[文章编号] 1003－4684(2025)04-0011-05

基于改进 GWO 算法的柔性作业车间调度问题求解

龚立雄，肖杪铃，王圆圆，梁嘉乐，范岩森
（湖北工业大学机械工程学院，湖北 武汉 430068）

[摘 要] 以最小化最大完工时间为目标，提出一种改进灰狼优化(IGWO)算法，用于求解柔性作业车间调度问题。首先，采用机器选择和工序排序分开编码；其次，运用 GLR 的初始化方法，提升解的质量并保证狼群多样化；接着，融合交叉与变异算子，有效抑制算法早熟收敛现象；最后，引入改进变邻域搜索策略，强化算法的局部搜索性能。通过对 MK 标准数据集的求解，以及与其他算法进行对比分析，结果表明 IGWO 算法在求解柔性作业车间调度问题具备显著优势。

[关键词] 柔性作业车间调度；最大完工时间；灰狼优化算法；改进变邻域搜索
[中图分类号] TH165 [文献标识码] A

作为制造业领域公认的 NP-hart 问题，传统的柔性作业车间调度问题(flexible job shop scheduling problem，FJSP)解决方法已难以适应生产环境的多样性和复杂性，因此，亟需引入创新性优化手段，以提高生产效率和资源利用率，从而更好地适应制造业的快速变革。灰狼优化算法作为一种启发式算法，通过模拟灰狼群体在自然界中的协同狩猎行为，展现出良好的全局搜索性能与收敛速度。该算法为制造业智能化发展提供了创新高效的优化途径，有力推动了工业 4.0 时代制造业的可持续发展。

自灰狼优化(grey wolf optimization，GWO)算法提出以来，众多国内外学者将其运用于生产调度问题研究。姜天华[1]在传统 GWO 算法基础上嵌入变邻域搜索策略，以增强算法的局部搜索能力。该研究以最小化最大完工时间为目标，通过实验数据验证了 HGWO 算法在求解 FJSP 问题上的有效性。姚远远等[2]通过采用进化种群动态调整、反向学习初始化和最优个体变异等策略改进标准 GWO 算法。实验结果表明，改进后的 GWO 算法具备更强的鲁棒性。王玉芳等[3]将基于种群规模的自适应社会等级制度分布策略用于 AGWO 算法，显著提升了算法的求解速度与稳定性。李稚等[4]设计了 IMOGWO 算法用于求解多目标动态 FJSP，通过仿真实验并与其他算法对比，证实该算法具有更优的收敛性和分布性。ZHU 等[5]提出混合细胞灰狼优化(SCEGWO)算法，用于解决由多级中间部件组成的复杂装配产品生成调度问题。大量实验结果表明，该算法在处理此类问题时性能优于其他算法。GU 等[6]提出离散遗传-灰狼优化(DGGWO)算法，将离散型遗传算法的交叉操作融入原始 GWO 算法，在求解多目标低碳作业车间调度问题上取得良好效果。JIANG 等[7]将具有双重搜索模式的新框架运用于 GWO 算法，提出双搜索模式的灰狼优化(DMGWO)算法，且仿真实验验证了该算法的有效性。WANG 等[8]根据焊接车间特性，构建了针对焊接车间动态事件的焊接车间逆调度问题(WSISP)模型，并提出一种改进灰狼优化算法进行求解。实例分析表明，该算法能够有效求解该模型。基于上述研究，本文提出一种改进的灰狼优化算法用于求解 FJSP。该算法采用 MSOS 编码方式，将机器选择与工序排序分开进行编码与解码；通过 GLR 初始化方法，在提升解质量的同时确保狼群多样性；利用交叉和变异操作避免算法早熟收敛；引入改进的变邻域搜索机制，进一步强化算法的局部搜索能力。最后，通过对 MK 数据集进行仿真求解，验证了 IGWO 算法在求解柔性作业车间调度问题上的优越性。

1 问题描述

FJSP 问题可描述为：n 个工件 $(J_1, J_2, \cdots, J_n)$ 在 m 台机器 $(M_1, M_2, \cdots, M_m)$ 上加工；每个工件包含一道或多道工序；每道工序可在多台机器上加

