单元调度问题，以加权延迟总和最小(total weighted tardiness, TWT)为优化目标优化了计算时间。文献[9]提出一种组合蚁群优化算法(combination ant colony optimization，CACO)，考虑了单处理机和批处理机协同优化，以最大完工时间为优化目标，优化了计算时间。文献[10]针对具有柔性路径的跨单元调度问题，提出了一种基于信息素的方法(pheromone-based approach，PBA)。

跨单元调度问题是一个复杂的组合优化问题，相关的子问题都已经被证明是一个 NP 难问题[11]。在文献[2-10]所被验证的最大问题规模(单元数 5~8 个/机器数 17~40 台/工件数 4~80 个)下，元启发式需耗时百秒(371 s)[9]，且文献[3-6]未考虑单元间转移工件的时间。在复杂装备的单元制造系统中，通常存在上百个不同类型的工件和机器，每件待加工的成品需要上千道工序，置于十几个单元中进行加工，再加上实际加工过程中部分机器加工能力重叠，导致加工路径不唯一，运输工具的数量和承载能力有限，导致问题规模增大，求解复杂度增加[12]，在可接受的时间内元启发式算法难以得到较为满意的解。

为了权衡优化性能和计算效率，文献[13]选择超启发式算法来解决优化调度问题。超启发式算法又被称作“搜索启发式算法的启发式算法”，可以在求解NP−难问题方面具备更佳的求解效率[14]。超启发式算法并不直接作用于具体问题，而是间接地对规则调用，降低人为主观因素对解的影响，从而对解决复杂的优化问题带来便利。文献[15-16]提出基于蚁群算法与遗传编程(genetic programming, GP) 的 AHGT(ACO-based hyper-heuristic approach with genetic programming) 和基于混合蛙跳算法(shuffled frog leaping algorithm，SFLA)与遗传编程的 SHAG(SFLA-based hyperheuristic approach with

DOI: 10.16182/j.issn1004731x.joss.22-1541

GP)，引入时间窗概念。文献[17]提出了一种序列选择的 SSHH(sequence selection hyper-heuristic)超启发式算法应用于柔性作业车间问题。

为了进一步提升解决大规模复杂问题的求解效率，一些学者在超启发式算法的框架上加入了决策块概念。文献[18]采用基于动态决策块和 GA 的超启发式算法(dynamic decision block GA-based hyper-heuristic，DRGA)，以解决多目标作业车间调度问题。在超启发式算法对跨作业单元调度的研究中，文献[19-20]分别提出了离散蜂群与决策块结合的超启发式算法和以分组遗传算法(grouping GA，GGA)作为启发式搜索框架的超启发式算法；文献[21]采用基于蚁群算法与动态决策块结合的超启发式算法(dynamic decision block and ACO based hyper-heuristic，DABH)对工序分派、排序以及小车运输 3 个子问题的规则进行选取，并通过实验证明了动态决策块在性能优化方面优于静态决策块。在以上加入决策块策略的文献中可以看出，决策块的大小会对计算效率产生显著影响。决策块策略可适当缩小搜索解的空间，有利于提高计算效率。然而这种形成方式未考虑决策块内部个体之间的属性联系，如工件的最晚交工时间、机器的加工能力以及小车的运输能力等，这些属性是调度规则中的主要计算依据，在属性与规则之间往往具有较强的应用映射关系。

本文从决策块成员相关性的角度出发，采用聚类算法构建决策块，以提升算法寻优性能，在文献[21]基础上设计了一种基于蚁群优化和在 K-means 聚类动态决策块的超启发式算法 K-DABH。在解决调度问题的过程中，考虑工件、加工机器和运输小车这些实体所具有的属性。在传统超启发式的框架基础上，根据实体属性的差异进行聚类，依据聚类结果构建出决策块，然后对决策块搜索启发式规则。本文的主要贡献有 3 点：①将聚类思想加入超启发式算法，将具有相近属性的实体归为同一决策块，促进了相似属性的实体对启发式规则选取的统一性；②通过轮廓系数选择 K-means 算法参数，解决了 K-means 参数选取困难的问题，确定了所需形成决策块的数量，有效提升算法的适应能力；③用蚁群算法为每个相近属性决策块选择相同的启发式规则，保证算法的优化性能。

1 问题描述

柔性跨作业单元问题可以描述为由多个生产单元组成，每个单元均有一台运输能力有限的工具，n 个工件共有 m 台机器上可供加工选择，每个工件有 K 道工序，工序存在跨单元柔性(每道工序可以选择的加工机器可以有多台)。本文主要根据跨单元调度问题抽象出问题模型，优化目标为最小化最大完工时间(makespan)。

1.1 问题假设

本文考虑具有柔性的跨单元调度问题假设：

(1) 准备时间独立于工序次序，所有工件从 0 时刻到达；

(2) 每个工件的最迟交货时间和工序加工顺序固定；

(3) 工件的加工路径由多道工序组成，每道工序只能在具有该工序加工能力的机器上进行加工；

(4) 工序的加工时间、加工流程已知，但机器加工能力重叠且机器加工效率不一定相同；

(5) 在同一时刻每台机器只可对一道工序进行加工；

(6) 工序加工是非中断式生产，加工期间不能停下来更换其他工件或是中断机器；

(7) 工件的任一工序在一个单元内只能有一台可加工的机器；

(8) 工件在单元内部的转移时间忽略不计，跨单元时间仍要考虑；

(9) 小车只装载所属单元的工件，运输的同一批次工件可以有不同的目的单元，但运输过程中

不可再次装载工件，装载时间忽略不计；

(10) 小车到达目的单元自动卸载工件，一旦工件全部卸载完毕立即返回所属单元，卸载时间忽略不计；

(11) 不考虑机器的磨损、调整、准备等。

1.2 符号列表

与本文有关的模型参数及符号：

(1) 系统变量

N: 工件总数

i: 工件索引, $i=1,2,\cdots,N$

M: 机器总数

m: 机器索引, $m=1,2,\cdots,M$

j: 工件 i 的工序索引, $j=1,2,\cdots,J(i)$

O_{ij}: 工件 i 的第 j 道工序

$J(i)$: 工件 i 的工序总数

M_{ij}: 工序 O_{ij} 可选择的机器集合

P_{ijm}: 机器 m 对工序 O_{ij} 的加工时间

W_i: 工件 i 的权重

d_i: 工件 i 的交货日期

S_i: 工件 i 的体积

v: 运输工具的承载力

S_{ij}: 工序 O_{ij} 的开始时间

P_{ij}: 实际加工 O_{ij} 工序的时间

f_{ij}: 工序 O_{ij} 的完工时间

t: 时间索引, $t=1,2,\cdots,T$

c: 单元内部载具索引, $c=1,2,\cdots,C$

c_{ij}: 完成工序 O_{ij} 工件将前往的目的单元

t_i: 工件 i 的完工时间

t_{ij}: 完成工序 O_{ij}工件前往目的单元的转移时间

T_{ij}: 在单元 c 与 c' 转移所花费的时间

b: 载具 c 上的批次索引, $b=1,2,\cdots,B(c)$

$$\alpha_{ijm}=\begin{cases}1, & \text{工序 } O_{ij} \text{ 允许在机器}m\text{上加工}\\ 0, & \text{其他}\end{cases}$$

$$\beta_{mc}=\begin{cases}1, & \text{机器 } m \text{ 属于单元}C\text{内}\\ 0, & \text{其他}\end{cases}$$

$$\gamma_{mc}=\begin{cases}1, & \text{工序 } O_{ij} \text{ 的加工需要跨其他单元加工}\\ 0, & \text{其他}\end{cases}$$

