

文章编号: 2095-2163(2024)03-0140-07

中图分类号: TH165

文献标志码: A

强化哈里斯鹰算法求解柔性车间调度问题

余晓东, 叶春明

(上海理工大学 管理学院, 上海 200093)

摘要: 针对以最小化最大完工时间为优化目标的单目标柔性作业车间调度问题, 本文提出了强化哈里斯鹰优化算法。通过 Circle 混沌映射改进算法初始种群的质量, 提高算法的收敛速度; 采用正余弦策略改进逃逸能量的变化方式, 并将逃逸能量与跳跃强度相结合, 以此提升算法的全局探索能力, 预防算法陷入局部最优; 使用柯西扰动的差分进化策略提高算法的局部搜索能力以及寻优性能。通过柔性作业车间调度问题典型算例的对比试验, 验证了强化哈里斯鹰算法能够有效求解柔性作业车间调度问题。

关键词: 柔性作业车间调度; 哈里斯鹰优化; 混沌映射; 柯西扰动; 差分进化

Enhanced Harris Hawk optimization algorithm for solving flexible job shop scheduling problems

YU Xiaodong, YE Chunming

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

Abstract: For the single-objective flexible job shop scheduling problem with the optimization objective of minimizing the maximum completion time, this paper proposes the enhanced Harris Hawk optimization algorithm. The quality of the initial population of the algorithm is improved by Circle chaotic mapping to improve the convergence speed of the algorithm. A sine cosine strategy is used to improve the variation of escape energy, and the escape energy is combined with the jump strength to improve the global exploration ability of the algorithm and prevent the algorithm from falling into local optimum. The differential evolution strategy with Cauchy perturbation is used to improve the local search ability of the algorithm and the performance of the optimization search. Through comparative experiments with typical examples of flexible job shop scheduling problems, it has been verified that the enhanced Harris Hawk algorithm can effectively solve flexible job shop scheduling problems.

Key words: flexible job shop scheduling; Harris Hawk optimization; chaotic map; Cauchy perturbation; differential evolution

0 引言

柔性作业车间调度问题 (FJSP) 作为调度研究中极为重要的作业车间调度的基本延伸问题, 已经被确认为非确定性多项式 (NP-hard) 问题。与基本作业车间调度问题相比, 柔性作业车间问题可以被细化为工序排序和机器选择两个子问题, 有更广阔的解空间以及更加复杂的调度方式。目前, 针对柔性作业车间调度的研究已经相当成熟, 主要集中在使用元启发式算法求解, 包括群智能算法以及进化类算法。杜凌浩等学者^[1] 使用联合邻域搜索策略以及种内竞争策略改进的候鸟优化算法求解柔性作业车间调度问题, 并通过与多种算法的实验对比验证其算法改善的有效性。He 等学者^[2] 在求解机器

和工人双资源约束条件下的柔性作业车间调度问题时, 用三段式编码表达可行解, 利用内存库和邻域搜索等手段对秃鹫优化算法进行改进, 最终通过实验案例证明算法有更强的效果。吴晓雯等学者^[3] 结合粒子群算法与遗传算法, 采用交叉搜索的变异式粒子进化等方式, 避免算法陷入局部最优, 求解出合适的解。马锴等学者^[4] 在灰狼优化算法中引入模拟退火、莱维飞行和汉明距离等改进算法, 从而得到柔性车间调度问题的更优解。李宝帅等学者^[5] 使用正余弦优化策略改善鲸鱼优化算法中解的分布, 从而提高算法的全局搜索能力, 并以此求解柔性作业车间调度问题。姜天华^[6] 在对灰狼优化算法中的头狼群体使用变邻域搜索策略, 以此增强算法的全局搜索能力, 在种群迭代过程中引入变异与交叉

作者简介: 余晓东 (1999-), 男, 硕士研究生, 主要研究方向: 智能算法、车间调度。

通讯作者: 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。Email: yechm6464@

163.com

收稿日期: 2023-04-24

哈尔滨工业大学主办 ◆ 专题设计与应用

算子进一步提高算法的性能,通过与多种灰狼优化算法的改进算法结果对比,验证其在柔性作业车间调度问题中的有效性。Li等学者^[7]通过禁忌搜索算法和轮盘赌策略对人工蜂群算法做出改进,以提升其对于柔性车间调度问题的寻优效率。Lu等学者^[8]改进了多目标灰狼优化算法,将遗传算法中的遗传因子与灰狼优化算法巧妙地结合,以此来增强灰狼优化算法的全局搜索能力,并将改进后的算法与其余算法对比来证明优越性。刘雪红等学者^[9]利用精英分批策略对候鸟算法进行改进提升,使其能够解决多目标车间调度问题。

