DOI:10.13878/j.cnki.jnuist.20230905003

曾亮[1]　石俊洋[1]　胡迈[1]　李明[1]

用于分布式置换流水变速车间的双种群算法

摘要

针对加工速度可变的分布式置换流水车间调度问题,以最大完工时间和机器总能量消耗为优化目标,提出了一种双种群算法.首先,采用混合四种策略的初始化方法来生成高质量的初始种群.其次,针对两个种群的特点分别设计了特定的进化方式,并引入了动态引导因子调整种群的进化方式.同时,提出调速节能策略,进一步优化能量消耗.最后,提出动态种群策略用于平衡两个种群的资源.通过仿真实验证明了各个策略的有效性,并与其他算法进行了对比,结果表明所提出的算法具有明显的优越性.

关键词

置换流水车间;双种群算法;分布式;变速车间;多目标优化

中图分类号 TP18

文献标志码 A

收稿日期 2023-09-05

资助项目 国家重点研发计划项目(2018YFC0116100);湖北省教育厅科学研究计划重点项目(D20211402);湖北省重点研发计划项目(2020BAB114,2023BAB094)

作者简介

曾亮,男,教授,研究方向为机器视觉与人工智能、优化计算方法、调度优化.zengliang@hbut.edu.cn

1　湖北工业大学　电气与电子工程学院,武汉,430068

0　引言

在制造业中,合理的调度能够大幅地提高生产效率、提高资源利用率,减少浪费.置换流水车间调度问题(Permutation Flowshop Scheduling Problem,PFSP)作为一种典型调度问题,在大规模制造生产中得到广泛应用.通过将生产过程分为多个步骤,并有序地安排在专门设计的生产线上,PFSP 显著提高了生产效率.然而,当机器的数量大于3个时,该问题已经被证明是 NP 难问题[1].目前,设计有效的元启发式算法是求解 PFSP 的主流方法.

随着工业技术和全球贸易的不断发展,企业订单数量急剧上升,生产商意识到分布式生产模式的多工厂协同才能满足市场和客户需求,适应不断变化的生产环境[2].分布式调度可让多个工厂协同生产,其关键优势在于其较高的生产效率和资源利用率[3].Naderi 等[4]首先提出了分布式置换流水车间调度问题(Distributed Permutation Flowshop Scheduling Problem,DPFSP).DPFSP 需要将作业分配给工厂,并确定每个工厂内作业的加工顺序,相较于 PFSP 更加复杂.为了解决 DPFSP,Li 等[5]提出一种改进的人工蜂群算法,设计了基于工作量平衡的初始化方法和局部搜索策略,通过实验测试,证明了算法搜索的高效性;此外,Zhao 等[6]改进了水波算法并设计了4种局部搜索方法,在优化具有阻塞的 DPFSP 时表现出优异性.

然而,针对 DPFSP 的相关研究显示,许多学者并没有考虑到加工速度对调度的影响,或者假设速度是恒定的.实际生产中,制造业的智能化发展加速,机器的加工速度灵活化成为了满足多种材料特性和工艺需求的必要条件.可变加工速度的 DPFSP(Distributed Permutation Flowshop Scheduling Problem with Variable Processing Speed,DPFSP-VPS)可根据实时情况进行动态调度,从而提高生产效率并降低成本,有效地减少制造过程中的等待时间和排队时间[7],展现出了巨大的应用潜力.如今,学者们逐渐意识到 DPFSP-VPS 研究的重要性,例如:Huang 等[8]指出数控机床变速加工的重要性,并提出一种双角色协同进化算法,设计了针对机器速度的多种局部搜索策略,实验结果表明了算法的优越性;Wang 等[9]提出一种基于知识的协同进化算法,设计了多个协同搜索策略,并采用了基于关键路径的方法来进一步提高算法性能;Wang 等[10]提出一种针对 DPFSP-VPS 的改进多目标鲸鱼群算法,能够在不影响生产效率的情况下提高能源效率,并针对完

工时间和机器速度提出了更新利用机制.

同时,随着资源的过度消耗和全球变暖形势的日益严峻,提高环境的可持续性已成为一个迫切的问题,环保组织已开始提倡各种制造领域的节能降耗[11].在此背景下,企业在调度中考虑能耗已经成为转型升级的必然趋势.不仅在 PFSP[12] 中,在混合车间[13] 和柔性作业车间[14] 等领域,节能调度也得到了广泛的研究.本文的研究主要聚焦于同时考虑最大完工时间和总能量消耗两个目标的 DPFSP-VPS.两个目标的优化带来额外的困难,如何在算法中设计优秀的种群更新方式和有效策略成为研究的重点.

针对上述问题,本文提出一种双种群算法(Dual-Population Algorithm,DPA)来求解双目标的 DPFSP-VPS.尽管双种群概念已在多个领域有相关研究,但在 DPFSP-VPS 领域的应用极少.与以往研究不同的是,本文提出的 DPA 首先针对每个种群的特性设计了不同的进化方式;其次,完全舍弃了其他算法中固定的交叉变异概率,通过引入动态引导因子来指导算法的进化方向;同时,两个种群在更新的过程中种群数量也在动态变化,这种灵活性能够平衡两种群的计算资源;最后,设计了变速节能策略来降低调度中能量消耗.实验结果表明,本文所提算法在解决 DPFSP-VPS 问题上高效可行,相较于其他算法具有较大的竞争性.