(2) 决策变量

$$X_{ijm}=\begin{cases}1, & \text{工序}O_{ij}\text{被分配在机器}m\text{上加工}\\ 0, & \text{其他}\end{cases}$$

$$Y_{ijt}=\begin{cases}1, & \text{工序}O_{ij}\text{在 } t \text{ 时刻进行加工}\\ 0, & \text{其他}\end{cases}$$

$$Y_{ijpqm}=\begin{cases}1, & \text{工序}O_{ij}\text{比工序}O_{pq}\text{优先在 } m \text{ 机器加工}\\ 0, & \text{其他}\end{cases}$$

$$Z_{ijcbq}=\begin{cases}1, & \text{加工完}O_{ij}\text{后工件}i\text{需进入小车 } c \text{ 的}\\ & \text{第}b\text{批次, 处于第 } q \text{ 个运输顺序}\\ 0, & \text{其他}\end{cases}$$

$$W_{cbt}=\begin{cases}1, & t \text{ 时刻小车 } c \text{ 开始运输第 } b \text{ 批次}\\ 0, & \text{其他}\end{cases}$$

1.3 目标函数及约束条件

本文问题考虑的目标主要是最小化最大完工时间，由工序加工时间与相邻工序转移工件时间组成。目标函数：

$$\min\max\left\{t_i=\sum_{j}^{J(i)}\sum_{m}^{M}\sum_{c}^{C}((p_{ij}+t_{ij})\alpha_{ijm}\beta_{mc})\right\},\quad i\in[1,n],\quad \forall j,m \tag{1}$$

问题的约束条件。

工件的任意一道工序只能被分派给一台具有加工该工序能力的机器进行加工一次：

$$\sum_{m=1}^{M}X_{ijm}=1,\quad \forall i,j \tag{2}$$

$$\sum_{m=1}^{M}X_{ijm}\,p_{ijm}=1,\quad \forall i,j \tag{3}$$

任何一道工序都必须有且只有一次加工：

$$\sum_{t=1}^{T}Y_{ijt}=1,\quad \forall i,j \tag{4}$$

工序只可分配至能加工该工序的机器上：

$$(1-\alpha_{ijm})X_{ijm}=0,\quad \forall i,j,m \tag{5}$$

工序的开始时间：

$$s_{ij}=\sum_{t=1}^{T}Y_{ijt}t,\quad \forall i,j \tag{6}$$

工序实际加工时间：

$$p_{ij}=\sum_{m=1}^{M}X_{xjm}p_{ijm},\quad \forall i,j \tag{7}$$

DOI: 10.16182/j.issn1004731x.joss.22-1541

完工时间：

$$f_{ij}=s_{ij}+p_{ij},\quad \forall i,j \tag{8}$$

确保工件进入机器加工队列的顺序性：

$$(X_{i'j'm}s_{i'j'}\geqslant X_{ijm}p_{ijm}+X_{ijm}s_{ij})\vee (X_{ijm}s_{ij}\geqslant X_{i'j'm}p_{i'j'm}+X_{i'j'm}s_{i'j'}),\ \forall i,i',j,j',m,X_{ijm}X_{i'j'm}=1,i\neq i',j\neq j' \tag{9}$$

每次小车只运输一个批次且不重复运输：

$$\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{t=1}^{T}W_{cbt}=1,\quad \forall c,b \tag{10}$$

工件需要运输时只能分派入本单元内一个小车的一个批次：

$$\sum_{c=1}^{C}\sum_{b=1}^{B(c)}\sum_{q=1}^{Q(b)}Z_{ijcbq}=\gamma_{ij},\quad \forall i,j \tag{11}$$

$$\sum_{m=1}^{M}X_{ijm}\beta_{mc}=\sum_{b=1}^{B}\sum_{q=1}^{Q}Z_{ijcbq},\forall i,j,c,\gamma_{ij}=1 \tag{12}$$

每个批次内小车的承载力都要大于等于所运输工件总的体积：

$$\sum_{i=1}^{N}\sum_{q=1}^{Q(b)}Z_{ijcbq}S_i\leqslant V,\quad \forall c,b \tag{13}$$

一次运输小车的目的单元：

$$c_{ij}=\gamma_{ij}\sum_{m=1}^{M}X_{ijm}\beta_{mc}c,\quad \forall i,j \tag{14}$$

工件依次等待小车到达目的单元才被卸载：

$$t_{ij}\geqslant Z_{ijcbq}T_c c_{ij},\quad \forall i,j,c,b \tag{15}$$

$$t_{ij}\geqslant Z_{ijcbq}Z_{i'j'cb(q+1)}(t_{ij}+T_{c_{ij}c_{i'j'}}),\forall i\,j,i',j',c,b,q \tag{16}$$

小车若要开始运输必须等待批次内所有工件的前一道工序加工完成：

$$Z_{ijcbq}f_{ij}\leqslant\sum_{t=1}^{T}W_{cbt}t,\quad \forall i,j,c,b,q \tag{17}$$

只有前一道工序加工完才可进行下一道工序的加工(若下道工序需要跨单元则工件需要被运输进目的单元)：

$$S_{i(j+1)}\geqslant\max\left\{f_{ij},\gamma_{ij}\left(\sum_{c=1}^{C}\sum_{b=1}^{B(c)}\sum_{q=1}^{Q(b)}Z_{ijcbq}\times\sum_{t=1}^{T}W_{cbt}(t+t_{ij})\right)\right\},\quad \forall i,j \tag{18}$$

每次小车将一个批次承载的工件运输完毕都要返回起始单元，等待下一批次运输：

$$\sum_{t=1}^{T}W_{cbt}t+t_{ij}+Z_{ijcbq}T_{c_{ij}c}\leqslant\sum_{t=1}^{T}W_{c(b+1)t}t,\quad \forall i,j,c,b,q \tag{19}$$

2 算法设计

蚁群算法由 Dorigo 提出，蚁群优化理论 ACO 作为群体智能的典型[22]逐渐吸引大批学者注意。Dorigo 通过模拟蚂蚁寻觅食物的行为和信息素通信方式，解决旅行商问题(TSP)[23]。文献[21]在传统超启发式蚁群算法应用的基础上引入动态决策块策略，提出了 DABH 算法，对解决跨单元生产调度和运输调度问题上有更好的效果。基于上述成功应用，本文提出一种基于 K-means 与 DABH 的 K-DABH 算法，解决柔性车间下跨单元调度生产问题。针对复杂性和灵活性大幅增加的跨作业单元调度问题，不仅需要兼顾多个子问题之间的协同，同时还需考虑到算法的优化性能和计算效率。如图 1 所示，K-DABH 利用 K-means 简单高效、所需设置初始参数少的特点来合理划分决策块策略，通过缩小搜索范围，进而达到算法效率的提升。除此之外，K-DABH 以超启发式算法为框架，将 ACO 作为高层优化，利用信息素来不断寻找更优的加工路径，得到更好的局部最优解，并为 K-means 形成的工件簇(工件决策块)寻找合适的启发式规则，利用启发式规则找到跨作业单元调度中的完整调度解。本文模型主要解决的问题有 2 个：①工件、机器、小车决策块如何形成；②规则分配。

2.1 基于 K-means 聚类确定工件决策块算法框架

在决策块形成问题中，决策块的数量会根据问题规模以及轮廓系数而发生改变，决策块的大小和决策块的组成会根据聚类算法以及实体属性而动态变化。工件决策块代表工件的某道工序将

