

文章编号: 2095-2163(2022)12-0008-08

中图分类号: TP301.6

文献标志码: A

改进麻雀搜索算法求解 DFJSP 问题

王 灿, 叶春明

(上海理工大学 管理学院, 上海 200093)

摘要: 针对分布式柔性作业车间调度最小化最大完工时间问题的特性, 提出一种改进麻雀搜索算法 (ISSA)。该算法中, 编码采用基于工序序列的编码方案, 利用随机生成和反向学习混合的策略初始化种群以增加种群的多样性。针对麻雀搜索算法 (SSA) 易陷入局部最优, 结合莱维飞行、学习更新策略、正态云模型优化 SSA, 同时通过交叉和变异算子提升算法的全局和局部搜索能力。进一步地, 通过扩展的 18 个标准算例分别对 2 家工厂、3 家工厂生产调度进行仿真实验, 将 ISSA 与现有算法进行对比, 验证了所提算法求解 DFJSP 的有效性。

关键词: 麻雀搜索算法; 分布式柔性作业车间调度; 莱维飞行; 学习更新策略; 正态云模型

Improved sparrow search algorithm for distributed flexible job-shop scheduling problem

WANG Can, YE Chunming

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

【Abstract】 Considering the characteristics of the distributed flexible job-shop scheduling problem (DFJSP) with minimizing makespan, an improved sparrow search algorithm (ISSA) is proposed. In this algorithm, the coding scheme is based on the operation sequence. In order to increase the diversity of the population, a strategy combining random generation and reverse learning is used to initialize the population. In view of the fact that the sparrow search algorithm (SSA) is easy to fall into local optimal, the SSA is optimized by combining Levy flight, learning update strategy and normal cloud model. At the same time, the crossover and mutation operators are used to improve the global and local search ability of the algorithm. Based on the aboved, the extended 18 standard instances are used to conduct simulation experiments on the production scheduling of two and three factories respectively, and the effectiveness of ISSA in solving DFJSP is verified by comparing the proposed algorithm with the existing algorithms.

【Key words】 sparrow search algorithm; distributed flexible job-shop scheduling; Levy flight; learning update strategy; normal cloud model

0 引言

随着经济全球化和制造企业的规模化, 分布式制造成为一种常见的生产模式。分布式柔性作业车间调度问题 (Distributed flexible job shop scheduling problem, DFJSP) 是一个具有重要研究意义的基础性问题, 传统的单一工厂集中性生产逐渐被淘汰, 多个柔性作业车间相互协同生产调度成为研究的重点。

目前, 有关 DFJSP 研究较少, 何怡^[1] 针对 DFJSP 提出一种改进蚁群算法, 通过两阶段参数自适应动态调整, 并利用旅行商问题验证了算法的有效性, 将其应用于 DFJSP 的求解。吴锐等人^[2] 针对

DFJSP 特点, 设计了三维向量编码, 并提出一种改进人工蜂群算法求解。吴秀丽等人^[3] 提出一种改进差分进化算法求解 DFJSP 的双目标优化模型, 对总成本和提前或延期惩罚同时进行了优化。王彪^[4] 提出引入自适应移动因子的改进混合蛙跳算法对分布式柔性生产调度进行优化。马庆吉^[5] 提出一种新的编码方式和基于启发式规则的解码方式, 设计了相应的优化策略, 并改进灰狼算法求解 DFJSP。孟磊磊等人^[6] 提出一种引入变邻域搜索算法、针对关键工厂全解空间禁忌搜索的混合蛙跳算法。陈文洲等人^[7] 设计一种基于帕雷托优化方法的人工蜂群算法对低碳 DFJSP 双目标优化模型进行求解。Giovanni 等人^[8] 提出一种改进遗传算法求解

基金项目: 国家自然科学基金 (71840003); 上海市科委“科技创新行动计划”软科学基金资助项目 (20692104300); 上海理工大学科技发展基金资助项目 (2018KJFZ043)。

作者简介: 王 灿 (1997-), 女, 硕士研究生, 主要研究方向: 智能算法、生产调度; 叶春明 (1964-), 男, 博士, 教授, 博士生导师, 主要研究方向: 工业工程、智能算法、医疗调度、生产调度等。

通讯作者: 叶春明 Email: yechm6464@163.com

收稿日期: 2022-06-19

DFJSP。Lu 等人^[9]认为 DFJSP 本质是三维解空间探索问题,涉及制造单元分配、机器选择和工序排序三个调度决策,并通过改进遗传算法求解该问题。

麻雀搜索算法(sparrow search algorithm, SSA)是 2020 年由薛建凯^[10]根据麻雀种群的觅食行为和反捕食行为提出的一种新型群体智能优化算法,与其他算法相比,其搜索精度、收敛速度和稳定性等方面极具竞争力,已被应用于多个领域,目前已然应用于解决车间调度问题。刘丽娜等人^[11]提出一种量子计算、正余弦搜索和警戒者数量递减策略改进的麻雀搜索算法,并引入多邻域搜索和高斯扰动策略来求解作业车间调度问题。因此,本文提出一种改进麻雀搜索算法(improved sparrow search algorithm, ISSA),并验证其在求解分布式柔性作业车间调度问题中的有效性。