[收稿日期] 2023－11－08
[基金项目] 湖北省科技计划重点研发专项(2023BAB042)
[第一作者] 龚立雄(1978－)，男，湖北仙桃人，工学博士，湖北工业大学副教授，研究方向为智能制造与决策优化。
[通信作者] 肖杪铃(1999－)，男，湖北武汉人，湖北工业大学硕士研究生，研究方向为车间调度与优化。

工;不同机器对同一工序,以及同一机器对不同工序的加工时间均存在差异。因此,该问题包含机器选择(MS)和工序排序(OS)两个子问题。约束条件如下:

1)所有工件优先级相同;

2)每台机器一次只能处理一个工件;

3)任何时间一个工件只能在一台机器上加工;

4)一道工序只能在可选机器集中的一台机器上加工,且加工过程不中断;

5)所有工件都可以在 0 时刻开始加工,所有机器在 0 时刻都可用;

6)每个工件的加工顺序不可改变;

7)不考虑机器转换和工件安装时间。

本文以优化最小化最大完工时间($C_{\max}$)为目标,则目标函数为:

$$\min C_{\max} = \min\{\max_{1\leqslant i\leqslant n}(C_i)\} \tag{1}$$

2 改进混合灰狼优化算法

2.1 传统灰狼优化算法

近些年来,灰狼优化算法在车间调度、图像分类、参数优化等领域都得到了成功的应用。GWO 算法的优化过程通常包括社会等级分层、猎物跟踪、猎物包围和猎物狩取四个阶段。社会等级分层具有 α、β、δ 和 ω 狼等四层。在捕猎行为中,α 狼是最高等级的决策者,β 狼则充当候选人,辅助并服从 α 狼的决策;δ 狼是其余狼群的领导者,但服从 α 狼和 β 狼的指令,ω 狼是等级最低的阶层。α、β 和 δ 狼适应度最高,负责完成种群的猎物跟踪和狩猎过程。每次狩猎完成后,狼群中个体位置调整的依据与 3 只狼的距离可表示为:

$$\boldsymbol{D} = \boldsymbol{C} \cdot X_p(t) - X(t) \tag{2}$$

$$\boldsymbol{X}(t+1) = X_p(t) - \boldsymbol{A} \cdot \boldsymbol{D} \tag{3}$$

$$\boldsymbol{A} = 2\alpha \cdot r_1 - \alpha \tag{4}$$

$$\boldsymbol{C} = 2r_2 \tag{5}$$

$$\begin{cases} D_\alpha = C_1 \cdot X_\alpha - X \\ D_\beta = C_2 \cdot X_\beta - X \\ D_\delta = C_3 \cdot X_\delta - X \end{cases} \tag{6}$$

$$\begin{cases} X_1 = X_\alpha - A_1 \cdot D_\alpha \\ X_2 = X_\beta - A_2 \cdot D_\beta \\ X_3 = X_\delta - A_3 \cdot D_\delta \end{cases} \tag{7}$$

$$\boldsymbol{X}(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{8}$$

其中:$\boldsymbol{D}$ 为个体与猎物距离向量;t 为迭代次数;$\boldsymbol{A}$、$\boldsymbol{C}$ 为协同系数向量;$\boldsymbol{X}_p$ 为猎物位置向量;$\boldsymbol{X}(t)$ 为当前灰狼位置向量;迭代过程中 α 由 2 线性降到 0;$\boldsymbol{r}_1$、$\boldsymbol{r}_2$ 为[0,1]随机向量;$\boldsymbol{X}_\alpha$、$\boldsymbol{X}_\beta$、$\boldsymbol{X}_\delta$ 分别表示当前种群 α、β、δ 位置向量;$\boldsymbol{D}_\alpha$、$\boldsymbol{D}_\beta$、$\boldsymbol{D}_\delta$ 分别表示当前候选狼与最优 3 只狼之间的距离;$|A|>1$ 为狼群分散在各个区域搜集猎物,$|A|<1$ 为狼群集中在某个或某些区域搜索猎物。

