DOI: 10.7641/CTA.2017.60768

基于新型蛙跳算法的低碳柔性作业车间调度

艾子义†, 雷德明

(武汉理工大学 自动化学院, 湖北 武汉 430070)

摘要: 针对低碳柔性作业车间调度问题(flexible job shop scheduling problem, FJSP), 提出一种新型蛙跳算法(shuffled frog leaping algorithm, SFLA)以总碳排放最小化, 该算法运用记忆保留搜索所得一定数量的最优解, 并采取基于种群和记忆的种群划分方法, 应用新的搜索策略如全局搜索与局部搜索的协调优化以实现模因组内的搜索, 取消种群重组使算法得到简化. 采用混合遗传算法和教–学优化算法作为对比算法, 大量仿真对比实验验证了SFLA对于求解低碳FJSP具有较强的搜索能力和竞争力.

关键词: 柔性作业车间; 碳排放; 蛙跳算法; 记忆

中图分类号: TP18　　**文献标识码**: A

A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling

AI Zi-yi†, LEI De-ming

(School of Automation, Wuhan University of Technology, Wuhan Hubei 430070, China)

Abstract: In this paper low carbon flexible job shop scheduling problem (FJSP) is considered. A new shuffled frog leaping algorithm (SFLA) is proposed to minimize total carbon emission, in which memory is used to store best solutions. Population division is done by using population and memory. Some new strategies such as cooperation of global search and local search are applied to realize the search in the memeplex. Population shuffling is deleted to simplify the algorithm. We compared hybrid genetic algorithm and teaching-learning-based optimization algorithm, which also considered the combination of local search and global search. Extensive experiments are conducted on a number of instances and result analyses show that SFLA has strong search ability and competitiveness for low carbon FJSP.

Key words: flexible job shop; carbon emission; shuffled frog leaping algorithm; memory

1 引言(Introduction)

柔性作业车间调度问题(flexible job shop scheduling problem, FJSP)是经典作业车间调度问题(job shop scheduling problem, JSP)的扩展, 由于突破了加工机器的唯一性约束, 它比JSP更接近实际生产环境. FJSP中工序可在多台可选机器上加工, 机器分配的存在扩大了可行解的搜索范围, 也增加了问题的复杂性和求解难度. 同时柔性的引入可以避免加工过程中的阻塞和拥挤现象, 使生产系统具有维持生产稳定的能力, 减少了中间装卸和搬运等造成的时间消耗, 从而缩短生产周期. 因此, 有必要深入研究FJSP.

能耗是制造过程的一个重要因素, 它不仅关系到企业的经济效益, 而且和企业的社会责任密切相关. 在能源供应日益紧张和价格日益高涨的环境下, 开展以节能为目标的低碳调度研究具有重要的理论和实际意义. 最近, 各种环境下如流水车间[1–7]和作业车间[8–15]环境下的低碳调度问题受到广泛关注.

低碳流水车间调度的研究取得了较大进展. Dai等[2]在柔性流水车间环境下提出了一种遗传–模拟退火算法, 并研究了最大完成时间和总能耗之间的冲突关系. Luo等[3]针对考虑电力消耗成本的混合流水车间调度提出了一种新的蚁群优化算法. Lin等[4]建立了一种针对加工参数优化和流水车间调度的集成模型, 并提出了一种教–学优化算法(teaching-learning-based optimization, TLBO)以最小化makespan和碳排放. Nagasawa等[5]建立了低碳流水车间调度的模型. Mansouri等[6]运用混合整数线性多目标优化模型和启发式算法解决了考虑最大完成时间和总能耗的双机

收稿日期: 2016−10−18; 录用日期: 2017−05−25.

†通信作者. E-mail: aiziyi1026@163.com; Tel.: +86 18771093491.

本文责任编委: 冯祖仁.

国家自然科学基金项目(61573264, 71471151, 61374151)资助.

Supported by National Natural Science Foundation of China (61573264, 71471151, 61374151).

流水车间调度. Ding等[7]应用多目标算法解决了置换流水车间低碳调度问题.