分配至哪一台加工机器；机器决策块代表在机器的缓冲区内工件的先后排列加工顺序；小车决策块包含了小车每次运输对路径的选择以及小车每次所要运输哪些工件。K-DABH 算法解决以上 3 个问题的流程描述如图 2 所示。

在图 2 中，选择工件的最小完工时间属性作为工件特征；K 的值根据轮廓系数来确定；工件的簇数即为工件决策块的数量；簇内的工件数为工件决策块的大小；因为工序排序和小车运输部分受前面工序分派的影响，机器、小车的决策块内工件部分属性差异不大，再次进行聚类会导致算法效率不高且解空间的多样性变差，因此，机器和小车的决策块大小仍然依靠蚁群算法的信息素来优化，最终合理划分。另外，通过从候选规则中对每个决策块选择信息素浓度高规则。在分派生产过程中，同一决策块下的实物采用同一种规则，生成最终调度解。

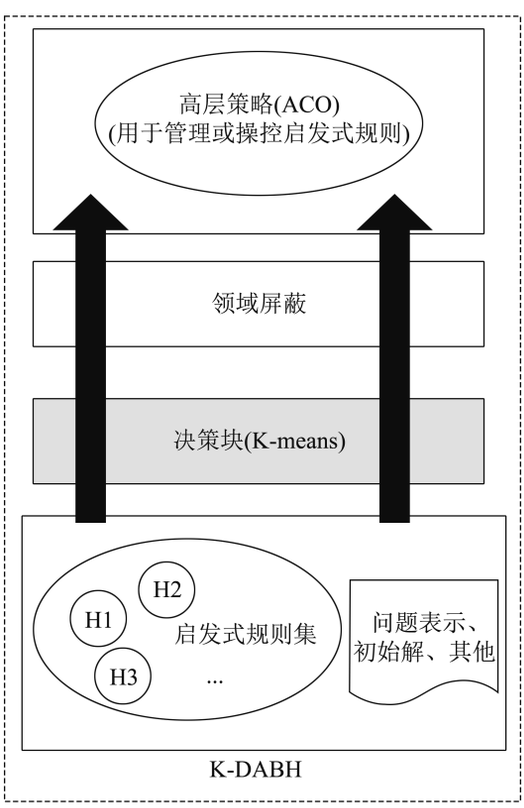

图 1　K-DABH 概念模型
Fig. 1　K-DABH conceptual model

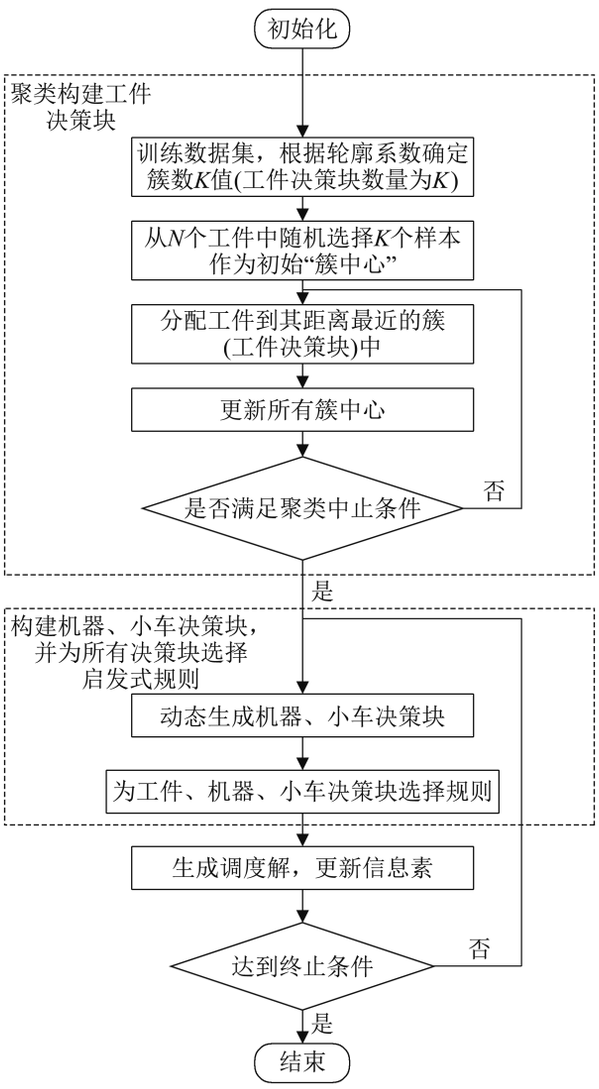

图 2　K-DABH 算法的整体流程图
Fig. 2　General algorithm of K-DABH

2.2 基于 K-means 聚类决策块的形成

2.2.1 K-means 算法原理

K-means 算法属于一种无监督学习，是聚类分析中一种基于划分的算法[24]。K-means 算法的核心思想是在数据集中随机选取 K 个初始聚类中心 C_i，$1 \leqslant i \leqslant K$，然后利用欧式距离等简单距离度量来反映两个元素之间的不同[25]。给定一个样本空间 $X=\{x_1, x_2, \cdots, x_n\}$，在空间中对于聚类中心与样本之间差值的欧式距离 D 计算公式为

$$D(X, C_i)=\sqrt{\sum_{j=1}^{m}(\boldsymbol{x}_j-\boldsymbol{C}_{ij})^2} \tag{20}$$

式中：$\boldsymbol{x}_j$为x的第j个属性向量；$\boldsymbol{C}_{ij}$为C_i的第j个属性向量；m为数据对象的维数。样本X_i分别与聚类中心C_i进行比较，若满足式(21)，则把样本归到差值最小的C_i所在的簇W_K中($X_i \in W_K$)。

$$d(X_i, C_i) = \min\{D(X_i, C_i), \quad i = 1, 2, \cdots, n\} \tag{21}$$

使用 K-means 将相似属性聚类成簇，簇之间属性差异较大。case34 样本空间X取 40 时，K取 14 时可形成的簇数目为 14，结果如图 3 所示。

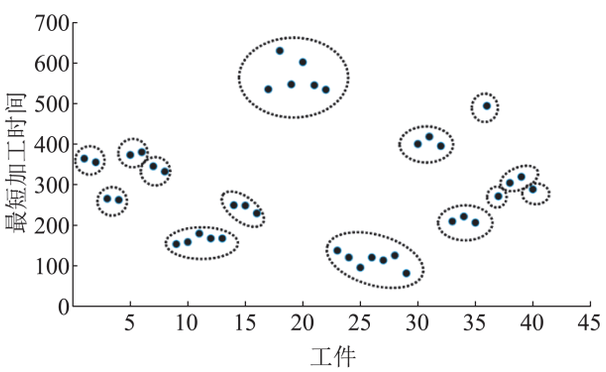

图 3　K-means 聚类示例
Fig. 3　K-means clustering example

2.2.2 轮廓系数

轮廓系数基于紧密性和分离性的比较，可以用来选择一个适当数量的聚类，并提供了一个聚类有效性的评估[26]。对于样本空间中一个点x_i而言，轮廓系数的计算为

$$S(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}} \tag{22}$$

式中：$a(x_i)$为点x_i和它所属簇内其余点的距离平均值，簇内不相似度体现凝聚度；$b(x_i)$为点x_i和非它所属簇的所有点间平均距离中最小值，簇间不相似度体现分离度。其轮廓系数$S(x_i)$取值范围为[−1,1]，其值越大，样本聚类效果越合理；其值越小，样本更应该聚类到另外的簇中；$S(x_i)$若近似为 0 则说明样本接近于 2 个簇的边界中。