2019年,Heidari等学者^[10]提出哈里斯鹰优化算法(HHO),该算法模拟了哈里斯鹰群的捕猎习惯,根据逃逸能量来平衡全局搜索与局部搜索,具有参数少、寻优快等优点,目前被广泛应用于各个领域问题的解决。Shan等学者^[11]利用柯西变异改进哈里斯鹰优化算法,使其不易陷入局部最优,在30个CEC2017基准函数以及4个实例工程上验证了改进算法的有效性。Kamboj等学者^[12]提出了一种结合哈里斯鹰算法和正余弦算法的混合算法(hHHO-SCA),在CEC2017、CEC2018基准函数和11个工程应用上的实验结果证明了算法的有效性。Al-Betar等学者^[13]提出将哈里斯鹰优化算法与自适应 β 爬山算法相结合,将HHO强大的全局探索能力与自适应 β 爬山算法优秀的局部搜索能力相结合,通过解决电力系统中的经济负荷分配(ELD)问题验证其算法的有效性。陈立伟等学者^[14]使用非线性能量因子和混沌系统来改进哈里斯鹰优化算法,并将其应用于无人飞行器的路径规划问题。白晓波等学者^[15]将哈里斯鹰优化算法与聚类算法相结合,利用指数递减策略来改进算法性能,通过与其余算法相对比来体现其改进机制的优越性。Wang等学者^[16]使用传递函数对哈里斯鹰优化算法进行改进使其可以解决离散型问题,并最终应用于共享电动汽车的调度问题上。综上所述,目前哈里斯鹰算法的应用广泛,但很少应用于柔性作业车间调度问题上,本文提出改进哈里斯鹰算法,改进如下:

(1)利用Circle混沌映射初始化种群,使初始解均匀地分布于解空间中,提升初始解质量,从而提高算法的效率;

(2)利用正余弦策略改善逃逸能量的下降方

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)|, & q \geq 0.5 \\ (X_{rabbi}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)), & q < 0.5 \end{cases} \quad (3)$$

式,使其从线性下降变为曲线下滑,提高算法前期全局探索的强度以及后期收敛速度,并将逃逸能量与算法后期的跳跃强度相结合,提高算法后期跳出局部最优解的能力;

(3)提出结合柯西扰动的差分进化策略,增强算法的局部开发能力。

1 柔性作业车间调度问题

柔性作业车间问题可以描述为车间内有 n 个待加工零件($J_1, J_2, J_3, \dots, J_n$)在 m 台机器上加工,每个待加工零件 J_i 需经过至少一道以上的工序,用 O_{ij} 来表示待加工零件 J_i 的第 j 道工序,用 M_{ij} 来表示每道工序的可选机器集合,工序 O_{ij} 在不同的机器上加工时间并不相同。柔性作业车间的调度目标为在一定的约束条件下通过工序排序以及机器选择策略使得某种指标达到最优。约束条件如下:

(1)待加工零件之间不存在优先级与先后顺序约束,工序之间存在先后顺序约束;

(2)每台机器一次只能加工一个待加工零件,且加工一旦开始不可中断;

(3)所有待加工零件在零时刻就可以完全投入加工。

本文的加工性能指标是最小化最大完工时间,目标函数可表示为:

$$f = \min(\max(C)), 1 \leq j \leq n \quad (1)$$

其中, C 为最大完工时间、即最后一道工序的完工时间, n 为工件数。

2 基本哈里斯鹰优化算法

哈里斯鹰优化算法是一种仿生智能优化算法,主要分为2个阶段,即全局探索阶段和局部搜索阶段。由猎物的逃逸能量来控制其全局探索以及局部搜索的节奏,当前猎物的逃逸能量 E 由式(2)决定:

$$E = E_0(1 - \frac{t}{T}) \quad (2)$$

其中, t 为当前迭代次数; T 为最大迭代次数; E_0 为 $(-1, 1)$ 之间的随机数。

当猎物的逃逸能量的绝对值 $|E| > 1$ 时,猎物体力充足,鹰群进入全局搜索模式。

生成一个随机数 q ,来表示发现猎物和未发现猎物两种情况,位置更新见式(3):