关于低碳FJSP, Tang等[9]考虑了FJSP的能源消耗并运用遗传模拟退火算法求解. Liu等[10]使用遗传算法求解双目标低碳FJSP. He等[11]提出了一种节能优化方法, 他们利用机床选择来减少机器加工能耗, 并调整操作序列以减少机器闲置时的能源浪费. Lei和Guo[12]应用一种动态邻域搜索解决了具有区间加工时间和包括碳排放在内的双目标双资源JSP. Zhang等[13]设计结合改进局部搜索的多目标遗传算法以解决低碳JSP. 蒋增强等[14]建立了低碳策略下多目标FJSP的求解模型, 并设计了基于血缘变异的改进非支配排序遗传算法. 唐立力[15]提出一种改进型候鸟优化算法以求解柔性作业车间中的低碳调度问题.

以上研究具有如下特点: ① 现有研究主要关注流水车间、作业车间以及柔性流水车间环境下的低碳调度问题, 而柔性作业车间作为一种重要的常见车间环境, 其低碳调度问题的相关研究较少; ② 现阶段主要是综合考虑碳排放和其他目标如探讨最大完成时间与碳排放之间的冲突关系, 很少单独以碳排放或总能耗为目标对问题进行优化, 而单独以总碳排放为目标, 可以弄清楚低碳FJSP的各子问题与碳排放之间的关系, 并探讨在各子问题间如何合理分配计算资源以更好地减少碳排放.

蛙跳算法(shuffled frog leaping algorithm, SFLA)由Eusuff和Lansey[16]于2003年提出, 该算法它结合了粒子群算法的思想和文化基因算法的进化特征. SFLA已成功应用于流水车间[17–18]、作业车间[19]和混合流水车间调度问题[20–21]. 潘全科等[17–18]研究了以提前或拖后惩罚指标为目标的批量流水车间调度问题, 给出了问题的数学模型, 提出了一种离散SFLA, 并结合扰动策略、模拟退火概率接受准则和插入邻域搜索对该算法进行改进. Lei等[20]提出了一种混合SFLA用于解决最小化两主体目标的总和的混合流水车间调度问题, 在之后的研究中进一步改进SFLA, 并在外包总成本不超过给定上限的情况下最小化总延迟时间[21]以上应用表明, SFLA对于调度问题具有较强的搜索优势和能力, 不过, 目前SFLA还没有应用于FJSP和低碳调度问题的求解.

本文研究以总碳排放为目标的低碳FJSP, 提出了一种新型SFLA, 其主要特征如下: 它采取基于种群和记忆的种群划分策略, 实现模因组内新的搜索策略如局搜索与局部搜索的协调优化, 并取消种群重组以简化算法的结构. 最后, 通过计算实验验证SFLA的优势和竞争力.

2 问题描述(Problem description)

低碳FJSP由n个独立工件和m台机器组成, 其中工件集合为$J=\{J_1,J_2,\cdots,J_n\}$, 机器集合为$M=\{M_1,M_2,\cdots,M_n\}$. 工件J_i具有h_i道工序, 其中o_{ij}表示工件J_i的第j道工序. 每道工序o_{ij}均可在一组相容机器中的任意一台$M_k\in M$上加工. 每台机器具有d种速度, $V=\{v_1,v_2,\cdots,v_d\}$为速度集合. 每个工件的加工过程中机器的加工速度一旦确定就不能改变. 在生产过程中, 机器存在两种不同状态: 工作状态和备用状态. 机器$M_k\in M$不加工工件时, 处于备用状态, 单位时间的该机器瞬时能耗为SP_k; 当机器M_k以速度v_l加工时, 工作状态下单位时间的能耗为PP_{kl}. 每道工序o_{ij}在机器$M_k\in M$上有一个给定的标准加工时间η_{ijk}, 当工序o_{ij}在机器M_k上以速度v_l加工时, 相应的加工时间$p_{ijkl}=\eta_{ijk}/v_l$.

关于p_{ijkl}和PP_{kl}的关系, Ding等[7]给出了一种假设, 即如果一个工件J_i在机器上以一个更高的速度加工, 它的加工时间将缩短但能耗上升. 对于低碳FJSP, 上述假设意味着

如果$\forall v_l>v_g, l,g\in\{1,2,\cdots,d\}$, 则

$$p_{ijkl}<p_{ijkg}, PP_{kl}\times p_{ijkl}>PP_{kg}\times p_{ijkg}. \quad (1)$$

低碳FJSP还要满足以下约束: 不同工件的工序之间没有先后顺序约束, 但同一工件的各道工序必须按照预先规定顺序完成; 每台机器在同一时刻最多只能加工一道工序; 每道工序同一时刻最多只能在一台机器上加工; 工序一旦开始加工不能中断等等.