文献[27]提出了根据数据集的大小n，找到最佳K取值范围应当是(1，$\sqrt{n}$)之间。本文通过轮廓系数评估动态确定K的取值范围。在评估过程中，每个轮廓系数对应相应的K值，因此，可以根据较好的轮廓系数确定K的合理取值范围。这种通过评估动态确定K值的方法可以去除不合理的K值，比如，K过大或过小，K大于实体数而产生空集合等情况。case34 参数实验中，通过对测试数据集进行聚类，利用轮廓系数评估确定最佳$K_{S(xi)}$的取值范围是[4 , 60)，从而约束K的取值，如图 4 所示。

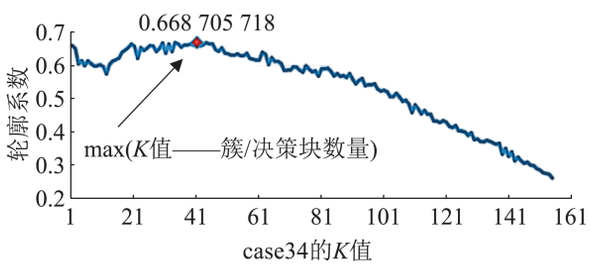

图 4　case34 的轮廓系数
Fig. 4　Profile coefficient of case34

2.3 规则选取

在 K-DABH 算法中，通过对决策块中的实体选择启发式规则进而得到调度解，规则的选取需要依靠信息素来搜索。算法寻优的目标需要根据调度解的优劣情况，在迭代算法过程中更新信息素。

2.3.1 信息素结构

K-DABH 构造信息素结构时，主要包含构建决策块矩阵以及启发式规则的选取矩阵。针对分派问题，工件的决策块部分由 K-means 聚类先一步组合生成，因此，工件决策块数以及决策块内部的实体数都已先一步确定，后续还需对其进行规则选取。但对于排序和运输子问题依旧要对机器和小车划分矩阵。

对于工件、机器、小车的启发式规则矩阵分别用$N \times R'$、$M \times R''$、$C \times R'''$表示，N、M、C分别表示工件、机器小车的数量；R'、R''、R'''分别表示分派、排序、运输 3 类规则的数量。在划分矩阵中用$\tau_{x,y}$表示大小为y的第x个决策块的信息素含量；用$\tau_{x,p}$表示使用了第p个规则的第x个决策块的信息素含量。

2.3.2 规则集

高层次启发式算法通过挑选和应用一些低层次启发式规则[28]，得到问题的调度解。对于本文低层次启发式规则[29]对应的主要为工序的分派和排序、工件运输。

(1) 分派

在工件开始加工前或加工完成一道工序后，扫描当前单元是否有可选择的机器，是否跨单元加工，对该工序进行规则选取。分派启发式规则如下：

1) 选择加工该工序用时最短的机器(shortest processing time，SPT)；

2) 选择当前时刻可用容量最大的缓冲区的机器(most available，MA)；

3) 选择最早可用的机器(first available，FA)；

4) 选择加工该工序后具有最早完成时间的机器(earliest finish time，EFT)；

5) 选择加工该工序后具有最低占用率的机器(least utilization，LU)。

(2) 排序

如果缓冲区存在工件，机器会对当前缓冲区内待加工的一组工件进行先后排序，然后按顺序选取工件加工。排序启发式规则如下：

1) 选择具有最小松弛时间的工件(minimum slack，MS)；

2) 选择当前机器缓冲队列中等待时间最长的工件(time in shop，TIS)；

3) 选择具有最早交货期的工件(earliest due date，EDD)；

4) 选择加工时间最短的工件(SPT)；

5) 选择具有最短剩余加工时间的工件(shortest remaining processing time，SRPT)；

6) 选择具有最小加工时间比率的工件(shortest processing time ratio，SPTR)；

7) 选择具有最小加权加工时间的工件(weighted shortest processing time，WSPT)；

8) 选择具有最小加权交货期的工件(weighted earliest due date，WEDD)；

9) 选择 COVERT(cost over time)值最大的工件，计算式为

$$\frac{w_i}{p_{ijm}}\max(1-\max(d_i-p_{ijm}-t),0,0) \tag{23}$$

10) 选择 ATC 值(apparent tardiness cost)最大的工件，计算式为

$$\frac{w_i}{p_{ijm}}\exp\left(-\frac{\max(d_i-p_{ijm}-t),0}{K\bar{p}}\right) \tag{24}$$

11) 选择 S/RPT(slack per remaining processing time)值最小的工件，计算式为

$$\max\left(\frac{d_i-p_{ijm}-t}{p_{ijm}},0\right) \tag{25}$$

(3) 运输

如果单元中存在已经加工完前一道工序并等待跨单元的工件，且小车不超载的情况下，采用运输规则对待转移工件进行批次运输与路径决策。完成批次运输后再回到原单元重复操作。运输规则如下：

1) 选择当前机器缓冲队列中等待时间最长的工件(TIS)；

2) 选择具有最小加权加工时间的工件(WSPT)；

3) 选择具有最小加权交货期的工件(WEDD)；

4) 选择加工时间最短的工件(SPT)；

5) 选择具有最短剩余加工时间的工件(SRPT)；

6) 选择具有最小加工时间比率的工件(SPTR)；

7) 选择具有最早交货期的工件(EDD)。

2.3.3 确定决策块大小和规则

对于决策块大小的确定，K-DABH 算法在原始 DABH 算法上增加了 K-means 聚类算法，将聚类形成的决策块再使用蚁群算法寻找规则。K-means 算法根据轮廓系数确定 K 值(即决策块数)的同时，决策块内也包含了一定数量的实体，并且决策块之间实体属性的差异较大，决策块内实体属性差异较小。蚁群算法寻找规则时通过信息

DOI: 10.16182/j.issn1004731x.joss.22-1541

素浓度大小决定选择规则的概率。

根据 K-means 确定决策块大小时，先选取实体属性，根据轮廓系数确定聚类数 K，然后用欧式距离计算公式将全部工件实体进行簇类划分：

$$N=\sum_{i=1}^{K} d_{in} \tag{26}$$

式中：d_{in} 为对于决策块 i 中的某类实体的个数；K 为决策块个数；N 为某类实体个数的总数。

根据蚁群算法选择启发式规则时，对于决策块 i' 选择了启发式 q 的概率为

$$P(d_{i'q})=\frac{\tau_{i'q}}{\sum_{k=1}^{R}\tau_{i'k}} \tag{27}$$

式中：$\tau_{i'q}$ 为决策块 i' 选择了启发式 q 的信息素浓度；R 为分派、排序、运输问题启发式规则的个数。

若决策块 i'' 选择第1个规则，对于启发式规则部分信息素的更新，所有信息素按一定比例减少。矩阵元素 $\tau_{i''l}$ 的信息素更新调整为

$$\tau_{i''l}=(1-\rho)\tau_{i''l}+\rho\Delta\tau \tag{28}$$

式中：$\Delta\tau=Q/S_c$，S_c 为更新信息素的调度解对应的目标函数值，Q 为信息素更新量影响因子；ρ 为信息素挥发系数。

2.4 基于决策块的编码

超启发式算法中的决策块指的是底层次启发式规则所作用的决策范围大小。在 K-DABH 算法中，将实物划分成若干的决策块，决策块的大小即为实物的数量。本节依据工件属性进行聚类分析，得到属于工件的决策块，依据启发式规则，作用于决策块。