1 分布式置换流水变速车间调度问题

1.1 问题描述

DPFSP-VPS 可描述为 n 个工件,在 F 个工厂加工.每个工厂都是一个包含 m 个机器的置换流水车间问题,即每个工件都包含 m 道工序.在 DPFSP-VPS 中,首先为每个作业分配工厂,确定工厂内作业的处理顺序,然后还需要为每个操作选择机器的处理速度.每道工序 $O_{i,j}$ 在每个机器上均有一个标准的处理时间 $t_{i,j}$,若 $O_{i,j}$ 的处理速度为 V_v,则其真实的处理时间为 $p_{i,j}=t_{i,j}/V_v$,其加工能耗与 $p_{i,j}$ 和 V_v^2 成正比.因此,加工速度越大,加工时间越短,但加工能耗越大.

关于 DPFSP-VPS 的假设如下:

1)所有机器可在零时刻开始加工;

2)每个工件可在任一工厂加工,但标准加工时间不同,具有异构性;

3)每个工件选定工厂后,则其所有工序均在此工厂加工完成;

4)一个作业的任何操作必须完成其前面的操作后才能进行;

5)不考虑工件加工过程中发生中断,不考虑机器发生故障.

1.2 数学模型

为了便于描述,对研究中所使用的符号进行了如下定义:

1)索引

j:机器索引,$j \in \{1,\cdots,m\}$;

f:工厂索引,$f \in \{1,\cdots,F\}$;

v:速度索引,$v \in \{1,\cdots,s\}$;

i:工件索引,$i \in \{1,\cdots,n\}$.

2) 参数

n:工件的数量;

m:每个工厂的机器数量;

F:工厂数量;

s:加工速度的数量;

l:按顺序排列的工件位置;

V_v:第 v 个处理速度;

$O_{i,j}$:第 i 个工件的第 j 道工序.

3) 决策变量

$t_{i,j}$:$O_{i,j}$ 的标准处理时间;

$p_{i,j}$:$O_{i,j}$ 的真实处理时间;

$PP_{f,j,v}$:工厂 f 中以速度 v 运行的机器 j 的单位能耗;

$SP_{f,j}$:工厂 f 中机器 j 单位时间的待机能耗;

$PEC_{f,j}$:工厂 f 中机器 j 的加工能耗;

$SEC_{f,j}$:工厂 f 中机器 j 的空闲能耗;

TEC:总能耗;

$C_{\max}$:调度的最大完工时间;

$C_{l,j,f}$:工厂 f 中机器 j 上位置 l 的完成时间;

$C(f)$:工厂 f 的最大完工时间;

$x_{i,l,f}$:二进制变量,如果作业 i 占用工厂 f 的位置 l,则值为 1,否则为 0;

$z_{i,j,v}$:二进制变量,如果作业 i 在机器 j 上以速度 v 处理,则值为 1,否则为 0.

DPFSP-VPS 包括最大完工时间(Makespan)和总能量消耗(TEC)两个目标.Makespan 和 TEC 是两个矛盾的目标,更小的 Makespan 意味更大的加工速度.但是,更大的加工速度会带来更大的加工能耗.因此,为了更好地平衡两个目标,需要有优秀的调度方案,以及合适的速度选择.

第一个目标 Makespan 如式(1)所示:

$$\text{Makespan} = C_{\max} = \max_f C(f), \quad \forall f. \tag{1}$$

第二个目标 TEC 计算方法见式(2).TEC 包括每个工厂中所有机器的加工能耗(PEC)和空闲能(SEC),PEC 和 SEC 的计算方法分别如式(3)和式(4)所示.

$$\text{TEC} = \sum_{f=1}^{F}\sum_{j=1}^{m}(\text{PEC}_{f,j} + \text{SEC}_{f,j}), \tag{2}$$

$$\text{PEC}_{f,j} = \sum_{i=1}^{n}\sum_{l=1}^{n}\left(x_{i,l,f}\cdot p_{i,j}\cdot\sum_{v=1}^{s} z_{i,j,v}\cdot \text{PP}_{f,j,v}\right), \quad \forall f,j, \tag{3}$$

$$\text{SEC}_{f,j} = \left(C(f) - \sum_{i=1}^{n} p_{i,j}\cdot\sum_{i=1}^{n} x_{i,l,f}\right)\cdot \text{SP}_{f,j}, \quad \forall f,j. \tag{4}$$

根据上文的假设和实际的生产情况,DPFSP-VPS 的约束如下:

$$\sum_{l=1}^{n}\sum_{f=1}^{F} x_{i,l,f} = 1, \quad \forall i, \tag{5}$$

$$\sum_{i=1}^{n}\sum_{f=1}^{F} x_{i,l,f} = 1, \quad \forall i, \tag{6}$$

$$\sum_{v=1}^{s} z_{i,j,v} = 1, \quad \forall i,j, \tag{7}$$

$$p_{i,j} = t_{i,j}\cdot\sum_{v=1}^{s}\frac{z_{i,j,v}}{V_v}, \quad \forall i,j, \tag{8}$$

$$C_{l,j,f} = \sum_{i=1}^{n} x_{i,l,f}\cdot p_{i,j}, \quad l=1, j=1, \forall f, \tag{9}$$

$$C_{l,i,f} \geqslant C_{l,j-1,f} + \sum_{i=1}^{n} x_{i,l,f}\cdot p_{i,j}, \quad \forall l,j,f, \tag{10}$$

$$C_{l,i,f} \geqslant C_{l,j-1,f} + \sum_{i=1}^{n} x_{i,l,f}\cdot p_{i,j}, \quad \forall l>1,j,f, \tag{11}$$

$$C(f) \geqslant C_{l,j,f}, \quad \forall l,f, \tag{12}$$

$$C_{l,j,f} \geqslant 0, \quad \forall l,j,f. \tag{13}$$

式(5)确保每个工件只能分配给一个工厂;式(6)确保每个机器同一时刻只加工一个工件;式(7)确保每道工序只能在一个速度下进行加工;式(8)表示 $O_{i,j}$ 的实际处理时间;式(9) 表示工厂中第一台机器从零时刻开始加工;式(10) 确保 $O_{i,j}$ 在 $O_{i,j-1}$ 完工后才可加工;式(11)确保每台机器不能同时加工多个工序;式(12)和(13)表示每个工厂的加工时间约束.