问题的目标函数为总碳排放量(TCE):

$$\mathrm{TCE}=\varepsilon\int_0^{C_{\max}}\Big(\sum_{i=1}^{n}\sum_{j=1}^{h_i}\sum_{k=1}^{m}\sum_{l=1}^{d}PP_{kl}y_{ijkl}(t)+\sum_{k=1}^{m}SP_k z_k(t)\Big)\mathrm{d}t, \quad (2)$$

其中: 积分部分为总能量消耗, ε为能耗与碳排放量之间的转换系数, 通常取0.7559[22]. $y_{ijkl}(t)$和$z_k(t)$均为二进制变量. 如果在时间t工序o_{ij}在机器$M_k\in M$以速度v_l加工, 则$y_{ijkl}(t)$等于1; 否则$y_{ijkl}(t)$等于0. 如果机器$M_k\in M$在时间t处于备用状态, 则$z_k(t)$为1, 否则为0. $C_{\max}$是工件的最大完工时间.

低碳FJSP包含3个子问题: 调度、机器分配和速度选择, 其中速度选择确定工件在所分配机器上的加工速度. 速度选择和机器分配影响每台机器加工模式的时间长短, 从而影响机器运转时的碳排放; 而机器分配和调度决定了每台机器处于备用模式的时间, 从而决定机器空闲时的碳排放. 因此, 根据式(2), 需要综合考虑3个子问题, 合理分配每台机器处于加工模式和备用模式的时间, 工件的完工时间也会相应减少, 从而达到降低总碳排放的目的.

3 蛙跳算法描述(Description on SFLA)

SFLA[16]中, 青蛙的位置表示问题的解x, 一组虚拟青蛙的集合为可能解的种群, 种群被分解为不同的组, 这个组称为模因组. 对每个模因组执行搜索过程.

当每个模因组完成指定数量的迭代后, 将它们结合成新的种群, 这个过程称为种群重组. 搜索和重组过程持续进行直到满足结束条件.

基本SFLA的详细过程描述如下:

步骤 1　参数初始化: 种群大小(N), 模因组数量(s)和每个模因组的迭代次数;

步骤 2　随机产生初始种群P;

步骤 3　对种群中的解按适应度值从大到小降序排序;

步骤 4　将种群分为s个模因组;

步骤 5　对每一个模因组执行搜索过程;

步骤 6　对进化后的模因组执行重组;

步骤 7　如果满足终止条件, 输出最优解; 否则转到步骤 3.

将种群划分为s个模因组的过程如下: 第1只蛙分到第1个模因组M_1, 第2只蛙分入第2个模因组M_2, 第s只蛙分入第s个模因组M_s, 第$s+1$只蛙分入第1个模因组M_1, 依此类推.

每个模因组的搜索过程如下: 首先确定M_i中的最好解$x_{\rm b}$、最差解$x_{\rm w}$和全局最好解$x_g \in P$; 然后利用$x_{\rm b}$和$x_{\rm w}$产生一个新的位置$x_{\rm w}^{\rm new}$,

$$x_{\rm w}^{\rm new} = x_{\rm w} + \alpha\left(x_{\rm b} - x_{\rm w}\right), \tag{3}$$

其中α是区间[0, 1]内服从均匀分布的随机数.

如果$x_{\rm w}^{\rm new}$优于$x_{\rm w}$, 则直接最差解$x_{\rm w}$; 否则利用x_g和$x_{\rm w}$产生一个新位置$x_{\rm w}^{\rm new}$, 如果这个新位置优于$x_{\rm w}$, 则直接代替$x_{\rm w}$; 如果上述两步均不能产生新的$x_{\rm w}$, 则随机产生一个新位置直接代替$x_{\rm w}$; 重复上述步骤直到满足给定的迭代次数.

对每一个模因组的搜索完成后, 进化后的模因组被重组, 这些模因组的解重新结合并获得一个新的种群P, 然后, 对新种群中的解重新根据适应度值降序排列.