其中, $X(t)$ 为当前哈里斯鹰个体的位置; $X_{rand}(t)$ 为当前代表种群内随机个体的位置; $X_{rabbit}(t)$ 表示当前猎物的位置; r_1, r_2, r_3, r_4 和 q 都是 $(0,1)$ 之间的随机数; $X_m(t)$ 表示种群个体的平均位置, 可由式(4)进行计算:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (4)$$

当猎物逃逸能量绝对值小于 1 时, 猎物体力不足, 鹰群进入围猎模式。此时生成随机数 p 来表示猎物是否逃脱包围圈, 根据 p 与逃逸能量绝对值的不同大小, 划分为 4 种围猎方式。

(1) 第一种: 强势围捕。当 $|E| < 0.5$ & $p \geq 0.5$ 时, 猎物未能逃出包围圈, 并且体力不足, 鹰群强势狩猎, 一次迭代时哈里斯鹰个体的位置 $X(t+1)$ 的数学定义公式为:

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \quad (5)$$

当前哈里斯鹰个体与猎物之间的位置差 $\Delta X(t)$ 的求值公式为:

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (6)$$

(2) 第二种: 诱导保留围捕。当 $|E| > 0.5$ & $p \geq 0.5$ 时, 猎物未能逃出包围圈, 但体力充足, 此时鹰群会对猎物进行诱导欺骗, 消耗体力, 位置更新公式为:

$$X(t+1) = \Delta X(t) - E|X_{rabbit}(t) - X(t)| \quad (7)$$

(3) 第三种: 诱导式追捕。当 $|E| > 0.5$ & $p \leq 0.5$ 时, 猎物逃出包围圈, 且体力充足, 此时使用 Levy 飞行模拟猎物的逃逸模式—蛙跳, 猎物的跳跃强度由式(8)决定:

$$J = 2(1 - rand) \quad (8)$$

其中, $rand$ 为取值 $(0,1)$ 之间的随机数。

鹰群使用轻柔围攻的方式对猎物进行诱导, 控制其方向并消耗其体力, 鹰群会根据式(9)的结果来决定下一步行动:

$$Y = X_{rabbit}(t) - E|JX_{rabbit} - X(t)| \quad (9)$$

如果诱导策略完成后, 适应度值没有改善, 则执行强制突袭, 强势突袭策略的公式具体如下:

$$Z = Y + S \times LF(D) \quad (10)$$

其中, S 为一个维度为 $1 \times D$ 的一维数组, $LF(D)$ 由 Levy 飞行函数得到。

结合式(9)和式(10), 诱导式追捕的最终决策为式(11):

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \quad (11)$$

(4) 第四种: 强势追捕。当 $|E| < 0.5$ & $p \leq 0.5$

时, 猎物虽然逃出包围圈, 但体力不足, 鹰群会尝试降低与所有猎物的平均距离, 如式(12)所示:

$$Y = X_{rabbit}(t) - E|JX_{rabbit} - X_m(t)| \quad (12)$$

其中, $X_m(t)$ 是由式(4)获得, 与诱导式追捕策略相似, 倘若式(12)并没有取得很好的效果, 则鹰群直接开始强势突袭, 如式(13)所示:

$$Z = Y + S \times LF(D) \quad (13)$$

结合式(12)和式(13), 强势追捕的最终决策为式(14):

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \quad (14)$$

3 强化哈里斯鹰优化算法

3.1 编码机制

FJSP 问题可以细分为工序排序与机器选择两个子问题, 本文针对工序排序子问题进行编码。种群中的每一个体代表着一种可行解的工序排序顺序的位置向量, 本文利用文献[17]中的最大价值位置规则(Largest Position Value, LPV), 先按照位置元素值的大小进行降序排列, 根据降序排列的顺序来决定工序编码的顺序, 实现工序顺序的排列。表 1 为工序排序过程示意(以 3 个工件为例)。

表 1 工序编码示意图

Table1 Schematic diagram of process coding

位置	项目				
	工序	位置向量	降序排列	改变后的工序	工序编码
1	O_{11}	0.32	0.84	O_{31}	3
2	O_{12}	0.21	0.68	O_{21}	2
3	O_{21}	0.68	0.32	O_{11}	1
4	O_{22}	0.30	0.30	O_{22}	2
5	O_{31}	0.84	0.21	O_{12}	1
6	O_{32}	0.15	0.15	O_{32}	3