2 双种群算法求解 DPFSP-VPS

2.1 编码解码

对于 DPFSP-VPS,包括 3 个子问题:1)确定每个工件加工的工厂;2)确定每个工厂内工件的处理顺序;3)确定每道工序在每个机器上的处理速度.本研究根据 3 个子问题的特点采用三段式编码,即工件编码(JS)、工厂编码(FS)、速度编码(VS).表 1 给出了一个 7 个工件、2 个工厂和 3 种速度的 DPFSP-VPS 示例.根据 JS 和 FS 可知:有 4 个工件在工厂 1 处理,加工顺序为{1,5,3,7};有 3 个工件在工厂 2 处理,加工顺序为{6,2,4}.每一个工件有 3 道工序,对应每个工厂内的 3 台机器,根据 VS 可知每道工序的加工速度.假设每道工序的 $t_{i,j}$ 均为 3 h,则根据表 1 生成的调度甘特图如图 1 所示.

解码时,首先根据 FS 将所有工件分配给各个工厂,然后依照 JS 中的工件加工顺序安排每个工厂内的工序加工序列,最后根据 VS 确定每道工序的加工速度.通过标准处理时间除以加工速度,可以得到真实的处理时间,并根据加工能耗与速度的平方正相关原则计算出总能耗.

表 1 一个关于 DPFSP-VPS 的例子

Table 1 An example of DPFSP-VPS

JS	1	5	6	3	2	7	4
FS	1	1	2	1	2	1	2
	3	1	3	1	2	3	1
VS	1	1	1	3	2	2	2
	2	2	2	1	1	2	2

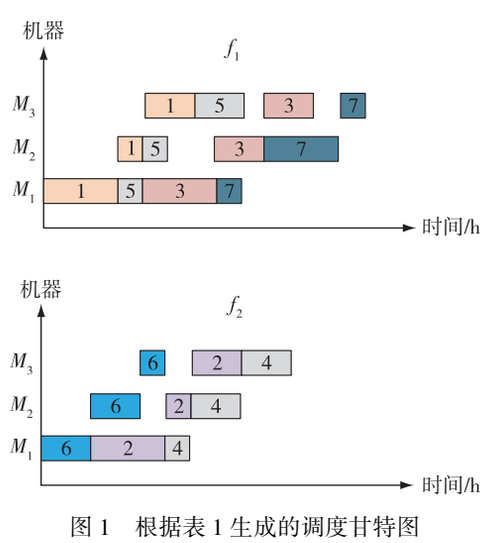

图 1 根据表 1 生成的调度甘特图

Fig. 1 Gantt chart of scheduling generated according to table 1

2.2 混合种群初始化

有效的初始化方法能够显著提高初始种群的质量,不仅有利于算法的快速收敛,还会提高最终解的质量.本文根据不同的优化目标提出了 4 种初始化

策略,最终由每种策略各生成 1/4 个体组成混合种群.

1) DNEH 原则:DNEH (Distributed Nawaz Enscore Ham)[15]是在 NEH 基础上针对最小化流经时间的 DPFSP 提出的.本文对其进行改进,应用于多目标 DPFSP.首先将所有工件的加工时间降序排列,假设为{3,2,4,1}.若有 2 个工厂,则将加工时间最长的工件 3 和工件 2 分别分配给 2 个工厂,保证每个工厂均有工件.至于工件 4 和工件 1 则逐一插入到任何可能的位置,并进行非支配排序,最后保留最好的方案.

2)完工时间优先原则:首先随机初始化 JS 和 FS,然后将所有操作的 VS 设为最大.此操作能够使完工时间最小化.

3)加工能耗优先原则:执行速度选择时与策略 2)相反,所有操作的 VS 被设为最小,以最小化加工能耗.

4)随机初始化:初始种群的多样性同样对最终的解方案至关重要[16].因此,本研究的一部分个体采用随机初始化策略,即随机初始化 JS、FS 和 VS,这将有利于丰富种群的多样性.

2.3 双种群进化

本文提出的算法将所有个体分为 2 个种群,分别命名为精英种群(Elite Population,EP)和普通种群(Normal Population,NP).在种群初始化完成后,根据非支配解集和拥挤度进行排序,初始时精英种群包含排名前 50%的个体,剩余个体归为普通种群.算法中针对不同种群设计了不同的进化方式.

2.3.1 EP 进化

EP 的进化方式有 2 种,如式(14)所示.其中,GF 为引导因子,可对个体的进化方式进行引导.

$$EP_{new}=\begin{cases}EP\ 自我进化, & Rand>GF,\\ EP\ 局部搜索, & 其他.\end{cases} \tag{14}$$

EP 中均是拥有优良基因的个体,因此,对其执行自我进化有利于优良基因的融合交叉,从而加速收敛过程.对于 EP 中的某个个体 EP_1,其自我进化方式为随机选取另一个体 EP_2 进行交叉进化.每个个体均包含 3 段编码的信息,在进化过程中,针对 JS 采用部分匹配交叉(Partial Match Crossover,PMX).图 2a 给出了一个 PMX 的示例.首先,对 2 个个体随机选择 2 个不同的点并交换 2 点之间的排列生成 2 个后代;然后,对个体内存在的非法解进行映射,通过映射将重复的非法解替换成另一个体相同位置的工件,若映射后仍存在非法解,则再次进行映射.由于每个工件的工厂分配和速度设定不存在相关性,速度也不存在排列约束,因此,针对 FS 和 VS 采用统一交叉(Uniform Crossover,UX).UX 即随机选取 2 个个体的对应的编码进行交换,图 2b 给出了一个关于 FS 的 UX 交叉的示意图.最终通过 EP_1 和 EP_2 的自我进化,产生 2 个新个体 EP_{new1} 和 EP_{new2}.