4　基于新型SFLA的低碳FJSP (A new SFLA for low carbon FJSP)

由于其新解产生公式的原因, 基本SFLA难以直接应用于低碳FJSP的求解, 为了解决调度问题, 需要对SFLA离散化, 构建离散SFLA.

为了解决低碳FJSP, 提出了一种新的SFLA: ① 其新解产生方式也是离散的, 模因组内的搜索在组内最好解而不是最差解上进行, 模因组中的所有解可以以相同的概率与最好解结合, 采用新的方式实现全局搜索和局部搜索的协调优化; ② 引入记忆用来保留搜索过程中产生的一定数量的精英解, 种群划分时, 运用记忆和种群P中的解构造模因组; ③ 取消了种群重组.

4.1　初始解(Initial solution)

由于低碳FJSP由3个子问题组成: 调度子问题、机器分配子问题和速度选择子问题, 采用3个串包括调度串、机器分配串以及速度选择串独立地表示问题的解, 这样可以按问题特点合理安排计算资源.

对于具有n个工件和m台机器的低碳FJSP, 调度串为$(z_1, z_2, \cdots, z_{OP})$, 机器分配串为$(\rho_{11}, \cdots, \rho_{1h_1}, \cdots, \rho_{i1}, \cdots, \rho_{ih_i}, \cdots, \rho_{n1}, \cdots, \rho_{nh_n})$, 速度选择串为$(u_{11}, \cdots, u_{1n_1}, \cdots u_{i1}, \cdots, u_{in_i}, \cdots, u_{n1}, \cdots, u_{nn_n})$, 其中串长均为$OP = \sum\limits_{1\leqslant i\leqslant n} h_i$, $z_i \in \{1, 2, \cdots, n\}$, ρ_{ij}表示工序o_{ij}的加工机器, u_{ij}为ρ_{ij}的加工o_{ij}的速度.

上述调度串和机器分配串均为整数串, 若采用基本SFLA中的式(3)产生新解, 可能会将整数串变成实数串, 并且可能使速度选择串中的部分值出现负数, 从而算法无法继续求解低碳FJSP. 而离散化的SFLA能够始终保持这些串的可行性, 因此构建离散SFLA求解该问题.

表1给出了具有3个工件、3台机器和6道工序的实例. 每台机器具有 5 种速度, 速度集合为$V = \{1.00, 1.30, 1.55, 1.75, 2.10\}$.

表 1　低碳FJSP的实例

Table 1　An example of low carbon FJSP

工序	M_1	M_2	M_3	工序	M_1	M_2	M_3
o_{11}	5	10	6	o_{22}	8	10	1
o_{12}	7	3	2	o_{31}	12	3	8
o_{21}	5	15	12	o_{32}	7	4	5

图1描述了上述实例的3个串, 其中调度串对应序列$o_{11} \succ o_{12} \succ o_{21} \succ o_{31} \succ o_{32} \succ o_{22}$, 即首先安排$o_{11}$的加工, 第2个被安排的工序为$o_{12}$, 依此类推, 最后一道工序为$o_{22}$. 机器分配串表示工序的机器分配情况, 即$\langle o_{11}, M_1\rangle$, $\langle o_{11}, M_1\rangle$, $\langle o_{12}, M_2\rangle$, $\langle o_{21}, M_1\rangle$,$\langle o_{22}, M_3\rangle$, $\langle o_{31}, M_2\rangle$和$\langle o_{32}, M_2\rangle$. 速度选择串(1.30,1.00, 1.75, 1.75, 1.55, 2.10)表示机器M_1加工o_{11}的速度为1.30, 机器M_2加工工序o_{12}的速度为1.00, 依此类推.

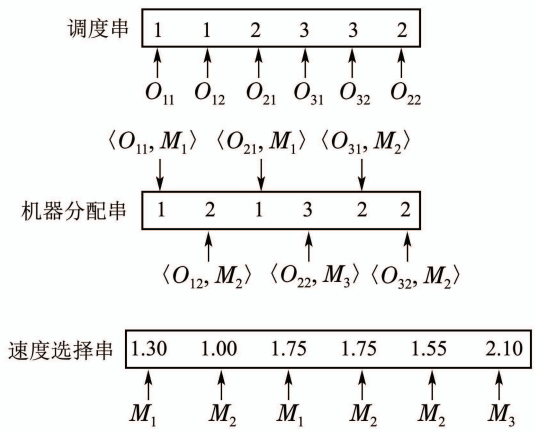


图 1　问题的编码表示

Fig. 1　Representation of the problem

4.2 构建模因组(Memeplex construction)