1 分布式柔性作业车间调度问题

1.1 符号定义

为方便理解,对本文出现的相关符号进行定义,具体含义见表 1。

表 1 符号定义

Tab. 1 Symbols definition

符号	含义
i	工件索引号, $i \in \{1, 2, \dots, n\}$
j	工序索引号, $j \in \{1, 2, \dots, q_i\}$
z	机器号, $z \in \{1, 2, \dots, m_r\}$
r	工厂号, $r \in \{1, 2, \dots, f\}$
D	所有工件工序数之和
m_r	工厂 r 中机器数目
q_i	工件 i 的工序数
O_{ij}	工件 i 的第 j 道工序
S_{ij}	工序 O_{ij} 的起始加工时间
C_{ij}	工序 O_{ij} 的结束加工时间
T_{ijrz}	工序 O_{ij} 在工厂 r 中第 z 台机器上的加工时间
V_{ir}	工件 i 是否在工厂 r 中加工, 为 0-1 变量
X_{ijrz}	工序 O_{ij} 是否在工厂 r 中第 z 台机器上加工, 为 0-1 变量
Y_{ijrt}	t 时刻工序 O_{ij} 是否在工厂 r 中第 z 台机器上加工, 为 0-1 变量
C_i	工件 i 的完工时间
C_{\max}	全局最大完工时间

1.2 问题描述

DFJSP 包含工序排序、工厂选择、机器选择三个子问题^[12],实质是一个典型的组合优化问题^[1],可将其描述为:将 n 个工件在 f 家工厂加工,每家工厂有 m_r 台机器,一个工件的所有工序只能在一家工厂加工,每道工序的可选择加工机器以及加工时间已知。DFJSP 满足以下假设:

(1) 所有机器从 0 时刻开始可用。

(2) 各工件加工的优先级相同,同一工件各道工序有先后约束。

(3) 每道工序在同一时刻只能被一台机器加工。

(4) 所有工件在 0 时刻可以被加工且不考虑工件的运输时间。

(5) 机器加工过程中没有故障。

调度的目的是安排每家工厂的每台机器上工件的加工顺序,为每一个工件选择合适的工厂,为每一道工序选择适当的机器,使完工时间达到最小。

1.3 模型建立

本文以最小化全局最大完工时间 C_{\max} 为目标进行优化。具体数学模型如下^[13]:

$$f = \min(C_{\max}) = \min(\max(C_i)) \quad (1)$$

$$\sum_{r=1}^f V_{ir} = 1, \forall i = 1, 2, \dots, n \quad (2)$$

$$\sum_{j=1}^{q_i} \sum_{r=1}^f \sum_{z=1}^{m_r} V_{ir} X_{ijrz} = q_i, \forall i = 1, 2, \dots, n \quad (3)$$

$$C_{ij} \leq S_{i(j+1)}, \forall i = 1, 2, \dots, n \quad (4)$$

$$\sum_{r=1}^f \sum_{z=1}^{m_r} X_{ijrz} = 1, \forall i = 1, 2, \dots, n; \forall j = 1, 2, \dots, q_i \quad (5)$$

$$\sum_{i=1}^n \sum_{j=1}^{q_i} Y_{ijrt} \leq 1, \forall r = 1, 2, \dots, f, \forall z = 1, 2, \dots, m_r \quad (6)$$

$$C_{ij} - S_{ij} = \sum_{r=1}^f \sum_{z=1}^{m_r} X_{ijrz} T_{ijrz} \quad (7)$$

其中,式(1)为目标函数,表示全局最大完工时间的最小化;式(2)表示每个工件只能被分到一家工厂;式(3)表示每个工件的所有工序须在一家工厂完成加工;式(4)表示各工件的工序加工有先后约束;式(5)表示各工序只能选择在一家工厂的一台机器上加工;式(6)表示一台机器在任意时间节点只能对一道工序进行加工;式(7)表示每道工序直到加工完成不会被中断。

2 改进麻雀搜索算法求解 DFJSP

2.1 SSA 概述

麻雀搜索算法模拟了麻雀群体的觅食和反捕食行为,在搜索精度、收敛速度和稳定性等方面有较强的竞争力。麻雀个体分为发现者、加入者和侦察者,发现者负责搜寻并为麻雀种群提供食物丰富的区域和方向,加入者跟随发现者以获得食物。此外,种群中存在一定比例的个体负责侦查预警,如发现危险则发出报警信号并在种群中移动,当预警值大于安全值时,发现者需带领加入者移动到其他安全的区域寻找食物。研究对此拟做阐释分述如下。

(1)发现者。每次迭代过程中,发现者位置更新如下:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left\{\frac{\alpha}{e}\right\} \frac{i}{\alpha \cdot iter_{max}} \cdot \frac{\delta}{\delta} & R_2 < ST \\ X_{i,j}^t + Q \cdot L & R_2 \geq ST \end{cases} \quad (8)$$