3.2 机器选择策略

每一个种群个体的位置向量代表了一个当前 FJSP 算例的可行解, 对于已经确定了工序排序的排产方案而言, 机器选择会极大地影响最大完工时间。由于本文的目标为最小化最大完工时间, 所以本文的机器选择策略为计算当前工序下每个可选机器的完工时间, 选择当前工序加工完成时间更短的机器, 通过每一段的机器选择来尽可能缩短最大完工时间, 具体如下:

(1) 初始化: $i = 1, j = 1$, 创建工序完工时间列表, 机器加工时间列表, 2 张列表中的初值为 0;

(2) 根据工序排序顺序, 对第 i 个工序的加工机

器进行选择;

(3) 获取此工序的可用设备集,最大可选机器数量 j_{\max} 和可用设备集对应的工序加工时间集,该机器的最后一道工序的完工时间 T_1 ,此工序的前序完工时间 T_2 ,该机器的每一道已排产工序的开工时间和完工时间表;

(4) 判断此工序能否排产在选中机器已排产的每道工序的间隔时间中,若能,则插入工序,当前工序完工时间 $T_c = T_1$;若不能,判断 $T_1 > T_2$,若为真, $T_c = T_1 + T_2$;若为假, $T_c = T_1 + T_3$;

(5) $j = j + 1$,判断 $j > j_{\max}$,若为真,继续进行下一步骤;若为假,则回到步骤(4);

(6) 比较可用设备集内各个加工机器对应的当前工序的完工时间,选取当前工序的完工时间最小的机器,更新工序的完工时间表,将工序的开工时间和完工时间更新入机器加工时间表;

(7) $i = i + 1$,判断 $i > L$,若为真,继续下一步骤;若为假,则回到步骤(2);

(8) 完成机器选择。

3.3 混沌初始化种群

混沌映射是一种非线性动力学系统,可以产生高度随机、不可预测的序列,并且对于初始值高度敏感。在种群初始化阶段引入混沌映射能够使种群的初始解均匀地分布于解空间中,从而提高算法找寻最优解的速度,使算法不易陷入局部最优。本文使用的是 Circle 混沌映射,需要用到的公式为:

$$x_{i+1} = \text{mod}(x_i + 0.2 - (\frac{0.5}{2\pi})\sin(2\pi x_i), 1) \quad (15)$$

其中, mod 为取余函数。

由于混沌系统对于初始值高度敏感的特性,本文使用伪随机数生成器生成种群个体的第一个值,接着利用式(15)来完成种群个体位置向量的生成。

3.4 基于正余弦策略的逃逸能量与跳跃强度相结合

在基本哈里斯鹰算法中,逃逸能量由式(2)决定,随着迭代次数的不断增加,逃逸能量的绝对值呈线性递减态势,决定了哈里斯鹰优化算法全局搜索与局部搜索模式的转变与切换,其自适应衰减决定了算法的收敛性。而跳跃强度由式(9)决定,取值(0,2)范围内的随机数,每次迭代都随机变化,来模拟猎物的移动强度,猎物跳跃的过程有助于增强种群多样性,在迭代后期在一定程度上能够避免算法陷入局部最优。

正余弦函数的周期性可以帮助算法在进行全局

探索时勘探更大的区域,进行局部搜索时更加精确。基本哈里斯鹰算法用线性递减作为逃逸能量的自适应衰减手段,容易使算法前期全局搜索的强度较小,使算法出现如早熟、陷入局部最优等问题。跳跃过程虽然能够增强种群的多样性,但完全随机策略使得优化效果随机性过大,所以本文首先使用正余弦策略优化逃逸能量的自适应衰减曲线,改进方式如式(16)所示,以此来增强 HHO 的全局探索能力以及局部搜索能力:

$$E = 2\sin(\frac{\pi}{2}(\frac{t}{T}) + \frac{\pi}{2}) \quad (16)$$

再将跳跃强度与逃逸能量相结合,在实际中猎物的跳跃强度上限应该与体力挂钩,体力越高,跳跃强度越高,随着逃逸的过程逐渐接近尾声,猎物的体力越来越少,跳跃强度也越来越低。将跳跃强度与逃逸能量相结合,令跳跃强度上限为 $1 + |Escape_energy|$,由此推得:

$$J = (1 + |E|)(1 - rand) \quad (17)$$

3.5 结合柯西扰动的差分进化策略

差分进化是一种基于群体差异的启发式随机搜索算法^[18-19],来源于遗传算法,具有变异、交叉、选择等操作,利用式(18)生成新个体:

$$U_i = X_{r_1} + F(X_{r_2} - X_{r_3}) \quad (18)$$

其中, r_1, r_2, r_3 表示随机从种群中选中的3个个体的索引,三者与 i 为互不相同的整数;寻优因子 F 控制影响算法的全局寻优性能。

以此为基础进行式(19)的交叉操作:

$$v_{i,d} = \begin{cases} u_{i,d}, & rand(0,1) < CR \text{ or } d = d_{rand} \\ x_{i,d}, & rand(0,1) \geq CR \text{ or } d \neq d_{rand} \end{cases} \quad (19)$$

其中, CR 为交叉概率,生成的 V_i^t 为差分进化策略最终生成的新的种群个体。

考虑到算法后期逐渐收敛,随着种群同质化加深,差分进化策略的效果会逐渐减弱,随机选择种群中的3个,其差异也很有可能不足以令算法跳出局部最优,故引进柯西扰动策略,见式(20):

$$X_i^{t+1} = \begin{cases} rand(1,D) * Cauchy(0,1) * X_i^t, & d \leq MU \\ X_{i,j}^t, & d > MU \end{cases} \quad (20)$$

其中, $Cauchy(0,1)$ 为峰值位置为0、宽度参数为1的标准柯西分布随机数; d 为取值在(0,1)之间的随机数; MU 为变异概率。

原种群与差分进化产生的新个体以及柯西变异产生的新个体共同组成一个新的临时种群,采用贪

婪策略保留适应度值最优的个体。

3.6 算法实现步骤

本文提出的强化哈里斯鹰算法(EHHO)流程如图1所示。

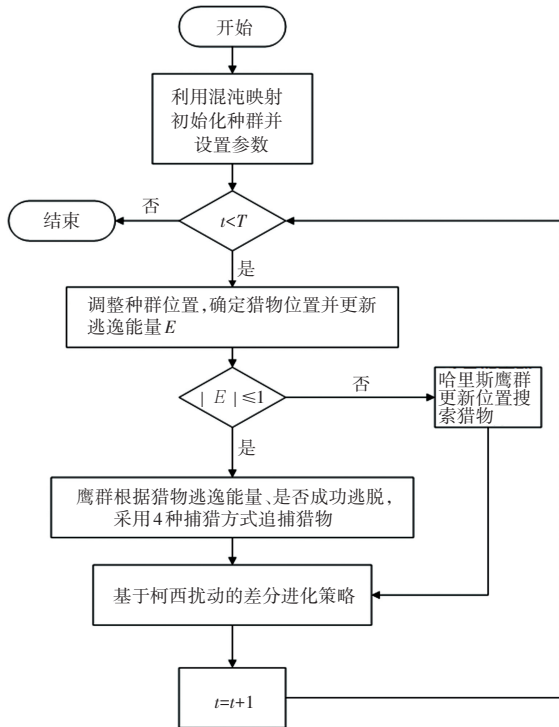


图1 强化哈里斯鹰优化算法流程图

Fig. 1 Flow chart of enhanced Harris Hawk optimization algorithm

具体步骤如下:

(1) 定义目标函数, 利用式(15)的 Circle 混沌映射来初始化种群, 并设置种群规模 N 、最大迭代次数 T 、逃逸能量 E 、跳跃强度 J 、概率 CR 、变异概率 MU 、寻优因子 F ;

(2) 进入循环迭代, 计算种群中每一个体的适应度, 选取最优个体作为当前种群猎物, 根据当前迭代次数更新猎物逃逸能量及跳跃能量;

(3) 根据逃逸能量, 选择位置更新策略, 当 $|E| > 1$ 时, 算法处于全局搜索状态, 采用式(3)更新位置; 当 $|E| < 1$ 时, 鹰群处于围猎状态, 根据逃逸能量以及猎物是否逃脱包围圈, 采用式(5)、式(7)、式(11)和式(14)来更新鹰群位置;

(4) 继续基于柯西扰动的差分进化, 将生成的新的种群个体与原种群个体合并成中间状态种群, 计算每一个体的适应度值, 并选取适应度值更优的数量为 T 的个体作为新种群;