由于SFLA的搜索在模因组内进行, 导致种群划分或者模因组构建成为算法的重要步骤. 除了第3节描述的种群划分策略外, Lei等[20]采用二元锦标赛选择将种群划分成s个模因组, 从种群P中随机选择两个解x_1和x_2, x_1和x_2中较好解进入M_1, 较差解回到P中; 再从P中随机选择两个解, 较好解进入M_2, 较差解回到P中; 第s次锦标赛选择的较好解进入M_s, 第$s+1$次锦标赛选择的较好解进入M_1, 依此类推.

引入了记忆Ω, 它用来保留SFLA搜索过程所获得的总碳排放最小的解. 设记忆的最大容量为S, 初始记忆由种群P内总碳排放最小的S个解组成, 搜索过程中记忆的容量始终为S.

构建模因组时, 首先将记忆中的解拷贝到种群P中, 形成新的种群P', 采用二元锦标赛策略, 只是每次从P'中, 而不是仅从种群P中随机选择两个解, 两个解中的好解进入相应的模因组, 差解回到P'.

由于记忆中的解目标函数值相对较小, 且部分解如精英解也同时存在于种群P, 上述过程将使总碳排放较少的解多次被选中进入模因组, 同时种群内的最差解往往不能被选择, 这样可取得和遗传算法复制类似的优胜劣汰效果.

4.3 模因组的搜索过程(Search process within memeplex)

模因组搜索是SFLA产生新解的主要途径. 通常, SFLA模因组内的搜索过程由3步组成: 在组内最好解$x_{\rm b}$和组内最差解$x_{\rm w}$之间进行; 在全局最好解x_g和组内最差解$x_{\rm w}$之间进行; 随机产生一个解替代$x_{\rm w}$.

根据低碳FJSP子问题多且子问题单独表示的特点, 采用一些新的搜索机制:

1) 模因组M_i中的搜索过程由不同的3步组成: 在组内最好解$x_{\rm b}\in M_i$和一个随机选择的解$x\in M_i$的之间进行一次全局搜索; 两次针对$x_{\rm b}\in M_i$运用多邻域搜索产生新解, 以实现全局搜索和邻域搜索的协调优化;

2) 组内所有解都有机会参与组内搜索, 组内搜索的对象是$x_{\rm b}$, 而不是$x_{\rm w}\in M_i$.

4.3.1 全局搜索(Global search)

对于模因组M_i的最好解$x_{\rm b}$和随机选择的解$x\in M_i$, 从调度串中随机选择两点, $x_{\rm b}$在这两点之间的工件直接从$x_{\rm b}$复制到解y中, $x_{\rm b}$中剩余的工件根据它们在解x中的顺序重新安排并加入到解y的调度串中, 这样在保证可行性的情况下获得新解y. 针对4个工件10道工序的实例的调度串全局搜索如图2所示.

对于模因组M_i的最好解$x_{\rm b}$和随机选择的解$x\in M_i$, 随机决定两个位置$1\leqslant g_1<g_2\leqslant OP$, 对于$x_{\rm b}$的机器分配串, 将位于$g_1$和$g_2$之间的机器用$x$的机器分配串中相应位置上的机器替代.

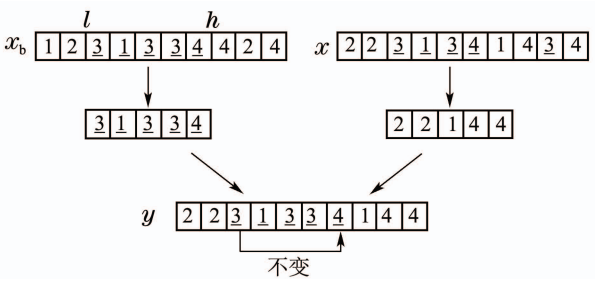


图 2 调度串全局搜索示意图

Fig. 2 Global search on schedule string

采用和机器分配串类似的搜索过程, 对$x_{\rm b}$和x的速度选择串进行全局搜索: 随机决定两个位置$1\leqslant g_1<g_2\leqslant OP$, $x_{\rm b}$的速度串中位于于g_1和g_2之间的速度值用解x的速度串中的相应位置的速度值代替.