其中, t 表示当前迭代次数, $j = 1, 2, \dots, d$; $iter_{max}$ 表示最大迭代次数; $X_{i,j}$ 表示第 i 个麻雀在第 j 维中的位置信息; $\alpha \in (0, 1]$ 表示一个随机数; $R_2 (R_2 \in [0, 1])$ 表示预警值; $ST (ST \in [0.5, 1])$ 表示安全值; Q 是服从正态分布的随机数; L 表示一个 $1 \times d$ 的矩阵且元素全部为 1。

(2)加入者。加入者位置更新如下:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left\{\frac{\alpha X_{worst}^t - X_{i,j}^t}{i^2}\right\} \cdot \frac{\delta}{\delta} & i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L & i \leq \frac{n}{2} \end{cases} \quad (9)$$

其中, X_p 表示当前发现者占据的最优位置; X_{worst} 表示当前全局最差的位置; A 表示一个 $1 \times d$ 的矩阵, 元素随机取值为 1 或 -1, 且 $A^+ = A^T (AA^T)^{-1}$ 。

(3)侦察者。侦察者位置更新如下:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t| & f_i \neq f_g \\ X_{i,j}^t + K \cdot \frac{\alpha |X_{i,j}^t - X_{worst}^t|}{e(f_i - f_w) + \varepsilon} & f_i = f_g \end{cases} \quad (10)$$

其中, X_{best} 是当前全局最优位置; β 是服从标准正态分布的随机数; $K \in [-1, 1]$, f_i 表示当前麻雀个体适应度值; f_g 表示当前种群中最佳适应度值; f_w 表示当前种群中最差适应度值; ε 为极小的常数。

2.2 编码与解码

本文选用基于工序序列的编码, 编码仅包含一

条工序序列, 序列长度等于所有工件的工序数之和, 用来确定工序排序。

麻雀搜索算法用于处理连续型问题, 无法直接应用于车间调度等离散型问题, 因此需要选择合适的编码方式。 n 个工件共 D 道工序, 个体位置向量为 $X = [x_1, x_2, \dots, x_d, \dots, x_D]$, 各元素可在 $[-\delta, \delta]$ 中任意取值, 并按一定顺序存储。假设有 3 个工件、共 8 道工序, 各元素在 $[-2, 2]$ 中任意取值, 个体位置向量如图 1 所示。

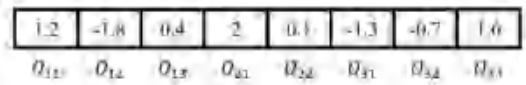


图 1 个体位置向量

Fig. 1 Individual position vector

采用文献[14]中的转换方法实现连续个体位置向量向离散车间调度之间的转化。转换结果如图 2 所示, 可得工序序列为 $O_{31} \rightarrow O_{11} \rightarrow O_{21} \rightarrow O_{32} \rightarrow O_{22} \rightarrow O_{12} \rightarrow O_{13} \rightarrow O_{33}$ 。

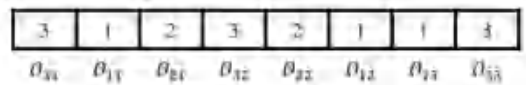


图 2 调度方案

Fig. 2 Scheduling scheme

与一般车间调度问题相比, DFJSP 需考虑工厂选择, 本文工厂选择采用文献[15]的方法, 包含 3 个步骤:

(1)选择每个工件的第一道工序, 并维持这些工序的序列, 推导得出一个工件序列, 如从 $O_{31} \rightarrow O_{11} \rightarrow O_{21} \rightarrow O_{32} \rightarrow O_{22} \rightarrow O_{12} \rightarrow O_{13} \rightarrow O_{33}$ 之中选出 $O_{31} \rightarrow O_{11} \rightarrow O_{21}$, 进而推出工件序列 $J_3 \rightarrow J_1 \rightarrow J_2$ 。

(2)计算各工件在各工厂的平均加工时间, 将工件-工厂-机器表缩减得到工件-工厂表。

(3)按照工件序列, 根据当前最大完工时间最小化规则将工件分配给特定工厂。

2.3 种群初始化

SSA 算法随机初始化产生种群个体的方式, 会导致生成种群分布不均匀, 种群多样性减少, 影响算法寻优效率。为保证求解速度和质量, 本文初始化种群的生成策略为随机生成与反向学习相混合, 50% 个体采用随机生成方式, 50% 个体采用反向学习生成方式。利用反向学习策略^[16]生成种群的主要思想: 首先随机生成初始种群, 再根据初始种群生成其反向种群, 从中选择较优的种群作为下一代种群。操作步骤为:

(1)随机初始化 n 个麻雀个体的位置 $X = [x_1,$

$x_2, \dots, x_d, \dots, x_D$] 作为初始种群 P_1 , 其中各元素可在 $[-\delta, \delta]$ 中任意取值。

(2) 生成初始种群 P_1 中每个麻雀个体的反向个体, 构成反向种群 P_2 , 其中反向个体位置 $X' = [x'_1, x'_2, \dots, x'_d, \dots, x'_D]$, 其中 $x'_d = -x_d$ 。