2.2 编码与解码

由于 FJSP 包含机器选择(MS)和工序排序(OS)两个子问题,本文算法采用分段编码方式。如图 1 所示,编码分为左右两部分:左侧的机器选择(MS)部分中,工序 o_{11} 可从 4 台机器中选择加工设备,染色体上第一个“4” 代表可选机器集上的第四台机器,即使用机器 M_4 进行加工;工序 o_{21} 可在 2 台机器上加工,染色体上的“2”表示可选机器集中的第二台机器,对应在机器 M_3 上加工。右侧为工序排序(OS)部分,其核心在于确定工件每个工序的加工先后顺序。其中,第一个“1”指的是工件 J_1 的 o_{11} 工序,紧接着的 “2”指的是工件 J_2 的 o_{21} 工序,以此类推,最终确定的工序加工顺序为 $o_{11} \rightarrow o_{21} \rightarrow o_{22} \rightarrow o_{12} \rightarrow o_{23}$。

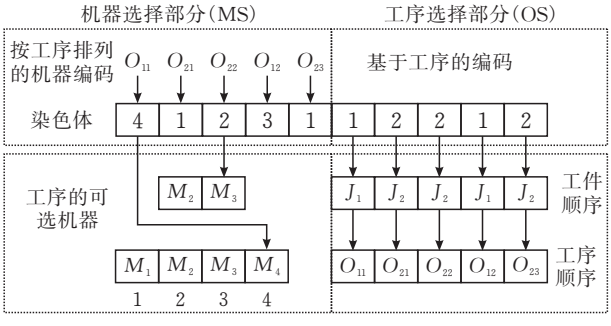


图 1 染色体编码

对机器选择(MS)部分和工序选择(OS)部分分别进行解码操作。机器选择部分的解码是按顺序读取编码数值,依据各工序可选机器集与编码数值的对应关系,将其转换成机器顺序矩阵 $\boldsymbol{J}_{\mathrm{M}}$ 和时间顺序矩阵 $\boldsymbol{T}$。例如,将图 1 所示染色体编码的机器选择部分进行转换后,$\boldsymbol{J}_{\mathrm{M}}$ 矩阵中每个元素代表对应工序所选机器编号,$\boldsymbol{T}$ 矩阵则记录相应工序在所选机器上的加工时间,具体转换结果如下:

$$\boldsymbol{J}_{\mathrm{M}} = \begin{bmatrix} 4 & 2 & \\ 3 & 3 & 2 \end{bmatrix} \tag{9}$$

$$\boldsymbol{T} = \begin{bmatrix} 1 & 7 & \\ 3 & 7 & 7 \end{bmatrix} \tag{10}$$

工序排序使用插入式贪婪解码[9],具体流程如下:1)按顺序读取工序染色体有序序列,确定第一道工序在机器选择部分已选定的机器上开始加工,并记录其加工开始与结束时间;2)读取第二道工序,根据对应机器当前是否空闲、已安排工序的结束时间等加工状态,计算该工序在对应机器上不同插入时刻的总加工时间,将其插入到使总加工时间最短的时刻,并更新该机器的加工时间安排和整体工序的

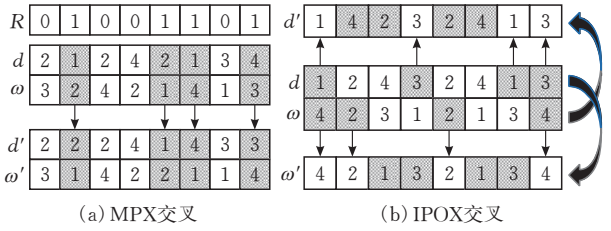
加工进度;3)重复上述插入操作,依次处理后续工序,直至完成所有工序的最优排序。这种解码方式通过局部最优策略,能够快速生成符合生产约束的可行解,且算法实现逻辑简单,有助于提升柔性作业车间调度问题(FJSP)的求解效率。