在进化过程中,EP 会逐渐靠近帕累托前沿,可

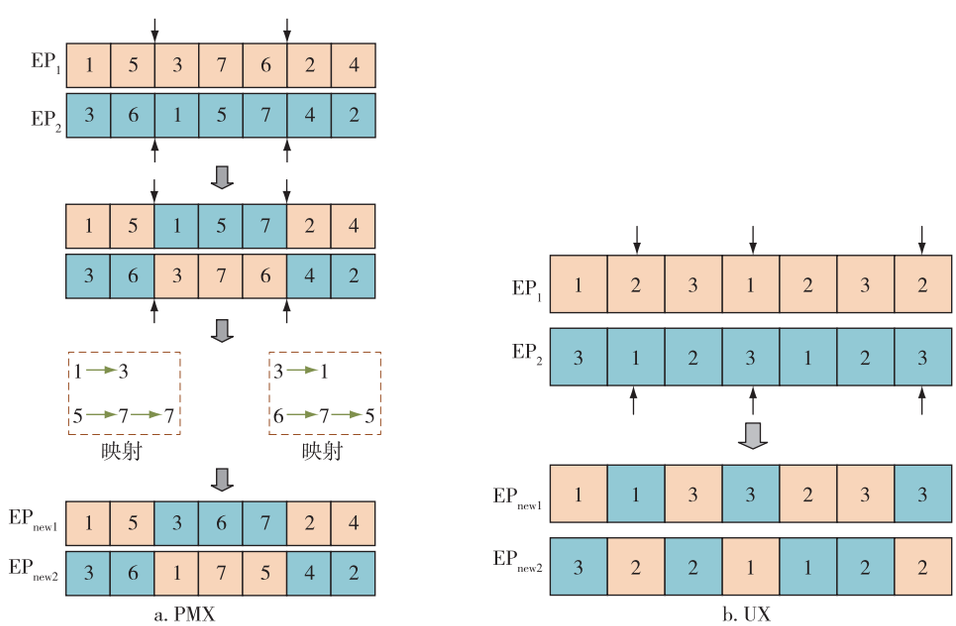

图 2 PMX 和 UX

Fig. 2 PMX and UX

能会陷入局部最优.因此,本文针对 EP 设计了 4 种针对关键工厂的局部搜索.关键工厂即完工时间最长的工厂,关键路径则为从开始到结束历时最长的加工路径.对关键路径中的关键操作进行优化,将有利于跳出局部最优,找到更优解.4 种局部搜索如下所示:

1)LS1:关键工厂内插入.在关键工厂内,随机选择一个关键操作,将其插入其他可能的位置.

2)LS2:关键工厂内交换.在关键工厂内,随机选择两个关键操作,交换它们的位置.

3)LS3:关键工厂外插入.从关键工厂内,随机选择一个关键操作,将其插入到其他工厂可能的位置.

4)LS4:关键工厂内加速.随机选择一个关键操作,增大其处理速度.这将有利于降低 Makespan.

2.3.2 NP 进化

NP 的进化方式也有两种,如式(15)所示:

$$NP_{new}=\begin{cases}NP\ 协同进化, & Rand>GF,\\ NP\ 变异搜索, & 其他.\end{cases} \tag{15}$$

NP 中为普通个体,为了加快其收敛速度,将 NP 与 EP 进行协同进化.具体而言,NP 中的个体会随机选择 EP 中的一个个体作为进化引导对象.进化方法仍是对 JS 使用 PMX,对 FS 和 VS 使用 UX.通过这种协同进化,可以避免算法盲目搜索,加快搜索过程.

变异是增加种群多样性的重要方式,NP 中本就是较差的个体,对其进行变异会有更大的概率搜索到更优解.根据三段式编码的特点,设计了相应的变异方式.对于 JS,变异是随机交换两个位置的工件.对于 FS,变异是随机选择一个工件重新分配给另一个工厂.对于 VS,变异是随机选择一个操作并改变其处理速度.

2.4 动态引导因子

EP 和 NP 的进化方式均受到引导因子 GF 的影响.为了在种群迭代的过程中动态调整 2 个种群的进化方式,对 GF 的大小进行动态调整,如式(16)所示:

$$GF=GF_{min}+(GF_{max}-GF_{min})\frac{Iter}{maxIter}, \tag{16}$$

其中:GF_{min} 和 GF_{max} 分别为 GF 的最小值和最大值,本文设置 $GF_{min}=0.2$,$GF_{max}=0.8$;Iter 和 MaxIter 分别为当前迭代次数和最大迭代次数.在迭代的前期,GF 较小,2 个种群将进行充分的自我进化和协同进化,此阶段是算法的快速收敛阶段.迭代过程中,GF 不断增大,2 个种群将更倾向于进行局部搜索和变异搜索.后期较大的 GF 可使 EP 通过局部策略进行精细搜索,跳出局部最优,而 NP 则可通过变异增加种群多样性.

通过 GF 的动态引导,可调节 2 个种群的分工,让算法在不同阶段利用不同的优化策略.同时,算法能够在不同迭代阶段更加灵活地进行搜索,从而提高算法的性能和收敛速度.