(5) 判断迭代次数是否达到最大, 若是, 则结束迭代, 输出最优个体位置与目标函数最优解; 若不是, 则继续迭代。

4 实验分析

实验环境: Windows10 操作系统, AMD Ryzen 5 2600X Six-Core Processor 3.60 GH 处理器, NVIDIA GeForce GTX 1060 Ti 显示适配器, 16 GB 内存。选取文献[20]中 BRdata 包含的 10 个标准柔性作业车间调度算例(MK01~MK10)来测试改进哈里斯鹰优化算法的效果。

4.1 算法对比

为了降低算法运行结果的偶然性, 在实验阶段每种算法都会运行 20 次, 先将强化哈里斯鹰优化算法(EHHO)与基本哈里斯鹰优化算法进行对比, 选用标准差来表示算法寻优结果的稳定性, 对比结果见表 2。

表 2 EHHO 与 HHO 算法实验对比

Table 2 Experiments comparison of EHHO with HHO

算例	规模	HHO				EHHO				提升率/ (% · min)	提升率/ (% · s)
		最优结果	最大结果	平均寻优结果	标准差	最优结果	最大结果	平均寻优结果	标准差		
MK01	10×6	42	42	42.00	0	40	41	40.80	0.40	5.00	-
MK02	10×6	29	30	29.80	0.40	28	28	28.00	0	3.57	100.00
MK03	15×8	204	204	204.00	0	204	204	204.00	0	0.00	-
MK04	15×8	67	72	69.50	1.71	65	67	66.50	0.76	3.08	55.28
MK05	15×4	180	185	182.00	1.91	177	179	177.50	0.76	1.69	60.11
MK06	10×15	69	75	72.44	1.71	65	69	67.00	1.33	6.15	21.89
MK07	20×5	151	156	153.67	1.70	144	146	144.83	0.69	4.86	59.57
MK08	20×10	523	528	524.80	1.83	523	523	523.00	0	0.00	100.00
MK09	20×10	323	332	327.50	3.10	311	316	313.00	1.91	3.86	38.14
MK10	20×15	237	250	244.00	3.65	223	228	226.44	1.64	6.28	55.07

由表2可以看出 EHHO 无论从最优寻优结果、平均寻优结果还是寻优结果的稳定性上, 都有明显的提升, 在中小规模算例 MK01 ~ MK05 上, 寻优结果虽然有一定的改进, 但并无明显的提高, 寻优结果的稳定性提升非常明显; 在大规模算例 MK06 ~ MK10 上, 寻优结果与寻优稳定性的提升都非常明显, 在 MK06 与 MK10 的最优寻优结果上都提升了 6%, 稳定性也分别提升了 21.89%、55.07%。综上所述, 在基本参数设定保持一致的情况下, 强化哈里斯鹰优化算法的效果更好、鲁棒性更强。

为了更好地体现本文对哈里斯鹰优化算法的改进效果, 将强化哈里斯鹰优化算法与基本哈里斯鹰优化算法 (HHO)、灰狼优化算法 (GWO)、蜣螂优化算法 (DBO)、柯西变异优化的哈里斯鹰优化算法 (CHHO) 以及结合自适应 β 爬山算法的混合哈里斯鹰优化算法 (HHHO) 进行对比实验, 算法的基本参数保持一致, 种群规模统一设置为 60, 最大迭代次数为 200, 实验结果见表 3。

表3 EHHO、DBO、GWO、CHHO、HHHO 算法实验对比

Table 3 Experiments comparison of EHHO, DBO, GWO, CHHO, and HHHO

算例	规模	DBO	GWO	CHHO	HHHO	EHHO
MK01	10×6	40	41	42	41	40
MK02	10×6	29	29	29	29	28
MK03	15×8	204	204	204	204	204
MK04	15×8	68	69	69	70	65
MK05	15×4	183	178	181	182	177
MK06	10×15	68	69	65	66	65
MK07	20×5	148	150	149	149	144
MK08	20×10	526	523	523	523	523
MK09	20×10	335	324	315	325	311
MK10	20×15	243	240	235	237	223

从表3可以看出, EHHO 相比其余的智能优化算法, 在各个算例上都有着明显更优越的寻优效果, 特别是在机器和工件数量规模较大的算例上。由此可得本文所提出的强化哈里斯鹰优化算法优于其余群智能优化算法。