(3) 将种群 P_1, P_2 合并, 根据适应度值对 $2n$ 个麻雀个体升序排序, 最终选取适应度值前 n 个麻雀个体为初始种群。

2.4 莱维飞行优化发现者位置

麻雀在觅食状态下的运动路线是莱维飞行轨迹, 特点是大范围的探索, 折线方式的急剧转向, 目标是获得更多食物机会。在 SSA 算法中, 由于发现者在搜索过程中的带头作用, 加入者位置更新易受发现者影响, 使得麻雀种群容易出现聚集, 陷入局部最优, 因此发现者的位置更新必须有一定的随机性。为避免 SSA 容易早熟而陷入局部最优, 提高全局寻优能力, 在发现者位置更新公式中引入莱维飞行^[17]。利用 Levy 飞行来增加发现者搜索方向的多元性, 以增强算法的全局搜索能力, 避免陷入局部最优。

Levy 飞行的随机步长 s 的计算公式为:

$$\begin{cases} s = \mu / |v|^{1/\beta} & \beta = 1.5 \\ \mu \sim N(0, \sigma_\mu^2) & v \sim N(0, \sigma_v^2) \\ \sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{\beta \Gamma[(1+\beta)/2] \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta} & \sigma_v = 1 \end{cases} \quad (11)$$

当 $R_2 < ST$ 时, 发现者个体位置的每一维都在缩小, 因此将发现者位置更新公式改进为:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + X_{i,j}^t \oplus Levy(s) & R_2 < ST \\ X_{i,j}^t + QL & R_2 \geq ST \end{cases} \quad (12)$$

其中, “ \oplus ”表示点乘, $Levy(\lambda)$ 表示步长服从 Levy 分布的随机搜索向量 $s(\mu, v)$ 。

2.5 学习更新策略优化加入者位置

加入者在有效控制全局探索和局部开发的平衡中发挥重要作用, 但发现最优解后, 加入者迅速向最优解靠拢, 容易陷入局部最优, 因此, 对加入者的位置引入学习更新策略, 使加入者不仅向最优麻雀个体学习, 也进行自我学习和随机个体学习。

(1) 自我学习。在加入者位置中引入自适应惯性权重因子 ω 来提升加入者对自身的学习, 权重因子 ω 计算公式如下:

$$\omega = \exp\left(1 - \frac{t_{\max} + t}{t_{\max} - t}\right) \quad (13)$$

迭代前期, 权重因子 ω 值较大, 可提高全局搜

索能力, 迭代后期 ω 值自适应减小, 有利于进行局部搜索, 并提高收敛速度。

(2) 最优个体学习。借鉴萤火虫算法^[18]中的吸引度规则, 萤火虫种群中发光弱的会被发光强的吸引并向其移动, 且吸引度仅仅与发光强度、距离相关, 距离越大, 吸引度越小。吸引度 β 计算方式如下:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (14)$$

$$r = X_p^{t+1} - X_{i,j}^t = \sqrt{\sum_{j=1}^D (X_p^{t+1} - X_{i,j}^t)^2} \quad (15)$$

其中, β_0 表示最大吸引度, 即 $r=0$ 时的吸引度; γ 表示光吸收系数, 在 $[0.1, 10]$ 中任意取值; r 表示 2 只萤火虫之间的距离; $\beta_0 = 1$ 。

(3) 随机个体学习。向随机个体学习有助于提升种群多样性, 但随机个体影响因子过大不利于算法收敛, 因此, 随机个体学习因子 S 定义为:

$$S = \exp(-|f_i - f_k|) \cdot randn() \quad (16)$$

其中, f_k 为随机个体 k 的适应度值。

加入者位置更新公式改进为:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{\alpha X_{worst}^t - X_{i,j}^t}{i^2}\right) & i > \frac{n}{2} \\ \omega \cdot X_{i,j}^t + \beta \cdot (X_p^{t+1} - X_{i,j}^t) + S \cdot |X_{i,j}^t - X_{k,j}^t| & i \leq \frac{n}{2} \end{cases} \quad (17)$$

其中, $X_{k,j}$ 表示随机个体 k 的位置。

2.6 正态云模型优化侦查者位置

为进一步提高 SSA 跳出局部最优的能力, 且提升算法的收敛速度和求解精度, 在侦查者位置更新公式中引入云模型^[19]。云模型是处理定性概念和定量描述间的不确定转换模型, 反映了随机性和模糊性^[20]。云的数字特征为期望 (E_x)、熵 (E_n) 和超熵。期望是能够代表定性概念量化的最典型样本, 熵反映了云滴的离散程度和取值范围, 超熵反映了云滴的凝聚程度。正态云滴生成过程可定义如下:

$$X[x_1, x_2, \dots, x_{N_d}] = Gnc(E_x, E_n, H_e, N_d) \quad (18)$$

其中, N_d 表示期望生成的云滴数目。

将当前全局最优位置 X_{best}^t 作为正态云模型的期望 E_x , 侦查者位置更新公式改进为:

$$P = Gnc(X_{best}^t, E_n, H_e, N_d) \quad (19)$$

其中, P 表示麻雀种群中侦查者的位置; E_n 表示侦查者距离麻雀最优个体的范围; H_e 表示侦查者位置的分散程度。在迭代前期, E_n 值较大, 能够提高全局搜索能力, 在迭代后期自适应减小, 有利于进