上述过程将产生新解y, 如果y的总碳排放量小于或等于$x_{\rm b}$的总碳排放量, 用y代替$x_{\rm b}$并更新记忆; 否则, $x_{\rm b}$不更新. 其中记忆更新过程如下: $x_{\rm b}$被新解替代后, 将$x_{\rm b}$与记忆中的解进行比较, 若$x_{\rm b}$优于记忆中碳排放值最大的解, 则利用$x_{\rm b}$直接替代记忆中的最差解, 每次最多替代记忆中的一个解. 从上述过程可以看出, 精英解一定保留在记忆中.

由于存在3种全局搜索方式, 如何为这些搜索方式分配合适的计算资源, 将直接影响SFLA的搜索能力. 根据调度子问题求解难度和搜索空间比另外两个子问题更大的特点, 采用如下策略实现全局搜索: 产生区间$[0,1]$内服从均匀分布的随机数α, 如果$\alpha<\beta_1$, 则对调度串执行全局搜索; 如果随机数$\alpha\in[\beta_1,\beta_2)$, 则执行机器分配串的全局搜索; 如果以上条件都不满足, 则对速度串进行全局搜索.

由于调度子问题的求解难度和复杂度往往高于另外两个子问题, 且对总碳排放的影响也更大, 为此, 要求β_1大于0.5以更好地优化调度子问题. 通过大量实验得到如下结果: $\beta_1=0.7,\ \beta_2=0.85$.

4.3.2 多邻域搜索(Multiple neighborhood search)

由于存在3个子问题, 并结合上述编码机制, 采用基于4种邻域结构的多邻域搜索.

根据调度子问题复杂性更高的特点, 给出了两种邻域结构. 对于调度串, 邻域结构swap描述如下: 随机选择$z_i\neq z_j$, 互换它们在调度串中的位置. 邻域结构insert定义如下: 随机地选择z_i和一个位置j, 将元素z_i插入到位置j.

邻域结构change 1用于改变部分被选中工序的机器分配. change 1的详细步骤如下: 首先从机器分配串中随机选择ρ_{ij}, 它对应工序o_{ij}, 确定可以加工该工序的所有机器的集合Θ, 从集合Θ中随机选择一台与ρ_{ij}不同的机器代替ρ_{ij}.

邻域结构change 2用于改变部分被选中机器的加工速度. change 2的具体过程如下: 首先从速度选择串中随机选择u_{ij}, 它对应工序o_{ij}和加工机器ρ_{ij}, 从ρ_{ij}的加工速度集合V中随机选择一档与u_{ij}不同的速度代替u_{ij}.

对于邻域搜索产生的新解y, 采用和第4.3.1节相同的条件比较$x_{\rm b}$和y.

4.4 算法描述(Algorithm description)

新型SFLA的详细过程描述如下:

步骤 1　随机产生具有N个解的初始种群P和初始记忆, $r_i=0, i=1,2,\cdots,N,\cdots,N+S$;

步骤 2　重复步骤 3–4直到满足终止条件;

步骤 3　构建模因组;

步骤 4　对于每一个模因组$M_i(i=1,2,\cdots,s)$. 下列步骤重复μ次:

1) 确定$x_{\rm b}\in M_i$, $\sigma=0$;

2) 若$\sigma=0$, 产生一个[0, 1]内均匀分布的随机数α, 根据随机数大小执行相应串的全局搜索. 比较新解y与$x_{\rm b}$, 如果新解满足上述替代条件, 则利用新解更新$x_{\rm b}$和记忆且$\sigma\leftarrow\sigma+1$;

3) 若$\sigma=1,2$, 进行多邻域搜索: 如果$r_i=0$, 实施swap; 如果$r_i=1$, 使用insert; 如果$r_i=2$, 则执行change1; 如果$r_i=3$, 则执行change2. 如果$x_{\rm b}$能够被新解y代替, 则更新$x_{\rm b}$和记忆且$\sigma\leftarrow\sigma+1$; 否则, 在$\sigma=1$时, 令$r_i\leftarrow r_i+1$且$r_i\leftarrow 0$当$r_i=4$.