2.5 调速节能策略

总能耗是 DPFSP-VPS 的另一个重要目标,如何设计有效的节能技术是节能调度的关键.在 DPFSP-VPS 中,加工速度是造成 Makespan 和 TEC 两个子目标相互冲突的关键因素.因此,本文设计了两种针对空闲时间的调速节能策略.这两种策略能够调整机器的加工速度,可以在不影响 Makespan 的情况下降低能耗.

1)减速策略:当同一台机器上的某个操作的开始时间大于上一个操作的结束时间时,就会产生空闲时间段.因此,可以降低空闲时间段前操作的加工速度,这样在减少空闲时间的同时还能降低加工能耗.如图 3 所示,将 f_1 中 $O_{5,2}$,$O_{5,3}$,$O_{3,3}$ 加工速度变慢一档,空闲时间 T_1 和 T_2 消失,T_3 减少一半.为了不影响 Makespan,此操作只在非关键路径上进行.

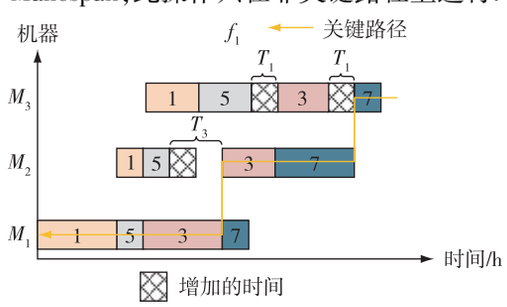

图 3 减速节能策略

Fig. 3 Energy-saving strategy via deceleration

2)加速策略:某个工件的上一道工序加工时间过长同样会产生空闲时间段.例如图 1 中 f_2 的 $O_{2,1}$ 加工时间过长造成 $O_{2,2}$ 的开始时间晚于 $O_{6,2}$ 的结束时间,产生了一段空闲时间.因此,适当地增大 $O_{2,1}$ 的加工速度,缩短其加工时间,即可提前 $O_{2,2}$ 的开始时间,减少空闲时间.如图 4 所示,增大 $O_{2,1}$ 加工速度后既减少了空闲时间还降低了 Makespan.在加速后会降低空闲能耗,但也会造成加工能耗增加,所以,此方法调速后需要判断新解是否支配旧解.若新解支配旧解,则保留新解,否则旧解被保留.

减速策略能延长空闲时间段前一道工序的完工

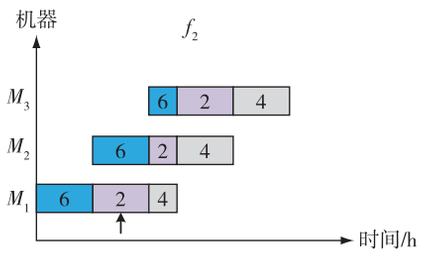

图 4　加速节能策略

Fig. 4　Energy-saving strategy via acceleration

时间,加速策略能提前空闲时间段后一个工序的开始加工时间.两种方式均可降低 TEC,且不会增大 Makespan.对于以上两种策略,设定各自随机选取概率为 0.5.另外,为避免复杂的运算,规定只有迭代超过总迭代次数的 90%时才执行调速策略.

2.6　动态种群

两个种群进化后,它们都会生成新的个体.这些新个体将与前一代的个体一同进行评价,根据拥挤度进行非支配排序.为保持种群数量的稳定,排名靠后的个体被淘汰,其余个体被选择并进入下一次迭代.

在双种群机制中,如何为两个种群分配合适的个体资源很重要.因此,本文设计了一种正反馈动态种群分配机制,如式(17)所示:

$$DP=\frac{|POP_{new}|}{|POP|}. \tag{17}$$

其中:$|POP_{new}|$为某个种群本代进化后被选择的个体数量;$|POP|$为该种群的个体总数.某个种群进化后,新生成个体被选择的比例越大,DP 越大.通过计算两个种群的 DP 值,为取得较大 DP 值的种群在进入下一次进化时,额外分配 5%的个体.通过计算两个种群的子代存活率,可以判断出当前哪个种群的进化效果更好.基于正反馈机制,为进化效果好的种群分配更多的个体,从而让其获得更多的资源,能够进一步提高算法的性能.

2.7　算法框架

本文提出了一种解决 DPFSP-VPS 的双种群算法,并为每个种群设计了不同的进化方式,算法的整体流程框架如图 5 所示.

3　实验结果与分析

3.1　实验介绍

目前尚未有关于 DPFSP-VPS 的开源基准.本研究参考文献[8]和[9],基于 Taillard 基准生成了 16 个不同规模的实例,以 *n_m_F* 方式命名,具体的实例参数