2.3　初始化

在传统灰狼算法中,初始种群采用随机选择的方式生成,每一次迭代本质上属于随机搜索过程。随着种群数量的增加,寻找到最优解的概率也会相应提升,但是为了避免算法复杂度过高,种群规模无法无限制扩大。相较于传统算法,本文提出的IGWO算法在机器选择部分使用 GLR 初始化方法,该方法融合全局搜索(GS)、局部搜索(LS)和随机选择(RS)三种策略。其中,GS 和 LS 策略能够有效平衡作业机器的工作负荷,避免部分机器过度繁忙而部分机器闲置的情况;RS 策略则确保初始种群在解空间内均匀分布,防止初始解过于集中。三者互相配合,不仅提高了机器利用率,还显著提升了初始解的质量。而在工序排序部分,IGWO 算法依旧沿用随机选择的方法,以此维持狼群的多样性,为算法后续迭代提供更丰富的搜索方向,助力算法跳出局部最优解,提升整体寻优性能。

2.4　交叉与变异

在机器选择部分采用 MPX 交叉策略,工序排序部分采用 IPOX 交叉策略。如图 2a 所示,MPX 交叉策略具体操作如下:从决策狼集合 ω 中任选一只狼作为引导狼,针对附属狼对应位置的元素执行 MPX 交叉操作。首先生成一个取值范围在[0,1]的随机向量 $\boldsymbol{R}$,随后将决策狼 d 和附属狼集合 ω 中,$\boldsymbol{R}$ 向量对应位置元素为 1 的元素进行相互交换,交换后形成的新染色体分别记为 d' 和 ω'。如图 2b 所示的 IPOX 交叉策略,首先从决策狼中任选一只作为引导狼。在工序排序部分,决策狼与附属狼均保留各自工件原工序的位置顺序,在此基础上,双方依次填充对方所保留的剩余工序部分,从而完成交叉操作。



(a) MPX交叉　　(b) IPOX交叉

图 2　交叉操作

变异操作能够挖掘种群个体的潜力,进一步提升算法的局部搜索能力。在机器选择部分,随机选择两个位置,并在机器集中随机选择加工时间小于当前加工时间的机器;在工序排序部分,随机选择两个位置,将这两个位置间的工序编码进行翻转。

2.5　改进变邻域搜索

灰狼优化算法(GWO)通过选择 α、β、δ 狼的最佳位置来更新种群中所有个体的位置,这一策略赋予了 GWO 算法较强的全局搜索能力。但由于该算法在局部搜索性能上存在短板,因此,研究者们通常引入变邻域搜索算法(VNS),以提升灰狼算法的局部搜索效率。在 GWO 运行过程中,狼群的位置信息主要依据 α 狼的个体信息进行更新,这种更新机制存在较高概率导致算法陷入局部最优解;并且,α 狼个体的表现会直接影响整个狼群的进化上限。为满足柔性作业车间调度问题(FJSP)的求解目标,同时设计出兼顾随机变异与定向搜索特性、结构简洁的邻域结构,本文提出一种改进的变邻域搜索策略,该策略包含以下三种变邻域结构:

N_1:针对不同工件的工序任意选择 OS 中的两个元素,然后交换这两个元素。

N_2:先后选择 OS 中的两个元素,然后将后者插入到前者的位置。

N_3:任意选择 MS 中的一个元素,若其工序相应的可选机器集多于一台,则使用可选机器集中不同机器进行加工,由此生成新的元素值。

改进变邻域搜索算法的步骤如下:

Step1:初始解 X 为 α 的个体,设置 $q \leftarrow 1$,$\lambda_{\max} \leftarrow 3$,$\lambda \leftarrow 1$,最终迭代次数 $q_{\max}$。

Step2:假设 $\lambda = 1$,则 $X' \leftarrow N_1(X)$;假设 $\lambda = 2$,则 $X' \leftarrow N_2(X)$;假设 $\lambda = 3$,则 $X' \leftarrow N_3(X)$.

Step3:把 X' 视为初始解,得到局部最优解 X''.

Step4:假设 X'' 优于 X,则 $X \leftarrow X''$,$\lambda \leftarrow 1$;否则 $\lambda \leftarrow \lambda + 1$.

Step5:假设满足 $\lambda > \lambda_{\max}$,则 $q \leftarrow q + 1$,跳到 Step6;否则跳到 Step3.