其中μ是一个整数.

通常, 对每一个模因组的搜索完成后, 需要对进化后的模因组被重组, 使进化后的组内解重新结合获得一个新种群P, 然后对新种群P中的解再次按照适应度值进行降序排列. 本文由于采用锦标赛方式并利用记忆构建模因组, 即使不进行种群重组, 新一轮的构建也能产生完成不同于当前模因组的新模因组, 取消种群重组能在达到相同效果, 还能使SFLA得到简化, 故取消种群重组.

由于SFLA与基本SFLA具有相同的模因组搜索过程复杂性, 导致两种算法的复杂性相同, 都是O(ZN^2), 其中Z是两种算法循环部分循环的次数.

5 数字仿真(Numerical simulation)

为了测试SFLA的性能, 运用一系列实例进行了大量的仿真. 所有仿真在Microsoft Visual C++ 6.0实现, 程序运行环境为4.0GRAM, 2.50 GHz CPU的个人计算机.

5.1 测试实例和对比算法(Test examples and comparative algorithms)

为了验证所提算法的性能, 本文以常用的MK1–15[23] 和DP1–12[24]作为算例. 由于本文考虑的低碳FJSP增加了速度选择子问题, 故增加了相关数据: $V=\{1.00,1.30,1.55,1.80,2.00\}$, $\mathrm{PP}_{kjl}=4v_l^2$ kW, $\mathrm{SP}_{kj}=1.00$ kW.

选择Xu等[25]提出的TLBO, Li等[26]提出的混合遗传算法(hybrid genetic algorithm, HGA)和唐立力提出的改进型候鸟优化算法(improved migratory bird optimization, IMBO)作为对比算法. 为了解决低碳FJSP, 对TLBO做如下扩展: 由于原算法没有对速度选择进行编码, 增加速度选择串, 并采用第4节过程对速度进行全局搜索, 增加邻域结构change 2. 为了应用HGA和IMBO求解低碳FJSP, 也做类似的扩展: 增加速度串以及相应的全局搜索方式和邻域结构.

评价指标为相对百分比偏差(relative percentage deviation, RPD), 其计算公式为RPD$=100(f-f^*)/f^*$, 其中f^*为所有算法获得的最好解. RPD的值越大表明相应的解越差. 每种算法关于每个实例随机运行20次, 其中ARPD为20组RPD的平均值, XRPD和NRPD表示在20次运行中得到的最大RPD和最小RPD. 这些指标的值越大, 表明相应的算法的性能越差.

5.2 结果分析(Result analyses)

为了进行算法对比, 先根据大量仿真, 获得SFLA的如下6个参数: 种群大小为40, 模因组个数为5, 模因组大小为8, 记忆容量为8, 模因组内搜索次数$\mu=100$以及目标函数的估计次数为$\max_it=10^5$.

对于HGA, 基于大量仿真得到如下参数: 最大遗传代数为$\max_it$/100, 种群规模为100, 交叉概率为0.8, 变异概率为0.1. 对于TLBO, 设置参数如下: 种群规模为100, 最好学生比例为0.2,

局部搜索次数tl为6. 对于IMBO, 设置参数如下: 种群大小为51, 邻域解个数为3, 共享邻域解个数为1, 巡回次数为10, 局部搜索最大迭代次数为10. 4种算法的结束条件完全相同.

为了更好地比较SFLA, TLBO, HGA和IMBO, 每种算法关于每个实例随机运行20次. 表2显示了4种算法的计算结果. f^*是这4种算法所获得的最好解. 4种算法的运算时间如表3所示. 图3描述了4种算法关于MK3和DP6的收敛曲线.

如表2所示, 总共27个实例中, SFLA获得了其中24个实例的最好解, HGA逼近了3个实例的最好解, 而TLBO和IMBO没有得到任何实例的最好解; SFLA关于几乎所有实例获得了优于HGA, TLBO和IMBO的ARPD; SFLA关于所有实例所获得的最差解明显优于HGA, TLBO和IMBO的相应解. 此外, 由于邻域结构设计简单, IMBO的运算速率明显优于其他3种算法, 但SFLA的运行时间均小于HGA和TLBO. 图3(a)和3(b)关于4种算法的收敛曲线也显示了SFLA具有优于另外3种算法的的性能.