4.2 寻优曲线比较

实验在 MK01 ~ MK10 算例上各个智能算法的寻优曲线如图2所示。

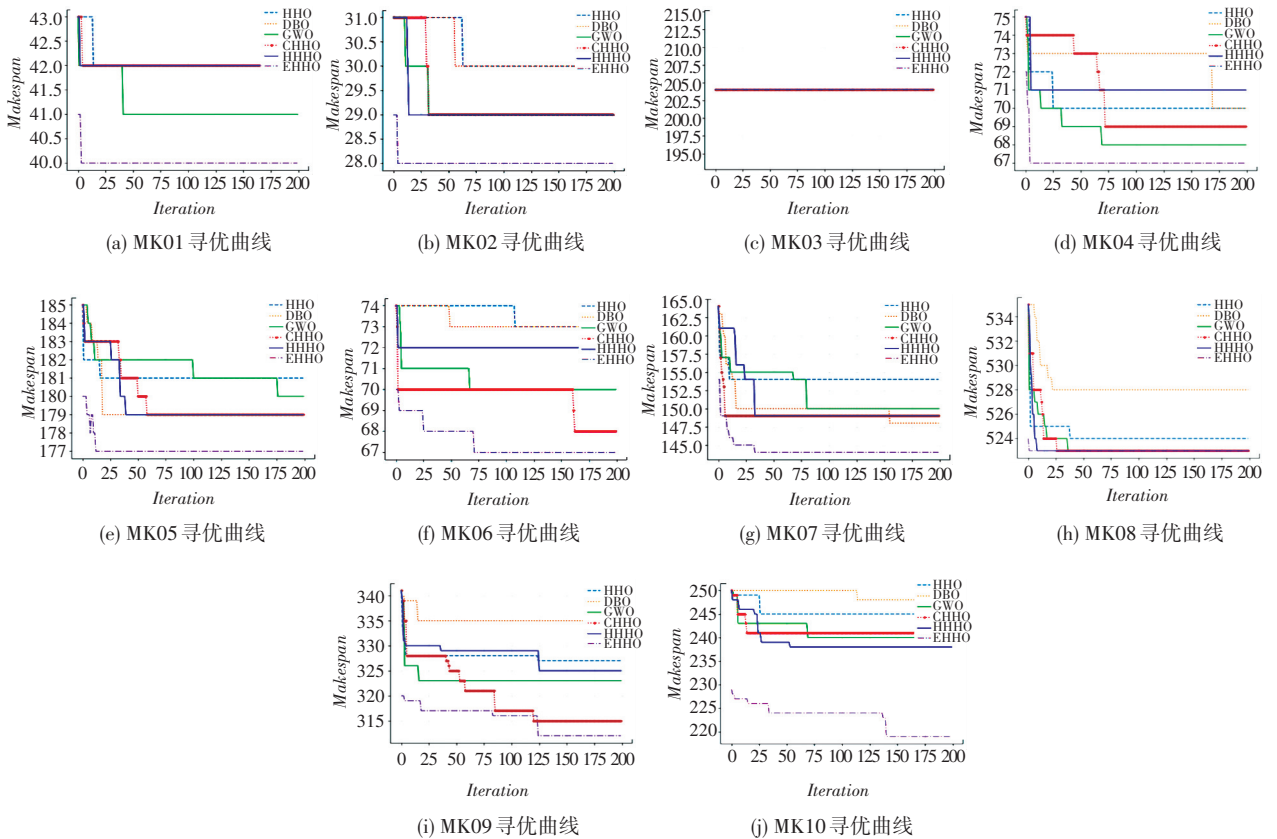


图2 寻优曲线对照图

Fig. 2 Comparison chart of optimization curve

从整体上看, EHHO 的初始解普遍优于其余算法, 而基于柯西扰动的差分进化策略能够很好地提高算法的收敛速度。对于较小规模的柔性车间调度

算例 MK01、MK02、MK04 和 MK05, EHHO 算法的收敛速度是最快的, 寻优效果也是最好的, HHHO 的收敛速度也很快, 但寻优效果不佳。在中等规模的

柔性车间调度实例 MK06、MK07 中,虽然 HHHO、CHHO、DBO 等智能算法收敛速度都曾超越过 EHHO,但却很容易陷入局部最优,寻优效果不如 EHHO。在大型柔性车间调度实例 MK08 - MK10 中,所有的智能算法都能寻到 MK08 算例的已知最优解,其中,EHHO 的收敛速度最快;而在 MK09 与 MK10 两个大型算例上,EHHO 在迭代后期依旧能够跳出局部最优,找寻到更优解。综上分析,本文提出的 EHHO 算法在柔性车间调度问题上拥有更加卓越的寻优能力,同时,在中小规模的调度实例问题上寻优能力略强于其余智能算法,且收敛速度更快;在大规模算例上虽然收敛速度略逊于其余算法,但寻优能力远远强于其余智能算法,算法综合能力较优,适用于求解柔性作业车间调度问题。