Step6:假设满足 $q > q_{\max}$,跳到 Step7;否则 $\lambda \leftarrow 1$,跳到 Step2.

Step7:算法终止.

改进变邻域搜索算法则采用阈值接受法进行局部搜索,步骤如下:

Step1:设置初始解 X',阈值 $\psi > 0$,$\rho \leftarrow 1$,$\vartheta \leftarrow 1$,最终迭代次数 $\rho_{\max}$.

Step2:假设 $\vartheta = 1$,则 $X'' \leftarrow N_1(X') \cup N_3(X')$;假设 $\vartheta = 0$,则 $X'' \leftarrow N_2(X') \cup N_3(X')$.

Step3:假设满足 $C_{\max}(X'') - C_{\max}(X') \leqslant \psi$,则 $X' \leftarrow X''$;否则令 $\vartheta \leftarrow |\vartheta - 1|$.

Step4:设置 $\rho \leftarrow \rho + 1$,假设满足 $\rho > \rho_{\max}$,则 $X'' \leftarrow X'$;否则跳到 Step2.

Step5:算法终止.

2.6 算法步骤

算法流程如图 3 所示。

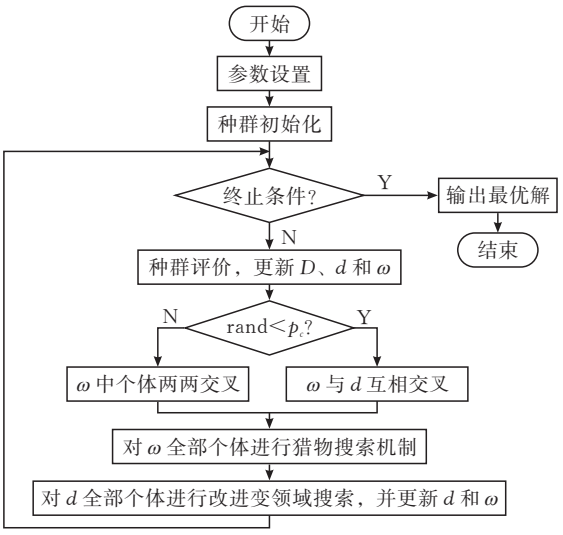


图 3 算法流程

3 实验结果及分析

本文选用 MK01-MK10 经典数据集对求解 FJSP 问题的算法进行验证，这些数据集的工件数量、机器数量和工序数量各不相同，能够有效检验算法的效率和精度。将本文提出的改进灰狼优化算法(IGWO)，与传统 GWO 算法、BBO 算法[10]、ABC 算法[11]以及 HGWO 算法[1]进行对比，通过求解上述数据集展开实验。实验采用 Python3.8 作为编程语言验证算法有效性，计算环境配置为：CPU 型号 8250U、内存 16 GB、Windows10 操作系统。IGWO 算法的参数设置如下：迭代次数为 100，初始化种群规模为 500，GS 为 0.6，LS 为 0.3，RS 为 0.1，变邻域搜索最大迭代次数为 20。对各算例分别进行 10 次求解，记录每次求解后的最优值和平均值，相关结果如表 1 所示。

表 1 标准算例结果比较

算例	$n\times m$	GWO 算法		BBO 算法		ABC 算法		HGWO 算法		IGWO 算法	
		best	Avg	best	Avg	best	Avg	best	Avg	best	Avg
MK01	10×6	44.00	48.60	**40.00**	41.00	**40.00**	40.00	**40.00**	41.60	**40.00**	40.00
MK02	10×6	39.00	41.62	28.00	28.25	**26.00**	26.50	29.00	30.30	**26.00**	26.90
MK03	15×8	239.00	244.40	204.00	204.00	204.00	204.00	204.00	204.10	**202.00**	204.40
MK04	15×8	76.00	78.35	64.00	66.00	**60.00**	61.22	65.00	67.40	**60.00**	61.20
MK05	15×4	192.00	195.45	173.00	173.50	**172.00**	172.98	175.00	178.20	173.00	173.98
MK06	10×15	114.00	119.35	66.00	66.50	**60.00**	64.48	79.00	79.90	**60.00**	62.55
MK07	20×5	198.00	205.05	144.00	144.25	**139.00**	141.42	149.00	156.40	**139.00**	140.75
MK08	20×10	538.00	544.55	**523.00**	523.00	**523.00**	523.00	**523.00**	523.00	**523.00**	523.00
MK09	20×10	388.00	409.15	310.00	310.75	**307.00**	308.76	325.00	342.30	310.00	310.80
MK10	20×15	335.00	345.05	230.00	230.75	208.00	212.84	253.00	262.70	**204.00**	216.45