将所有实物划分到相应的决策块内，实物的集合为 $E=\{e_1,e_2,\cdots,e_N\}$，其中，N 为工件、机器与小车的数量。所有实物划分的决策块集合为 $B=\{b_1,b_2,\cdots,b_L\}$。所有的规则集合为 $H=\{H'_\alpha,H''_\beta,H'''_\lambda\}$，其中，$H'_\alpha=\{h'_1,h'_2,\cdots,h'_\alpha\}$、$H''_\beta=\{h''_1,h''_2,\cdots,h''_\beta\}$、$H'''_\lambda=\{h'''_1,h'''_2,\cdots,h'''_\lambda\}$，分别对应分派、排序、运输规则集合。此外还需满足 2 个条件：①$\bigcup_{i=1}^{L} b_i=E$；②$\bigcap_{i=1}^{L} b_i=\varnothing$。

条件①意味着所有的实物都将划分入对应的决策块中，条件②意味着不论哪一种实物都只能进入一个决策块中不可重复出现。举例编码示例如图 5 所示。

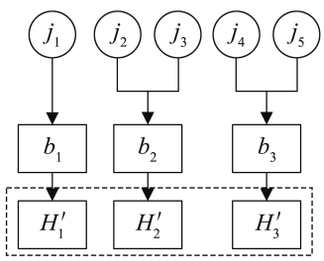

图 5 决策块编码示例

Fig. 5 Example of decision block encoding

图 5 表示 5 个工件实物 j，形成 3 个决策块 b，决策块对应规则 H'_1、H'_2、H'_3。决策块大小对结果影响很大。当决策块过大会漏掉很多解，优化性能受到限制，从而出现解的单一。反之，若决策块太小则会大大降低计算效率，不利于解决本文问题。因此，如何有效缩小搜索空间，保证解的多样性还能提高算法效率，是本文研究的重点。

2.5 基于决策块的解码

K-DABH 主要有决策块的形成和规则分配两个部分。其中，工件决策块需要选择工件属性，确定 K-means 算法的参数，并进行 K-means 聚类，最终形成总体数量不变的工件决策块。机器和小车决策块通过信息素搜寻。

K-DABH 通过蚁群算法为决策块挑选启发式规则，解码过程即系统将得到的一组规则编码序列转换成具体调度解。

步骤 1：初始化参数，置离散事件模拟器时钟 t=0。

步骤 2：对各个单元的工件、机器以及小车分别随机分派相应的规则。

步骤 3：当加工作业全部完成，工件都已完工，转步骤 11。

步骤4：依据分派规则，对存在柔性路径的工件选择加工机器；若不需分派工件则说明下一道加工工序一定，直接将此工件置于该机器的缓冲队列等待加工。

步骤5：判断单元内小车是否处于空闲状态且有带运输的工件，若是则转到步骤6；否则转到步骤7。

步骤6：小车依据运输规则判断待转移工件的先后顺序，确定小车的运输途径同时确保小车每次批次转移时不能超过其运载量。

步骤7：倘若机器处于闲置状态，转到步骤8；否则，转到步骤10。

步骤8：依据排序规则调度工件进行加工。

步骤9：将工件加工过程的开工、完工时间记录下来。

步骤10：$t = t+1$，转到步骤3。

步骤11：通过观察步骤9所记录的信息，用式(1)计算相应的目标函数值。

3 实验与分析

为了验证K-DABH算法的计算效率与优化性能，本文对比实验了多组数据，硬件环境为64位Win10操作系统，仿真实验采用JAVA语言实现，运行在2.80GHz Inter(R)Core(TM)i7-7700HQ CPU，内存为8 GB RAM的PC机上。

3.1 实验设计

关于柔性车间跨单元调度问题目前没有相关的Benchmark测试集，因此，为了验证算法的有效性，本文根据不同的机器与工件设计了多组测试用例。将最小化最大完工时间作为优化目标，并对比5种工件属性下的目标值。

实验模拟实体的规模各不相同，随机产生的实体规模与属性取值如表1所示。

参数实验总共设计了20个不同规模的测试用例。每个规模的测试用例下随机产生10个算例(Instance)，为了验证算法有效性，以这10个算例的目标函数值Makespan的平均值作为此测试问题的性能指标。对于不同的算例分别进行5次仿真实验。以Jn1Mm1Cc1表示测试问题，例如J5M6C3意味着该测试问题下需加工工件数为5、单元数为3、机器数为6。

表1 生成算例属性值
Table 1 Parameters of test problems

生成算例参数	取值分布
工件数	[5,450]
工件权重	(0,1]
机器数	[6,120]
每个单元机器数	[2,6]
单元数	(3,15]
小车承载量	[2,10]
工序加工时间	[1,80]
跨单元所需时间	[6,50]
工序数	[4,25]

3.2 实验

3.2.1 属性选取

从工件属性集合中挑选5个属性进行对比分析，5个属性以Makespan为目标在相同的6个随机挑选的测试用例，相同其余参数下，进行10次实验，实验结果去除最大、最小值，将剩余结果求取平均值作最终解。记录不同属性在K-DABH首先需要选取实体属性并聚类。K-means聚类下的轮廓系数与$K_{S(xi)}$如表2所示。

从轮廓系数来看，工序数属性聚类效果最好。但根据表3所示，以最短加工时间属性聚类对于优化Makespan目标函数更加稳定与优异，因此，本文主要选择最短加工时间用作聚类属性元素。

3.2.2 参数分析

无论是蚁群、DABH还是K-means、K-DABH算法的参数设置都会对算法造成很大影响。为了确定最优的参数组合，需要将不同参数组合进行对比分析。本文采用方差分析法进行参数实验分析，通过分析参数对目标Makespan的影响，找到最优参数。

DOI: 10.16182/j.issn1004731x.joss.22-1541

表2　5个实体属性轮廓系数和聚类簇数的比较
Table 2　Comparison of contour coefficient and cluster number of 5 entity attributes

测试问题	最短加工时间	交货期	最晚完工时间	工序数	平均最小工序时间
J50M15C5	0.59/14	0.69/17	0.64/12	0.96/18	0.93/14
J60M16C5	0.64/16	0.65/14	0.65/18	0.96/19	0.89/19
J70M20C7	0.67/19	0.69/17	0.64/22	0.96/20	0.87/13
J80M21C7	0.64/10	0.65/08	0.63/06	0.98/20	0.92/17
J90M21C7	0.65/13	0.66/21	0.66/26	1/19	0.97/18
J100M25C9	0.60/08	0.70/38	0.61/06	1/20	0.99/18

注："/"为轮廓系数/$K_{S(xi)}$。

表3　属性之间的性能比较
Table 3　Performance comparisons between properties

测试问题	Makespan				
	最短加工时间	交货期	最晚完工时间	工序数	平均最小工序时间
J50M15C5	4 612.0	4 650.6	4 604.4	4 650.2	4 612.0
J60M16C5	5 254.2	5 286.6	5 339.0	5 245.8	5 302.0
J70M20C7	4 918.6	5 082.0	4 915.4	5 206.8	5 010.4
J80M21C7	3 945.0	4 725.0	4 470.6	4 316.8	4 802.2
J90M21C7	4 507.0	4 844.2	4 657.8	4 814.0	4 758.4
J100M25C9	6 752.0	6 819.6	6 745.4	6 785.4	6 675.8
平均值	4 998.1	5 234.7	5 122.1	5 169.8	5 193.5