行局部搜索,提高收敛速度。因此, E_n, H_e 计算方式如下:

$$E_n = DISP_{max} \cdot \frac{\varphi \cdot iter_{max} - t}{\varphi \cdot iter_{max}} \quad (20)$$

$$H_e = E_n \times 10^{-\delta} \quad (21)$$

其中, $DISP_{max}$ 表示侦查者距离全局最优位置的最大距离, φ, δ 为正整数。

2.7 交叉和变异算子

交叉和变异两个操作相互配合、相互竞争,能够均衡算法的全局搜索和局部搜索能力。考虑到算法的运行时间,本文对每次迭代发现者中任2个麻雀个体进行交叉操作,因个体位置为连续型向量,故本文设计了一种交叉策略。对每次迭代发现者中任一麻雀个体进行变异操作,变异算子采用 SIM 算子。

交叉策略如图3所示。首先,在父代中随机选择几个位置可不连续的基因,2条染色体选择位置相同;然后,将父代1中所选基因复制到相同位置的子代1中,并将父代2中所选基因复制到相同位置的子代2中;接着,将父代1中的剩余基因复制到子代2的同一位置,并将父代2中的剩余基因复制到子代1的同一位置。

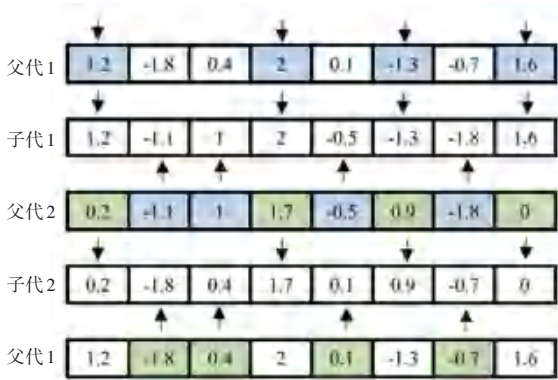


图3 交叉示意图

Fig. 3 Schematic diagram of crossing

变异算子采用 SIM 算子,即反转突变,如图4所示。在麻雀个体中选择一个随机的基因序列,将该序列中的基因顺序颠倒。



图4 变异示意图

Fig. 4 Schematic diagram of mutation

2.8 改进麻雀搜索算法步骤

至此,本文改进麻雀搜索算法流程如图5所示。

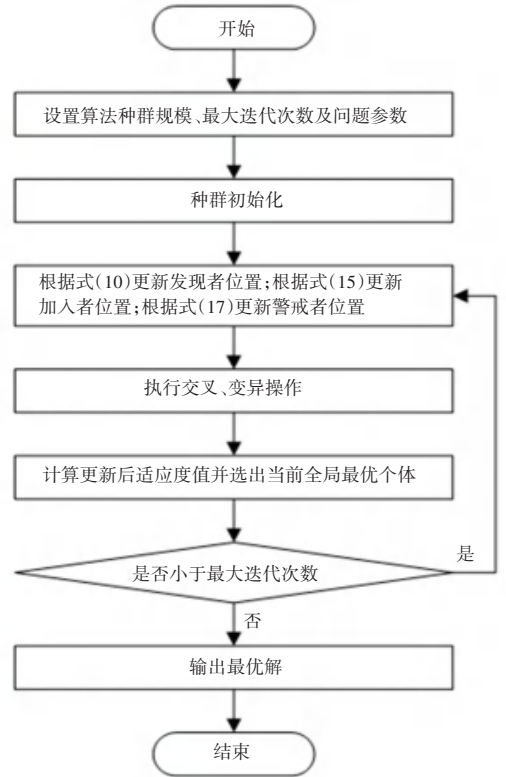


图5 ISSA 算法流程

Fig. 5 Algorithm process of ISSA

3 仿真实验与分析

3.1 算例生成

本文使用 Fisher 等人^[21]设计的3个典型问题 mt06、mt10 和 mt20 以及 15 个 Lawrence^[22]基准测试例子 la01~la15,并假设所有工厂完全相同,分别考虑2家、3家工厂的情况。

3.2 实验结果及分析

为验证 ISSA 算法求解分布式柔性作业车间调度的有效性,与 SSA 算法、PSO 算法对比进行分析。

本文算法运行环境为:Windows10 64 位操作系统, Intel (R) Core (TM) i5 - 1135G7 @ 2.40 GHz 2.42 GHz处理器,16.0 GB 内存, Matlab R2017a 开发软件。

为避免测试结果的随机性,更加准确地对 ISSA、SSA、PSO 算法就 DFJSP 进行比较,设置每个算法最大迭代次数为 500 次,种群规模为 100,针对每个算例,每种算法独立运行 20 次,取运行结果的最小值和平均值,2 家工厂、3 家工厂实验结果分别见表 2、表 3。