5 结束语

哈里斯鹰优化算法作为一种群智能算法被广泛应用于各个领域,本文通过混沌映射初始化、正余弦策略改进逃逸能量以优化平衡算法全局探索与局部搜索,使用柯西扰动的差分进化策略来加强局部搜索能力。以标准算例对比试验验证算法的有效性,实验表明:强化哈里斯鹰优化算法拥有更强的寻优能力以及更佳的寻优稳定性。下一步工作将哈里斯鹰优化算法应用于更复杂的柔性作业车间调度问题上,为响应国家的碳中和等政策,可以将碳排放量作为优化目标。

参考文献

[1] 杜凌浩, 向凤红. 改进多邻域候鸟优化算法的柔性作业车间调度研究[J]. 兵器装备工程学报, 2022, 43(12): 299-306.

[2] HE Z, TANG B, LUAN F. An improved African vulture optimization algorithm for dual - resource constrained multi - objective flexible job shop scheduling problems [J]. Sensors, 2022, 23(1): 90.

[3] 吴晓雯, 郑巧仙. 基于改进粒子群的柔性作业车间调度问题优化研究[J]. 湖北大学学报(自然科学版), 2022, 44(5): 501-507.

[4] 马镔, 郭丙文, 黄燕, 等. 基于改进离散狼群算法的柔性作业车间调度[C]//第34届中国控制与决策会议. 合肥: 东北大学、中国自动化学会, 2022: 548-552.

[5] 李宝帅, 叶春明. 混合鲸鱼优化算法求解柔性作业车间调度问题[J]. 计算机系统应用, 2022, 31(4): 244-252.

[6] 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(3): 503-508

[7] LI Xixing, PENG Zhao, DU Baigang, et al. Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems [J]. Computers & Industrial Engineering, 2017, 113: 10-26.

[8] LU Chao, GAO Liang, LI Xinyu, et al. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real - world welding industry [J]. Engineering Applications of Artificial Intelligence, 2017, 57: 61-79.

[9] 刘雪红, 段程, 王磊. 基于改进候鸟算法的柔性作业车间分批调度问题[J]. 计算机集成制造系统, 2021, 27(11): 3185-3195.

[10] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: Algorithm and applications [J]. Future Generation Computer Systems, 2019, 97: 849-872.

[11] SHAN Weifeng, HE Xinxin, LIU Haijun, et al. Cauchy mutation boosted Harris Hawk algorithm: Optimal performance design and engineering applications[J]. Journal of Computational Design and Engineering, 2023, 10: 503-526.

[12] KAMBOJ V K, NANDI A, BHADORIA A, et al. An intensify Harris Hawks optimizer for numerical and engineering optimization problems[J]. Applied Soft Computing, 2020, 89: 106018.

[13] AI-BETAR M A, AWADALLAH M A, MAKHADMEH S N, et al. A hybrid Harris Hawks optimizer for economic load dispatch problems[J]. Alexandria Engineering Journal, 2023, 64: 365-389.

[14] 陈立伟, 马泽华, 王桐, 等. 基于改进哈里斯鹰算法的无人飞行器路径规划[J/OL]. 应用科技: 1-8[2024-02-20]. <http://kns.cnki.net/kcms/detail/23.1191.U.20240205.1026.002.html>.

[15] 白晓波, 江梦茜, 王铁山, 等. 聚类质心与指数递减方法改进的哈里斯鹰算法[J]. 计算机与现代化, 2023(12): 30-35.

[16] WANG Yuheng, ZHOU Yongquan, LUO Qifang. Parameter optimization of shared electric vehicle dispatching model using discrete Harris Hawks optimization[J]. Mathematical Biosciences and Engineering, 2022, 19(7): 7284-7313.

[17] WANG Ling, PAN Quanke, TASGETIREN M F. Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms[J]. Expert Systems with Applications, 2010, 37(12): 7929-7936.

[18] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey Wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.

[19] XUE Jiankai, SHEN Bo. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization [J]. The Journal of Supercomputing, 2023, 79(6): 7305-7336.

[20] BRANDIMARTE P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41: 157-183.