本文考虑的K-DABH参数共有3个：聚类决策块数K值、蚁群搜索规则中信息素更新因子Q与信息素浓度最大值$\tau_{\max}$的比值$Q_{\max}$、信息素挥发量ρ。其中，$\tau_{\max}$取值为5，该测试用例下最优轮廓系数对应的$K_{S(xi)}$值为13，参数K的取值分3个层次、$Q_{\max}$和ρ取值分4个层次，如表4所示。

表4　K-DABH参数
Table 4　Parameters in K-DABH

参数	范围
K	(13, 20, 40)
$Q_{\tau\max}$	(0.01, 0.05, 0.2, 0.8)
ρ	(0.01, 0.05, 0.2, 0.8)

将Makespan作为优化目标时，其单因子效应如图6所示。分析多因子交互作用，观察其余因子对算法的影响，如图7所示。根据轮廓系数动态选取决策块大小，聚类工件实体。决策块个数越接近$K_{S(xi)}$优化性能越好，反之解越弱。ρ取0.05，$Q_{\max}$取0.2时，K-DABH更有利于解的性能的提高。

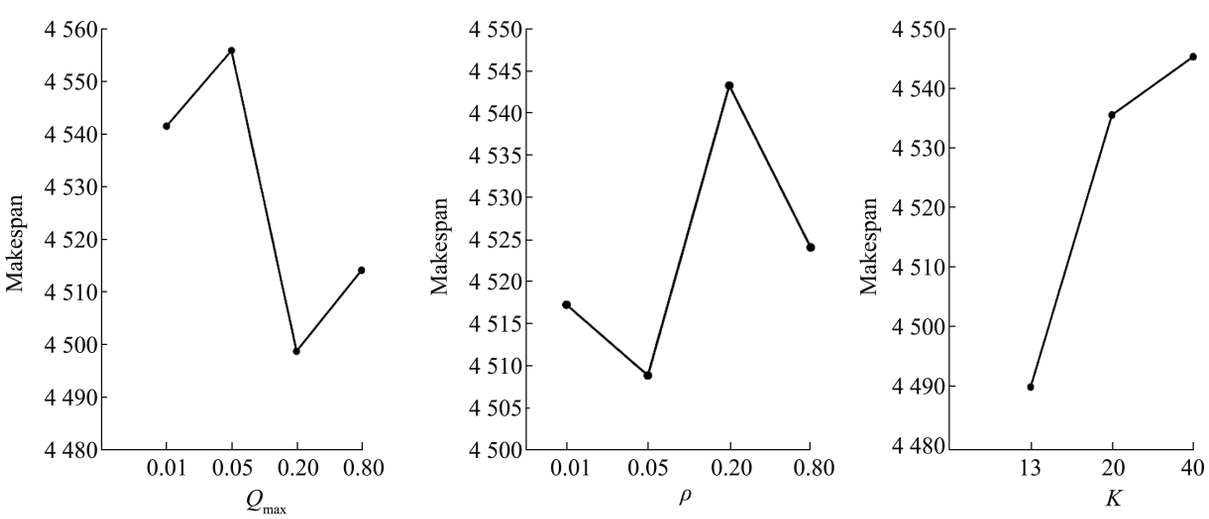

图6　最小化Makespan目标下的单因子主效应图
Fig. 6　Influence of each factor with respect to minimizing Makespan

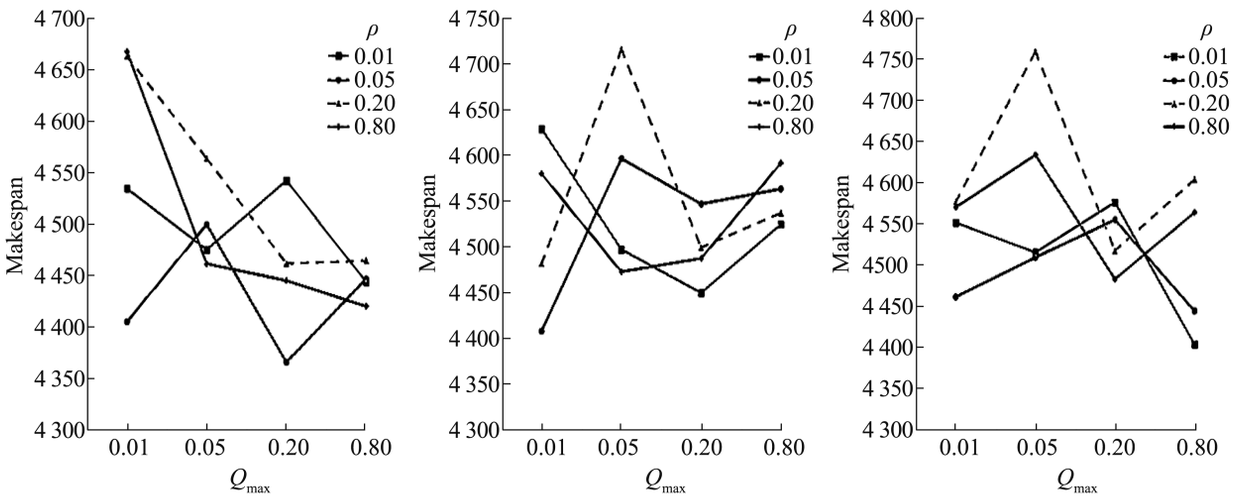

图7 最小化Makespan目标下的三因子交互作用图

Fig. 7 Influence of 3-factor interaction with respect to minimizing Makespan

3.3 与基于静态决策块的方法比较

为验证动态决策块的有效性，在K-DABH算法框架下，将静态决策块划分策略(决策块大小为定值)与动态决策块策略(决策块大小不定)进行对比实验。本节对比2种不同静态决策块划分策略。为确保实验公平，2组对比实验除K值设定不同，其余参数设置与问题规模均相同。

(1) 将某类实体视作一个决策块

每一类实体视作一个决策块，如工件类实体，将参数K设置为固定值1，这种方法记作K-DABH-ONE。

通过计算Gap值对比算法优化效果，K-DABH与K-DABH-ONE之间的Gap值为

$$G_{\text{one}}=(V_{\text{one}}-V_{\text{k}})/V_{\text{k}} \tag{29}$$

式中：V_{one}为K-DABH-ONE的目标函数值；V_{k}为K-DABH的目标函数值。

实验结果如表5所示，面对不同规模的测试用例，K-DABH始终有着较好的优化效果，G_{one}平均值达到6.9%。该实验表明K-DABH-ONE对于该问题求解过程中将同一类实体作为一个决策块，使用相同规则，也就意味着决策块数远小于实体数，极大地缩小了搜索解的范围，失去了解的多样性。而K-DABH算法考虑了实体之间的差异性，利用轮廓系数将决策块数目选择在合理的范围内，并使得决策块内的实体具有较近的属性，让这些相似属性的实体使用同一种规则。

(2) 将每一个实体都视作一个决策块

每个决策块中只含单个实体，如工件类实体，将参数K设置为固定值(工件数N)，这种方法记作K-DABH-ALL。K-DABH与K-DABH-ALL之间的Gap值记为G_{all}。对于不同问题规模的实体划分为包含一个实体的若干个决策块。表5表明，K-DABH与K-DABH-ALL相比，在20个测试问题中的平均性能G_{A}提升了5.8%。由此可见，DABH-ALL中决策块的数量过多，每个决策块只包含较少的实体。虽然能关注每个实体的状态，扩大了搜索空间，但在相同条件下过大的搜索空间大大降低了计算能力，且存在属性差异较大的实体使用同一规则的不利影响，因此，综合求解能力与K-DABH相比较差。