表 2 2 家工厂测试结果

Tab. 2 2 FMU test results

算例名称	结果	mt06	mt10	mt20	la01	la02	la03	la04	la05	la06	la07	la08	la09	la10	la11	la12	la13	la14	la15
		6×6	10×10	20×5	10×5	10×5	10×5	10×5	10×5	15×5	15×5	15×5	15×5	15×5	20×5	20×5	20×5	20×5	20×5
ISSA	最小值	47	723.00	748.00	432.00	396.00	369.00	397.00	380	564.00	526.00	552.00	593.00	573.00	674.00	619.00	699.00	724.00	731.00
	平均值	47	744.25	774.35	460.85	429.60	394.00	412.35	380	591.30	561.05	568.75	623.85	602.10	755.85	661.50	729.25	758.35	777.10
SSA	最小值	47	738.00	750.00	436.00	407.00	371.00	398.00	380	571.00	546.00	562.00	604.00	582.00	711.00	621.00	699.00	725.00	743.00
	平均值	47	759.10	774.95	462.35	431.50	396.25	418.60	380	594.40	563.10	577.60	628.55	604.20	766.65	665.45	730.95	760.50	778.70
PSO	最小值	47	725.00	736.00	441.00	422.00	368.00	401.00	380	567.00	522.00	542.00	606.00	582.00	736.00	634.00	701.00	730.00	757.00
	平均值	47	752.35	772.50	464.40	432.45	396.05	419.70	380	590.05	560.70	572.50	628.60	605.25	762.75	666.50	732.75	761.25	782.70

表 3 3 家工厂测试结果

Tab. 3 3 FMU test results

算例名称	结果	mt06	mt10	mt20	la01	la02	la03	la04	la05	la06	la07	la08	la09	la10	la11	la12	la13	la14	la15
		6×6	10×10	20×5	10×5	10×5	10×5	10×5	10×5	15×5	15×5	15×5	15×5	15×5	20×5	20×5	20×5	20×5	20×5
ISSA	最小值	47	655.00	551.00	413	394	349	369	380	451.00	421.00	428.00	468.00	460.00	556.00	501.0	541.00	546.0	564.0
	平均值	47	659.15	591.20	413	394	349	369	380	470.55	448.10	455.80	497.45	481.95	593.95	520.2	570.85	589.0	604.8
SSA	最小值	47	655.00	563.00	413	394	349	369	380	456.00	428.00	447.00	476.00	461.00	570.00	503.0	545.00	549.0	578.0
	平均值	47	659.20	592.95	413	394	349	369	380	474.60	450.05	460.40	503.75	482.80	591.90	526.4	571.10	590.3	612.0
PSO	最小值	47	655.00	571.00	413	394	349	369	380	462.00	429.00	436.00	489.00	457.00	565.00	513.0	556.00	564.0	585.0
	平均值	47	658.75	598.80	413	394	349	369	380	478.30	448.70	455.85	503.95	487.75	588.55	526.5	576.30	589.2	614.1

由表 2、表 3 结果可知,在最小值方面,ISSA 优于 SSA,大部分优于 PSO;在平均值方面,ISSA 绝大部分优于 SSA 和 PSO。综上,ISSA 在多工厂大部分算例上均表现优异,整体寻优性能优于 SSA 和 PSO。

为更好地对比 3 种算法的收敛效果,选取具有代表性的实例 mt10、la05、la10、la15,分别绘制了 2 家工厂、3 家工厂的研究中各算法的收敛曲线对比图,如图 6、图 7 所示。

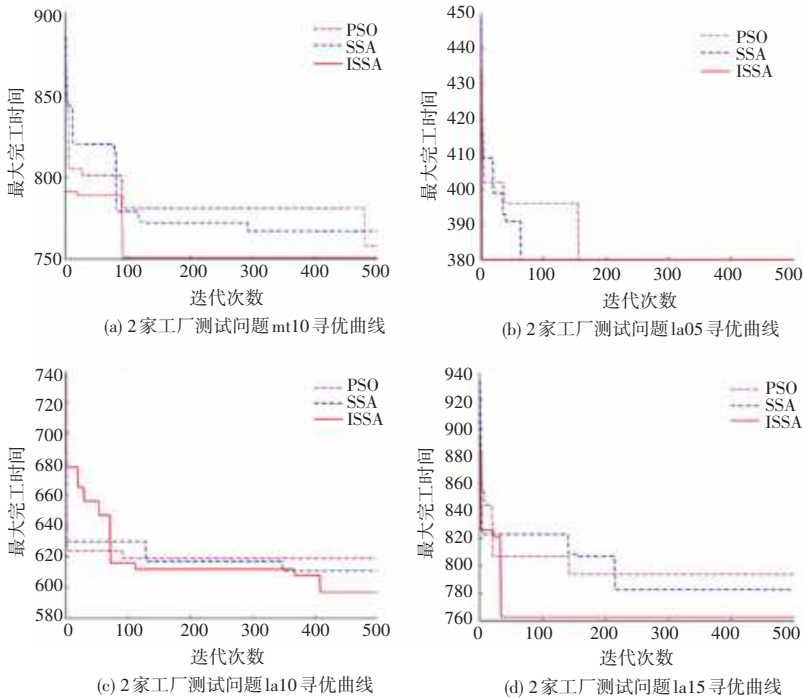


图 6 2 家工厂不同算法寻优曲线对比

Fig. 6 Comparison of optimization curves of different algorithms for two factories