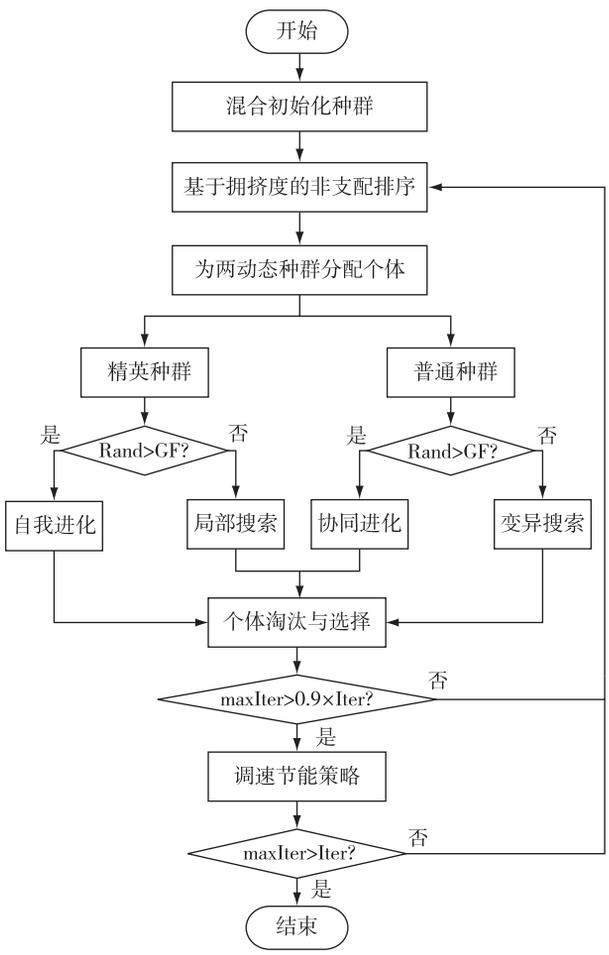

图 5　算法流程

Fig. 5　Flow of the dual-population algorithm for DPFSP-VPS

如表 2 所示.在异构工厂中,每个操作在不同工厂中的处理时间不同,工序的标准处理时间符合均匀分布 $U(1,99)$.种群数量和迭代次数根据以往研究中的主流选择,分别设置为 100 和 200.所有代码使用 Matlab2019b 编程,软件的运行环境为:CPU:Intel(R) Core(TM) i9-9900K @3.60 GHz;RAM:64 GB,Win10.

表 2　数据集分布

Table 2　Distribution of the dataset

变量	数值
工件数量(n)	20,50,100,200
工厂数量(F)	2,3
机器数量(m)	10,20
加工速度(V_v)	1.0,1.3,1.55,1.75,2.1
标准处理时间($t_{i,j}$)/h	符合(1,99)内的均匀分布
加工功率/(kW·h)	2.0
空闲功率/(kW·h)	1.0

由于真实的帕累托前沿未知,本文采用超体积

度量(Hypervolume,HV)[17]作为评价指标.HV 可以同时评价解的多样性和收敛性,HV 值越大说明算法的综合性能越好.本文对每个问题均独立运算 20 次,以确保结果的可靠性.

3.2 策略有效性验证

表 3 展示了不同算法所拥有的策略.为了验证各个策略对算法性能的影响,将这些算法进行对比测试.表 4 展示了算法 20 次运行求得的平均 HV 值,加粗体为最优值.可以看出,随着策略的增加,算法的性能逐步提升.包含所有策略的 DPA 取得了最多的最优值,证实了各个策略对算法的性能均有提升.

表 3 拥有不同策略的算法

Table 3 Algorithms with different strategies

算法名称	算法特性
DPA1	无任何策略
DPA2	混合初始化
DPA3	混合初始化+局部搜索
DPA	混合初始化+局部搜索+调速节能策略

表 4 拥有不同策略算法的平均 HV

Table 4 Average HV for algorithms with different strategies

实例	DPA1	DPA2	DPA3	DPA
20_10_2	0.055 9	0.221 6	**0.231 1**	0.230 0
20_20_2	0.076 4	0.210 0	0.210 3	**0.221 8**
50_10_2	0.024 0	0.212 9	0.223 5	**0.223 8**
50_20_2	0.030 5	0.194 5	0.223 2	**0.230 1**
100_10_2	0.081 9	0.190 4	0.220 3	**0.222 5**
100_20_2	0.020 7	0.175 7	**0.218 2**	0.215 5
200_10_2	0.063 4	0.175 2	0.207 8	**0.209 4**
200_20_2	0.084 3	0.136 7	0.204 4	**0.205 6**
20_10_3	0.046 8	**0.225 7**	0.215 0	0.218 0
20_20_3	0.083 3	0.211 1	0.209 4	**0.218 8**
50_10_3	0.027 9	**0.224 7**	0.221 6	0.220 7
50_20_3	0.096 4	**0.211 5**	0.205 2	0.211 2
100_10_3	0.019 9	**0.216 9**	0.208 5	0.212 9
100_20_3	0.022 3	0.198 4	0.210 0	**0.217 9**
200_10_3	0.054 3	0.200 8	0.216 7	**0.221 7**
200_20_3	0.075 5	0.175 2	0.203 7	**0.214 1**

注:黑体为最优值.

3.3 对比实验

为了进一步验证 DPA 求解 DPFSP-VPS 的有效性,将其与主流的多目标进化算法进行对比,包括 NSGA-Ⅱ、MOEA/D、SPEA2 和专门用于 DPFSP-VPS 的新颖算法:BRCE[8].表 5 展示了所有算法的平均 HV 值(Ave)和相对标准误差(Dev),Dev 为标准差与平均值的比值,表中加粗体为最优值.

正如表 5 所示,DPA 的平均 HV 值和相对标准误差均显著优于其他对比算法,这表明 DPA 性能最佳且最稳定.此外,通过观察表中的数据,可以发现随着工件数量和机器数量的增加,算法的 Ave 均有不同程度的下降,但 DPA 的下降幅度是最小的.这主要归因于随着计算难度的增大,搜索空间不断膨胀,在有限的计算资源下更难找到高质量的解.而 EP 的自我进化能够在一部分空间进行精细搜索,NP 的协同进化实现了不同空间的信息交流,两种群不同的进化方式使得 DPA 能够对解空间进行充分的搜索.复杂的多目标问题通常具有更多的局部最优解,这使得算法更易陷入局部最优.但是,DPA 中的局部搜索以及变异搜索可有效地帮助算法跳出局部最优,避免了像其他算法那样性能大幅下降.

表 6 展示了所有算法的 Friedman 秩和检验结果,其中 DPA 排名第一.此外 P 值为 $5.440\ 2\times10^{-12}$,说明所有算法求得的结果具有明显差异.这进一步验证了 DPA 算法在求解多目标 DPFSP-VPS 时的优越性.