综合表5发现，决策块的大小会影响算法的计算效率和优化性能。决策块大小随着问题规模保持一种不变的定值也会影响优化性能。动态K-DABH引入了轮廓系数，动态确定不同规模下的工件决策块的个数，使决策块大小会随问题规模而变化，有效提升算法的计算效率和适应能力，从而保证算法的优化能力。

DOI: 10.16182/j.issn1004731x.joss.22-1541

表 5　最小 Makespan 目标下 K-DABH 与静态划分决策块策略的性能比较
Table 5　Performance comparison between K-DABh and static partition decision block strategy at minimum Makespan target

测试问题	Makespan			G_{one}/%	G_{all}/%
	K-DABH	K-DABH-ONE	K-DABH-ALL		
J5M6C3	3 344.0	3 735.0	3 352.6	11.7	0.3
J15M8C3	4 885.0	4 888.0	4 888.2	0.1	0.1
J20M11C3	1 737.2	2 254.0	1 754.2	29.7	1.0
J40M13C5	4 700.0	4 796.0	4 820.0	2.0	2.6
J50M15C5	4 612.0	4 734.2	4 724.4	2.6	2.4
J60M16C5	5 254.2	5 342.2	5 379.4	1.7	2.4
J70M20C7	4 918.6	5 049.0	5 259.2	2.7	6.9
J80M21C7	4 364.0	4 622.6	4 707.0	5.9	7.8
J90M21C7	4 507.0	4 789.4	4 746.4	6.3	5.3
J100M25C9	6 752.0	6 943.2	6 973.6	2.8	3.3
J120M30C9	5 138.0	5 281.0	5 383.6	2.8	4.8
J140M35C11	4 412.8	4 631.0	4 748.0	5.0	7.6
J160M40C11	7 267.0	7 537.5	7 565.7	3.7	4.1
J180M45C13	6 646.4	6 946.8	7 023.8	4.5	5.7
J200M50C15	6 241.2	6 637.6	6 510.0	6.3	4.3
J250M65C15	6 379.0	6 756.0	6 853.4	5.9	7.4
J300M75C15	6 002.0	6 335.2	6 467.6	5.6	7.8
J350M90C15	7 007.0	7 301.6	7 489.0	4.2	6.9
J400M100C15	9 606.2	10 785.4	10 809.8	12.3	12.5
J450M120C15	9 408.2	11 467.4	11 511.4	21.9	22.4
平均值				6.9	5.8

3.4 与基于动态决策块的方法比较

K-DABH 是基于 DABH 算法的一种改进。文献[20]在解决跨单元调度问题中采用了动态决策块思想的 DABH 算法。DABH 算法主要是用智能算法将所有实体划分为适当大小的决策块，在保证优化性能的前提下，通过增大计算粒度有效提升了算法的计算效率。而 K-DABH 在 DABH 算法的基础上，进一步考虑决策块内部实体的属性差异，采用聚类思想将具有相似属性的工件归入同一个决策块内，使得决策块间工件属性差异相对较大，有利于为一类实体选择合适的规则。

为确保实验公平性，对比实验的问题规模和候选规则集相同，种群数目为120，迭代次数设置为200次。

此次实验将 DABH 与 K-DABH 之间 Gap 值记为 G_{DABH}。实验结果如表 6 所示，当问题规模发生变化时，K-DABH 在 Makespan 性能方面平均提高了 3.1%。

在算法的收敛性对比实验中，图 8 体现了当问题规模为 J160M40C11 下 2 种方法运行 200 代的收敛情况。根据图 8 所示得知，K-DABH 在第 50 代后趋于稳定，DABH 在第 101 代之后趋于稳定，K-DABH 的收敛速度更快，对于解的搜索跨度也比 DABH 要大，展示出 K-DABH 在搜寻局部最优解能力上的优势。

实验结果表明，K-DABH 在继承了 DABH 的计算优势之上，有效提升了算法的优化性能和收敛速度。分析其原因，主要是因为 K-DABH 考虑决策块内部实体属性的差异，根据属性进行实体聚类形成决策块，这有利于实现属性与规则之间的映射，为决策块分配更加合理的规则，从而提高了算法的寻优能力。

表 6　K-DABH 与 DABH 方法的性能比较
Table 6　Comparison between K-DABH and DABH

测试问题	Makespan		G_{DABH}/%
	K-DABH	DABH	
J40M13C5	4 700.0	4 727.2	0.6
J50M15C5	4 612.0	4 690.6	1.7
J60M16C5	5 254.2	5 274.4	0.4
J70M20C7	4 918.6	5 018.6	2.0
J80M21C7	4 364.0	4 454.2	2.1
J90M21C7	4 507.0	4 650.4	3.2
J100M25C9	6 752.0	6 977.4	3.3
J120M30C9	5 138.0	5 316.2	3.5
J140M35C11	4 412.8	4 725.4	6.6
J160M40C11	7 267.0	7 514.0	3.4
J180M45C13	6 646.4	6 894.0	3.7
J200M50C15	6 241.2	6 372.4	2.1
J250M65C15	6 379.0	6 667.2	4.5
J300M75C15	6 002.0	6 184.6	3.0
J350M90C15	7 007.0	7 281.6	3.9
J400M100C15	9 606.2	10 064.6	4.8
J450M120C15	9 408.2	9 769.8	3.8
平均值			3.1

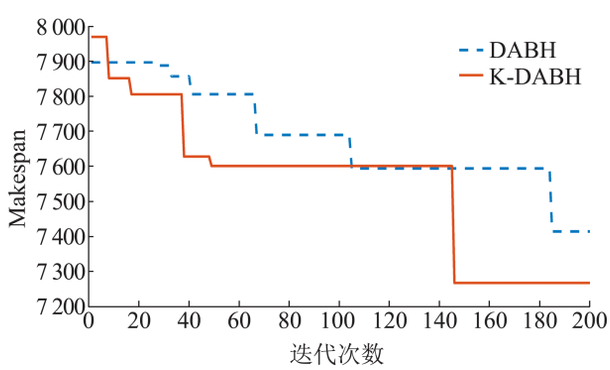

图 8　DABH 与 K-DABH 的收敛过程比较
Fig. 8　Evolutionary processes of DABH and K-DABH

4　结论

本文针对跨单元调度问题，提出一种基于动态聚类决策块的超启发式算法，该算法基于 DABH 算法进行改进，其中，决策块的大小由轮廓系数动态决定，决策块的组成由 K-means 聚类生成，将相似的实体归类并进行规则的匹配，在保证计算效率的同时提升算法优化性能。实验结果表明：决策块方法可以适当减少搜索空间，进而提高超启发式算法的计算效率；用轮廓系数动态确定的决策块数量要比静态确定的决策块数量更合理，能够有效提升计算效率和算法适应度；基于聚类的决策块形成方法，有利于为相近属性的实体匹配更加合理有效的规则，从而提升算法优化性能。基于计算优势和优化性能，K-DABH 对求解大规模、复杂性高的柔性车间跨单元调度问题具有很大优势。