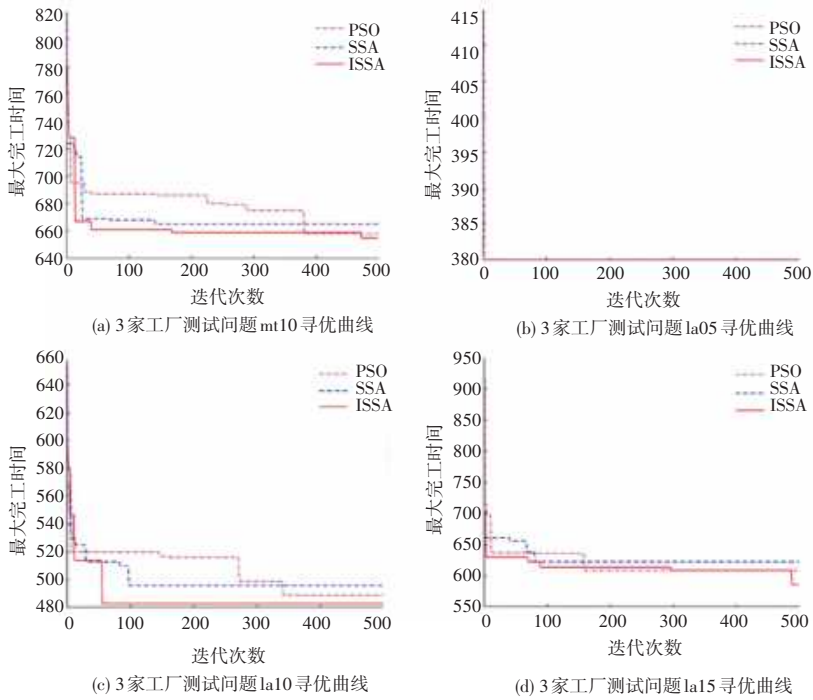
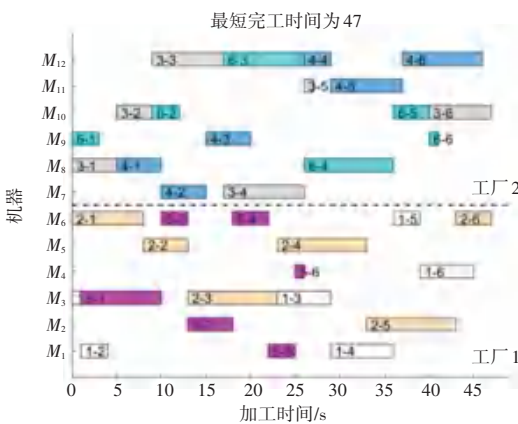


图7 3家工厂不同算法寻优曲线对比

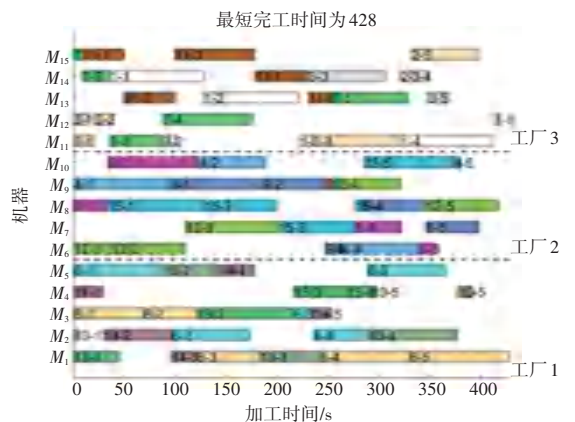
Fig. 7 Comparison of optimization curves of different algorithms for three factories

由图6、图7中ISSA与SSA、PSO算法寻优曲线对比可以看出,在求解2家工厂、3家工厂mt10、la05、la10、la15测试问题时,ISSA寻优曲线基本位于SSA、PSO寻优曲线左下方,且除2家工厂la10、3家工厂mt10、la15测试问题外,其余测试问题寻优曲线ISSA均在100次迭代内居于领先地位,说明ISSA在求解DFJSP时的收敛精度和收敛速度优于SSA和PSO。此外,2家工厂la10、3家工厂mt10、

la15测试问题在迭代进行100次后虽没能持续保持领先,但ISSA能够在迭代后期跳出局部最优,找到更优解。综上所述,利用莱维飞行、学习更新策略、正态云模型、交叉变异等对经典麻雀搜索算法改进后,ISSA能够有效平衡全局搜索和局部开发,在求解DFJSP过程中具有更高的收敛精度和更快的收敛速度。实验利用ISSA对DFJSP求解得到的最好调度结果的部分加工甘特图如图8所示。