图 6 展示了在实例 20_10_2 上,所有算法得到的帕累托前沿.可以观察到 DPA 相比其他算法在分布性和收敛性方面表现更优.尽管在中部区域少部分解的质量稍逊于 NSGA-Ⅱ和 BRCE,但这是可接受的,因为 DPA 在两个目标的极值区域进行了更充分的探索.得益于调速节能策略的有效性,DPA 找到了能耗更低的调度方案.在实际生产中,这意味着 DPA 给出的方案更有能耗优势.在图 6 中的 A 点(827.6,26 142)附近,DPA 相比第二名 NSGA-Ⅱ(827,26 376)能耗节省了 0.9%,相比最后一名 MOEA/D(840.9,28 412)能耗节省了 8.7%,证明了 DPA 在节能方面的有效性和实用价值.

4 总结

本文提出了一种双种群算法,用于求解加工速度可变的分布式置换流水车间调度问题,同时优化最大完工时间和能量总耗.为了提高初始种群的质量,提出一种针对不同目标的混合初始化策略.针对两个种群的特点,设计了不同的进化方式,实现了对解空间的充分探索.同时,设计了动态引导因子,可在迭代过程中动态引导种群的进化方式.提出了一种新颖的调速节能策略,可在不影响最大完工时间的情况下降低能量消耗.最后,基于正反馈的动态种

表 5　所有算法对比结果

Table 5　Result comparison between all algorithms

实例	MOEA/D		NSGA-Ⅱ		SPEA2		BRCE		DPA	
	Ave	Dev	Ave	Dev	Ave	Dev	Ave	Dev	Ave	Dev
20_10_2	0.041 0	0.209 9	0.069 0	0.661 1	0.051 5	0.219 1	0.225 6	0.158 8	**0.233 1**	**0.091 5**
20_20_2	0.030 4	0.227 0	0.049 8	0.854 9	0.036 4	0.175 3	0.219 9	0.369 5	**0.223 4**	**0.098 4**
50_10_2	0.023 8	0.283 1	0.029 9	0.185 4	0.029 7	0.150 7	**0.236 0**	**0.091 7**	0.227 0	0.068 8
50_20_2	0.016 7	0.158 0	0.033 2	1.374 5	0.020 8	0.133 7	0.216 9	0.093 5	**0.225 0**	**0.062 3**
100_10_2	0.021 0	0.231 3	0.029 2	0.764 1	0.022 7	0.165 1	0.201 6	0.258 0	**0.219 0**	**0.057 7**
100_20_2	0.012 7	0.233 3	0.023 3	1.323 2	0.015 6	0.157 6	0.210 9	0.169 6	**0.216 3**	**0.054 4**
200_10_2	0.012 3	0.141 0	0.019 4	0.284 1	0.018 6	0.127 2	**0.228 1**	**0.081 1**	0.203 0	0.118 2
200_20_2	0.011 7	0.187 9	0.022 5	1.320 3	0.013 2	0.097 4	0.200 1	0.136 6	**0.210 2**	**0.086 6**
20_10_3	0.054 2	0.313 2	0.064 6	0.158 1	0.052 5	0.273 7	0.227 2	0.111 3	**0.235 0**	**0.100 6**
20_20_3	0.034 1	0.151 7	0.048 3	0.960 2	0.040 9	0.259 8	0.209 1	0.177 1	**0.216 9**	**0.113 4**
50_10_3	0.027 7	0.265 3	0.043 0	1.199 9	0.029 4	0.252 9	0.219 3	0.086 2	**0.228 7**	**0.070 9**
50_20_3	0.020 5	0.228 0	0.044 8	1.339 8	0.019 9	0.179 2	0.202 3	0.255 5	**0.207 7**	**0.084 8**
100_10_3	0.023 7	0.187 9	0.040 9	1.073 4	0.022 7	0.203 7	0.215 9	0.083 8	**0.222 7**	**0.069 3**
100_20_3	0.013 4	0.207 9	0.024 3	1.636 7	0.015 6	0.166 1	0.212 3	0.154 7	**0.227 2**	**0.066 0**
200_10_3	0.014 4	0.212 9	0.015 8	0.144 3	0.016 0	0.109 4	**0.227 5**	**0.099 8**	0.220 5	0.087 6
200_20_3	0.014 1	0.181 1	0.021 9	1.526 3	0.013 2	0.146 6	0.201 2	0.136 6	**0.209 5**	**0.082 4**

注:黑体为最优值.

表 6　Frideman 秩和检验

Table 6　Frideman rank-sum test

算法	排名	*P* 值
NSGA-Ⅱ	1.25	
MOEA/D	2.94	
SPEA2	1.81	$5.440\ 2\times10^{-12}$
BRCE	4.19	
DPA	**4.81**	

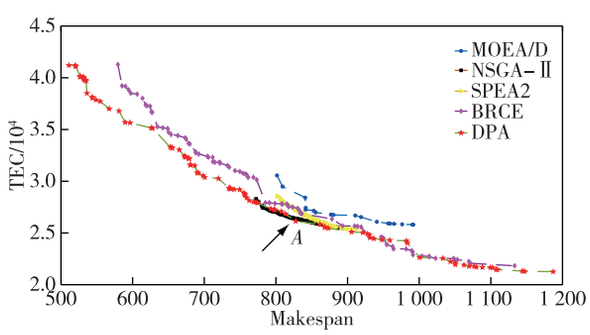

图 6　算法在 20_10_2 上的帕累托前沿

Fig. 6　Pareto fronts of algorithms on 20_10_2

群机制,合理地分配了两个种群的资源.通过实验验证了各个策略的有效性,与其他算法的对比结果,证明了本文所提出的算法具有优异的性能.