“best”代表 10 次求解过程中最小化最大完工时间的最优解；“Avg”代表 10 次求解最小化最大完工时间的平均值；加黑数字为 5 种算法求解结果对比后的最优解。

由表 1 的结果可知：本文提出的 IGWO 算法与 ABC 算法在 10 个算例中最优结果都达到了 8 个，但二者存在细微差异；ABC 算法在 MK03 和 MK10 算例中的表现逊于 IGWO 算法，而 IGWO 算法在 MK05 和 MK09 两个算例结果上则略逊于 ABC 算法。BBO 算法和 HGWO 算法最优结果仅仅只有 2 个，传统 GWO 算法无最优结果。这一对比结果表明，IGWO 算法明显优于传统 GWO 算法、BBO 算法和 HGWO 算法。为进一步比较 IGWO 算法与 ABC 算法的性能差异，采用相对百分比偏差(RPD)值进行分析，其计算公式为：RPD=(Avg−best)/best。表 2 展示了 5 种算法在 10 个算例中的 RPD 计算结果。为便于对比分析，分别计算各算法的平均 RPD 值(记为 Mean)。由表 2 数据可知，传统 GWO 算法的 Mean 值为 0.402，显著高于其他 4 种算法，表明该算法的优化效果和鲁棒性最差；其次为 HGWO 算法(Mean 值第二大)和 BBO 算法(Mean 值第三大)。ABC 算法的 Mean 值为 0.020，略高于 IGWO 算法的 0.019，这说明本文提出的 IGWO 算法具有最优的鲁棒性，性能略微优于 ABC 算法。

表 2 RPD 结果对比

算例	算法				
	GWO	BBO	ABC	HGWO	IGWO
MK01	0.215	0.025	0.000	0.040	0.000
MK02	0.601	0.087	0.019	0.165	0.035
MK03	0.198	0.000	0.000	0.001	0.012
MK04	0.306	0.100	0.020	0.123	0.020
MK05	0.136	0.009	0.006	0.036	0.006
MK06	1.023	0.128	0.093	0.350	0.043
MK07	0.475	0.038	0.017	0.125	0.013
MK08	0.041	0.000	0.000	0.000	0.000
MK09	0.333	0.012	0.006	0.115	0.003
MK10	0.691	0.131	0.043	0.288	0.061
平均数	0.402	0.053	0.020	0.125	0.019

为了验证本文算法在同类算法中的收敛性能优势，图 4 展示了 IGWO 算法、HGWO 算法和 GWO

算法在算例 MK03 上的收敛曲线对比。在相同参数设置和计算环境下,对三者的收敛特性进行分析:传统 GWO 算法虽然初始收敛速度最快,但最终优化效果最差,表明该算法较早陷入局部最优;HGWO算法的求解质量优于 GWO 算法,但其收敛速度低于IGWO算法。综合来看,IGWO 算法在收敛速度和最终解质量上均略微优于HGWO算法,而 GWO 算法由于陷入局部最优导致优化效果最差。这一结果表明,本文提出的改进变邻域搜索策略能够有效提升算法的局部搜索能力,增强算法稳定性。因此,在求解单目标 FJSP 问题时,IGWO 算法在同类算法中具有显著的有效性优势。