(a) 2家工厂标准测试问题 mt06 加工甘特图



(b) 3家工厂标准测试问题 la08 加工甘特图

图8 部分算例甘特图

Fig. 8 Partial examples Gantt chart

4 结束语

本文研究了具有较高复杂度和求解难度的 DFJSP, 针对其特性, 建立了以最小化最大完工时间为优化目标的调度模型。为了有效求解该模型, 结合反向学习策略、莱维飞行、学习更新策略、正态云模型及交叉变异算子, 提出了一种改进麻雀搜索算法 (ISSA), 有效避免了 SSA 求解时易陷入局部最优的缺点。通过仿真实验与其他算法进行对比, 分析验证了 ISSA 的性能, 表明所提算法能有效求解 DFJSP, 且具有一定的优越性。目前, DFJSP 的相关研究还有所欠缺, 在未来工作中, 将进一步改进麻雀搜索算法或利用其他算法及其改进算法来求解该问题, 并探索研究更加符合生产实际的多目标 DFJSP。

参考文献

- [1] 何怡. 蚁群算法及其在分布式柔性作业车间调度中的应用[D]. 合肥: 合肥工业大学, 2016.
- [2] 吴锐, 郭顺生, 李益兵, 等. 改进人工蜂群算法求解分布式柔性作业车间调度问题[J]. 控制与决策, 2019, 34(12): 2527-2536.
- [3] 吴秀丽, 刘夏晶. 差分进化算法求解分布式柔性作业车间调度问题[J]. 计算机集成制造系统, 2019, 25(10): 2539-2558.
- [4] 王彪. 基于改进蛙跳算法的分布式柔性作业车间调度研究[D]. 武汉: 武汉理工大学, 2019.
- [5] 马庆吉. 基于改进灰狼算法的柔性作业车间调度方法研究[D]. 武汉: 华中科技大学, 2019.
- [6] 孟磊磊, 张彪, 任亚平, 等. 求解分布式柔性作业车间调度的混合蛙跳算法[J]. 机械工程学报, 2021, 57(17): 263-272.
- [7] 陈文洲, 李登峰, 郑小雪. 考虑低碳的分布式柔性作业车间调度优化[J]. 组合机床与自动化加工技术, 2021(11): 164-168.
- [8] GIOVANNI L D, PEZZELLA F. An improved genetic algorithm for the distributed and flexible job-shop scheduling problem [J]. European Journal of Operational Research, 2010, 200(2): 395-408.
- [9] LU P H, WU M C, TAN Hao, et al. A genetic algorithm

- embedded with a concise chromosome representation for distributed and flexible job-shop scheduling problems [J]. Journal of Intelligent Manufacturing, 2015, 29: 19-34.
- [10] 薛建凯. 一种新型的群智能优化技术的研究与应用[D]. 上海: 东华大学, 2020.
- [11] 刘丽娜, 南新元, 石跃飞. 改进麻雀搜索算法求解作业车间调度问题[J]. 计算机应用研究, 2021, 38(12): 3634-3639.
- [12] 李瑞, 王凌, 龚文引. 知识驱动的模因算法求解分布式绿色柔性调度[J]. 华中科技大学学报(自然科学版), 2022, 50(06): 55-60.
- [13] CHAN F T S, CHUNG S H, CHAN P L Y. Application of genetic algorithms with dominant genes in a distributed scheduling problem in flexible manufacturing systems [J]. International Journal of Production Research, 2006, 44(3): 523-543.
- [14] YUAN Yuan, XU Hua, YANG Jiadong. A hybrid harmony search algorithm for the flexible job shop scheduling problem [J]. Applied Soft Computing, 2013, 13(7): 3259-3272.
- [15] WU M C, LIN C S, LIN C H, et al. Effects of different chromosome representations in developing genetic algorithms to solve DFJS scheduling problems [J]. Computers & Operations Research, 2017, 80(4): 101-112.
- [16] TIZHOOSH H R. Opposition-based learning: A new scheme for machine intelligence [C]// International Conference on International Conference on Computational Intelligence for Modelling, Control & Automation. Vienna, Austria; IEEE, 2005: 695-701.
- [17] 王庆喜, 郭晓波. 基于莱维飞行的粒子群优化算法[J]. 计算机应用研究, 2016, 33(09): 2588-2591.
- [18] YANG X S. Nature-inspired metaheuristic algorithms [M]. Frome: Luniver Press, 2008: 83-96.
- [19] 李德毅, 孟海军, 史雪梅. 隶属云和隶属云发生器[J]. 计算机研究与发展, 1995, 32(06): 15-20.
- [20] 李德毅, 刘常显. 论正态云模型的普适性[J]. 中国工程科学, 2004, 6(08): 28-34.
- [21] FISHER H, THOMPSON G L. Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules [M]. Upper Saddle River, N. J., USA: Prentice Hall, 1963.
- [22] LAWRENCE S. Supplement to resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques [J]. Energy Procedia, 1984, 4(7): 4411-4417.

(上接第 7 页)

- [26] WANG Zhou, SIMONCELLI E P, BOVIK A C. Multiscale structural similarity for image quality assessment [C]//The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers. Pacific Grove, CA, USA; IEEE, 2003, 2: 1398-1402.

- [27] CHEN Chaofeng, LIU Wei, TAN Xiao, et al. Semi-supervised learning for face sketch synthesis in the wild [C]//Asian Conference on Computer Vision. Cham; Springer, 2018: 216-231.