在今后工作中，可尝试针对多属性进行聚类，以使产生的决策块保留实体的多个有效属性特征，进而提高算法的整体优化性能。

参考文献：

[1] Garza O, Smunt T L. Countering the Negative Impact of Intercell Flow in Cellular Manufacturing[J]. Journal of Operations Management, 1991, 10(1): 92-118.
[2] Kourosh Halat, Reza Bashirzadeh. Concurrent Scheduling of Manufacturing Cells Considering Sequence-dependent Family Setup Times and Intercellular Transportation Times[J]. The International Journal of Advanced Manufacturing Technology, 2015, 77(9): 1907-1915.
[3] Feng Yanling, Li Guo, Sethi S P. A Three-layer Chromosome Genetic Algorithm for Multi-cell Scheduling with Flexible Routes and Machine Sharing[J]. International Journal of Production Economics, 2018, 196: 269-283.
[4] Yan Hongsen, Wan Xiaoqin, Xiong Fuli. Integrated Production Planning and Scheduling for a Mixed Batch Job-shop Based on Alternant Iterative Genetic Algorithm[J]. Journal of the Operational Research Society, 2015, 66(8): 1250-1258.
[5] Solimanpur M, Prem Vrat, Ravi Shankar. A Heuristic to Minimize Makespan of Cell Scheduling Problem[J]. International Journal of Production Economics, 2004, 88 (3): 231-241.
[6] Maghsud Solimanpur, Atabak Elmi. A Tabu Search Approach for Cell Scheduling Problem with Makespan Criterion[J]. International Journal of Production Economics, 2013, 141(2): 639-645.
[7] Huang Z, Yang J J. A New Model for Optimization of Cell Scheduling Considering Inter-cell Movement[J]. International Journal of Simulation Modelling, 2022, 21 (1): 136-147.

[8] Tang Jiafu, Wang Xiaoqing, Iko Kaku, et al. Optimization of Parts Scheduling in Multiple Cells Considering Intercell Move Using Scatter Search Approach[J]. Journal of Intelligent Manufacturing, 2010, 21(4): 525-537.

[9] Li Dongni, Meng Xianwen, Li Miao, et al. An ACO-based Intercell Scheduling Approach for Job Shop Cells with Multiple Single Processing Machines and One Batch Processing Machine[J]. Journal of Intelligent Manufacturing, 2016, 27(2): 283-296.

[10] Li Dongni, Wang Yan, Xiao Guangxue, et al. Dynamic Parts Scheduling in Multiple Job Shop Cells Considering Intercell Moves and Flexible Routes[J]. Computers & Operations Research, 2013, 40(5): 1207-1223.

[11] 湛荣鑫, 李冬妮, 马涛, 等. 面向快速响应的赛汝生产系统构建模型与方法[J]. 自动化学报, 2022, 48(12): 2922-2930.
Zhan Rongxin, Li Dongni, Ma Tao, et al. Configuration Model and Approach of a Seru Production System for Quick Response[J]. Acta Automatica Sinica, 2022, 48 (12): 2922-2930.

[12] 田园, 田云娜, 刘雪. 求解柔性作业车间调度问题算法综述[J]. 延安大学学报(自然科学版), 2021, 40(3): 64-70.
Tian Yuan, Tian Yunna, Liu Xue. Review on Algorithms for Flexible Job Shop Scheduling Problem[J]. Journal of Yan'an University(Natural Science Edition), 2021, 40(3): 64-70.

[13] Cowling P, Kendall G, Soubeiga E. A Parameter-free Hyperheuristic for Scheduling a Sales Summit[C]// Proceedings of the 4th Metaheuristic International Conference. [S.l.]: [s.n.], 2001: 127-131.

[14] Anna Karen Gárate-Escamilla, Ivan Amaya, Jorge M Cruz-Duarte, et al. Identifying Hyper-heuristic Trends Through a Text Mining Approach on the Current Literature[J]. Applied Sciences, 2022, 12(20): 10576.

[15] 贾凌云, 李冬妮, 田云娜. 基于混合蛙跳和遗传规划的跨单元调度方法[J]. 自动化学报, 2015, 41(5): 936-948.
Jia Lingyun, Li Dongni, Tian Yunna. An Intercell Scheduling Approach Using Shuffled Frog Leaping Algorithm and Genetic Programming[J]. Acta Automatica Sinica, 2015, 41(5): 936-948.

[16] 李冬妮, 贾晓宇, 陈琳, 等. 基于蚁群算法和遗传规划的跨单元调度方法[J]. 北京理工大学学报, 2017, 37(7): 704-710.
Li Dongni, Jia Xiaoyu, Chen Lin, et al. Intercell Scheduling Approach Based on Ant Colony Optimization Algorithm and Genetic Programming[J]. Transactions of Beijing Institute of Technology, 2017, 37(7): 704-710.

[17] 罗敏. 基于超启发式算法的多模具限制柔性作业车间问题研究[D]. 杭州: 浙江大学, 2021.
Luo Min. Research on Hyper-heuristic Approach for Molds-constrained Flexible Job-shop Problem[D]. Hangzhou: Zhejiang University, 2021.

[18] José Antonio Vázquez-Rodríguez, Petrovic S. A New Dispatching Rule Based Genetic Algorithm for the Multi-objective Job Shop Problem[J]. Journal of Heuristics, 2010, 16(6): 771-793.

[19] 刘兆赫, 李冬妮, 王乐衡, 等. 考虑运输能力限制的跨单元调度方法[J]. 自动化学报, 2015, 41(5): 885-898.
Liu Zhaohe, Li Dongni, Wang Leheng, et al. An Inter-cell Scheduling Approach Considering Transportation Capacity Constraints[J]. Acta Automatica Sinica, 2015, 41(5): 885-898.

[20] 刘兆赫. 超启发式方法的动态决策块机制及其在调度问题中的应用[D]. 北京: 北京理工大学, 2016.
Liu Zhaohe. Hyper-heuristic with Dynamic Decision Blocks and Its Application in Scheduling Problems[D]. Beijing: Beijing Institute of Technology, 2016.

[21] 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 Yunna, Li Dongni, Liu Zhaohe, et al. A Hyper-heuristic Approach with Dynamic Decision Blocks for Inter-cell Scheduling[J]. Acta Automatica Sinica, 2016, 42(4): 524-534.

[22] Colorni A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies[C]//1991 1st Proceedings of the First European Conference on Artificial Life. [S.l.]: [s.n.], 1991: 134-142.

[23] Marco Dorigo, Luca Maria Gambardella. Ant Colonies for the Travelling Salesman Problem[J]. Bio Systems, 1997, 43(2): 73-81.

[24] Jain A K, Murty M N, Flynn P J. Data Clustering: A Review[J]. ACM Computing Surveys, 1999, 31(3): 264-323.

[25] 杨俊闯, 赵超. K-means聚类算法研究综述[J]. 计算机工程与应用, 2019, 55(23): 7-14, 63.
Yang Junchuang, Zhao Chao. Survey on K-means Clustering Algorithm[J]. Computer Engineering and Applications, 2019, 55(23): 7-14, 63.

[26] Peter J Rousseeuw. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis[J]. Journal of Computational and Applied Mathematics, 1987, 20: 53-65.

[27] Ramze Rezaee M, Lelieveldt B P F, Reiber J H C. A New Cluster Validity Index for the Fuzzy C-mean[J]. Pattern Recognition Letters, 1998, 19(3/4): 237-246.

[28] Huang P Y. A Comparative Study of Priority Dispatching Rules in a Hybrid Assembly/Job Shop[J]. International Journal of Production Research, 1984, 22(3): 375-387.

[29] Vepsalainen A P J, Morton T E. Priority Rules for Job Shops with Weighted Tardiness Costs[J]. Management Science, 1987, 33(8): 1035-1047.

DOI: 10.16182/j.issn1004731x.joss.22-1541