参考文献

References

[1] Garey M R,Johnson D S,Sethi R.The complexity of flowshop and jobshop scheduling[J].Mathematics of Operations Research,1976,1(2):117-129

[2] 轩华,李文婷,李冰.混合离散人工蜂群算法求解含不相关并行机的分布式柔性流水线调度[J].控制与决策,2023,38(3):779-789
XUAN Hua, LI Wenting, LI Bing. Hybrid discrete artificial bee colony algorithm for distributed flexible flowline scheduling with unrelated parallel machines[J]. Control and Decision,2023,38(3):779-789

[3] 王思涵,李新宇,高亮,等.分布式车间调度研究综述[J].华中科技大学学报(自然科学版),2022,50(6):1-10
WANG Sihan, LI Xinyu, GAO Liang, et al. A review of distributed shop scheduling problems[J].Journal of Huazhong University of Science and Technology (Natural Science Edition),2022,50(6):1-10

[4] Naderi B, Ruiz R. The distributed permutation flowshop scheduling problem [J]. Computers & Operations Research,2010,37(4):754-768

[5] Li J Q, Bai S C, Duan P Y, et al. An improved artificial bee colony algorithm for addressing distributed flow shop with distance coefficient in a prefabricated system[J].International Journal of Production Research, 2019, 57(22):6922-6942

[6] Zhao F Q, Shao D Q, Wang L, et al. An effective water wave optimization algorithm with problem-specific knowledge for the distributed assembly blocking flow-shop scheduling problem [J]. Knowledge-Based Systems,

2022,243:108471
[7] Jiang J W, An Y J, Dong Y F, et al. Integrated optimization of non-permutation flow shop scheduling and maintenance planning with variable processing speed[J]. Reliability Engineering & System Safety, 2023, 234:109143
[8] Huang K H,Li R,Gong W Y,et al.BRCE:bi-roles co-evolution for energy-efficient distributed heterogeneous permutation flow shop scheduling with flexible machine speed[J].Complex & Intelligent Systems,2023,9(5):4805-4816
[9] Wang J J,Wang L.A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop[J].IEEE Transactions on Systems,Man,and Cybernetics:Systems,2020,50(5):1805-1819
[10] Wang G C,Gao L,Li X Y,et al.Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm[J].Swarm and Evolutionary Computation,2020,57:100716
[11] 罗聪,龚文引.混合分解多目标进化算法求解绿色置换流水车间调度问题[J/OL].控制与决策:1-9[2023-07-20].https://doi.org/10.13195/j.kzyjc.2022.2145
LUO Cong,GONG Wenyin.A hybrid multi-objective evolutionary algorithm based on decomposition for green permutation flow shop-scheduling problem[J].Control and Decision:1-9[2023-07-20].https://doi.org/10.13195/j.kzyjc.2022.2145
[12] Fathollahi-Fard A M,Woodward L,Akhrif O.Sustainable distributed permutation flow-shop scheduling model based on a triple bottom line concept[J].Journal of Industrial Information Integration,2021,24:100233
[13] 吴秀丽,闫晓燕.基于改进Q学习的可重入混合流水车间绿色动态调度[J].机械工程学报,2023,59(13):246-259
WU Xiuli, YAN Xiaoyan. An improved Q learning algorithm to optimize green dynamic scheduling problem in a reentrant hybrid flow shop[J].Journal of Mechanical Engineering,2023,59(13):246-259
[14] 徐明,张剑铭,陈松航,等.柔性作业车间调度问题的多目标优化算法[J].计算机与现代化,2021(12):1-6
XU Ming, ZHANG Jianming, CHEN Songhang, et al. Multi-objective optimization algorithm for flexible job shop scheduling problem [J]. Computer and Modernization,2021(12):1-6
[15] Pan Q K,Gao L,Li X Y,et al.Effective constructive heuristics and meta-heuristics for the distributed assembly permutation flowshop scheduling problem[J].Applied Soft Computing,2019,81:105492
[16] 丁美芳,吴克晴,肖鹏.多策略融合的黄金正弦樽海鞘群算法[J].南京信息工程大学学报(自然科学版),2023,15(6):662-675
DING Meifang,WU Keqing,XIAO Peng.Golden sine salp swarm algorithm with multi-strategy [J]. Journal of Nanjing University of Information Science & Technology (Natural Science Edition),2023,15(6):662-675
[17] 王秋莲,段星皓.基于高维多目标候鸟优化算法的柔性作业车间调度[J].中国机械工程,2022,33(21):2601-2612
WANG Qiulian, DUAN Xinghao. Scheduling of flexible job shop based on high-dimension and multi-objective migrating bird optimization algorithm [J]. China Mechanical Engineering,2022,33(21):2601-2612

Dual-population algorithm for distributed permutation flowshop scheduling problem with variable processing speed

ZENG Liang[1] SHI Junyang[1] HU Mai[1] LI Ming[1]
1 School of Electrical and Electronic Engineering,Hubei University of Technology,Wuhan 430068,China

Abstract Aiming at the distributed permutation flowshop scheduling problem with variable processing speed,a dual-population algorithm is proposed to optimize the makespan and the total energy consumption of the machine.First, an initialization method that mixes four strategies is used to generate a high-quality initial population.Second, specific evolution methods are designed according to the characteristics of the two populations,and the dynamic guide factor is introduced to adjust the evolution mode of the populations.Meanwhile,an energy-saving strategy for speed regulation is proposed to further optimize energy consumption.Finally,a dynamic population strategy is proposed to balance the resources of the two populations.Simulation results verify the effectiveness of each strategy,and show that the proposed dual population algorithm outperforms current multi-objective evolutionary algorithms.

Key words permutation flowshop;dual-population algorithm;distributed;variable speed shop;multi-objective optimization