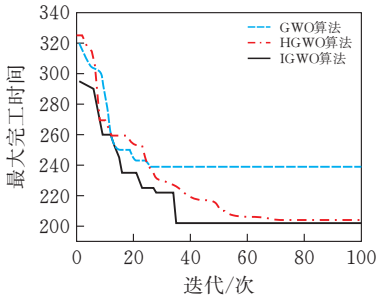


图 4　MK03 收敛对比

IGWO 算法求解 MK03 算例的最优甘特图如图 5 所示。

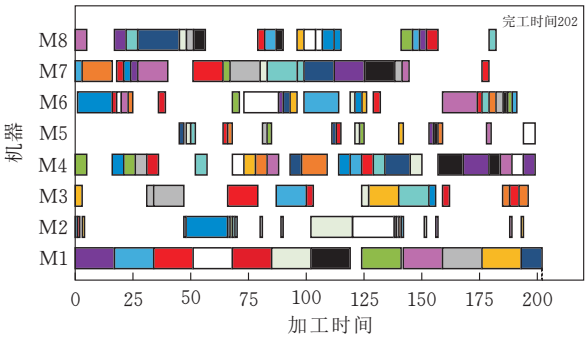


图 5　MK03 算例甘特图

4　结束语

本文围绕柔性作业车间调度问题(FJSP),以最小化最大完工时间为优化目标,构建相应数学模型,并提出一种改进灰狼优化算法(IGWO)用于求解。具体研究内容包括:创新采用 MSOS 分段编码与解码策略,可快速生成符合约束条件的可行解,有效提升算法求解效率;设计 GLR 初始化方法,在保证初始解质量的同时,维持种群多样性,为算法后续寻优奠定良好基础;引入交叉和变异算子,通过合理调节算子参数,实现算法全局搜索能力与局部开发能力的平衡,避免算法陷入局部最优;提出改进变邻域搜索策略,进一步增强算法的局部搜索性能,助力算法获取更优解。

在实验环节,通过对 MK 标准数据集的求解,并与其他经典算法进行对比分析,充分验证了IGWO算法在解决以最小化最大加工时间为目标的 FJSP 问题上的有效性。研究结果显示,相较于传统 GWO 算法及其他对比算法,IGWO 算法在收敛速度和求解精度方面均展现出显著优势。后续研究将聚焦于多算法融合策略的探索,并尝试将 IGWO 算法应用于实际生产调度场景。

[参 考 文 献]

[1] 姜天华.混合灰狼优化算法求解柔性作业车间调度问题[J].控制与决策,2018,33(03):503-508.

[2] 姚远远,叶春明.求解作业车间调度问题的改进混合灰狼优化算法[J].计算机应用研究,2018,35(05):1310-1314.

[3] 王玉芳,曾亚志,蒋亚飞.基于自适应灰狼优化算法的柔性作业车间调度问题[J].现代制造工程,2022(07):1-10.

[4] 李稚,周双牛.面向绿色智能制造的高维多目标动态作业车间调度优化[J].运筹与管理,2023,32(01):47-53.

[5] ZHU Z W, ZHOU X H, CAO D S, et al. A shuffled cellular evolutionary grey wolf optimizer for flexible job shop scheduling problem with tree-structure job precedence constraints[J]. Applied Soft Computing Journal,2022,125: 109235.

[6] GU J C, JIANG T H, ZHU H Q, et al. Low-carbon job shop scheduling problem with discrete genetic-grey wolf optimization algorithm[J]. Journal of Advanced Manufacturing Systems,2020,19(01): 1-14.

[7] JIANG T H, ZHANG C, ZHU H Q, et al. Energy-efficient scheduling for a job shop using grey wolf optimization algorithm with double-searching mode[J]. Mathematical Problems in Engineering, 2018, 2018: 8574892.

[8] WANG C Y, ZHAO L, LI X Y, et al. An improved grey wolf optimizer for welding shop inverse scheduling[J]. Computers & Industrial Engineering, 2022, 163: 107809.

[9] 杨宏安,孙启峰,李锦远.求解作业车间提前/拖期调度问题的一种新解码方法[J].计算机集成制造系统,2011,17(12):2652-2659.

[10] RAHMATI S, ZANDIEH M. A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem[J]. International Journal of Advanced Manufacturing Technology, 2012,58(9-12): 1115-1129.

[11] WANG L, ZHOU G, XU Y, et al. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem[J]. International Journal of Advanced Manufacturing Technology, 2012,60:303-315.

(下转第 49 页)