表 2 SFLA, HGA, TLBO和IMBO的计算结果

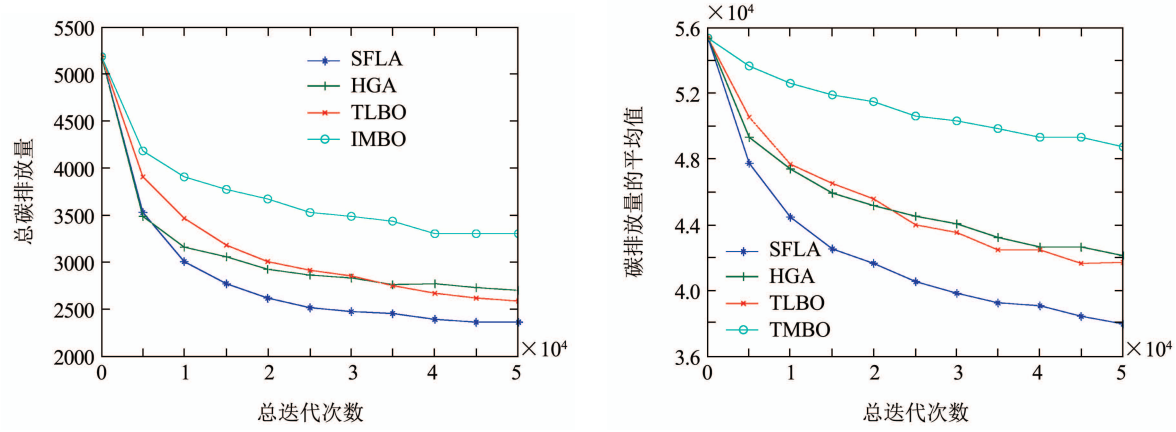
Table 2 Computational results of SFLA, HGA, TLBO and IMBO

实例	XRPD				NRPD				ARPD			
	SFLA	HGA	TLBO	IMBO	SFLA	HGA	TLBO	IMBO	SFLA	HGA	TLBO	IMBO
MK1	1.85	4.40	4.16	26.1	0.00	1.25	1.37	20.6	1.10	2.73	2.73	23.9
MK2	1.73	16.5	4.02	37.3	0.00	5.70	2.57	31.0	0.96	8.62	3.16	34.0
MK3	4.22	20.6	10.7	34.2	0.00	11.7	5.05	28.0	2.28	16.7	7.60	31.5
MK4	2.60	16.1	6.31	31.4	0.00	3.22	2.27	26.0	0.93	4.36	4.09	28.5
MK5	1.44	1.32	8.33	26.6	0.00	0.24	4.18	24.0	0.54	0.70	5.54	25.3
MK6	4.57	20.6	10.9	28.9	0.00	10.3	13.7	23.9	1.49	16.2	6.95	26.2
MK7	2.19	13.9	7.18	35.5	0.00	6.65	2.94	30.7	0.91	9.35	4.46	32.9
MK8	5.75	5.69	11.1	31.4	0.00	1.69	5.86	26.0	2.47	3.59	9.14	28.7
MK9	2.95	5.55	16.1	38.8	0.00	7.28	4.26	33.0	0.05	9.90	10.0	36.4
MK10	7.93	21.4	17.7	38.9	0.00	14.1	10.7	34.6	4.97	18.8	13.7	36.6
MK11	0.99	1.78	9.42	30.0	0.00	0.43	6.28	28.5	0.47	0.97	7.64	29.3
MK12	3.34	4.42	14.3	30.6	0.00	0.23	8.91	27.6	1.70	2.40	10.6	29.6
MK13	3.98	11.8	20.9	34.2	0.00	7.02	9.29	31.9	2.35	9.42	12.1	33.2
MK14	5.91	5.37	15.9	29.8	0.52	0.00	8.99	28.4	2.39	3.33	11.2	29.1
MK15	3.22	11.8	12.0	35.5	0.00	8.86	9.45	30.3	1.13	10.6	11.3	21.9
DP1	7.29	1.41	19.1	20.3	2.97	0.00	13.1	16.6	4.68	1.79	15.5	18.6
DP2	3.06	7.48	12.7	28.4	0.00	5.54	3.04	24.8	1.26	7.29	9.92	27.3
DP3	3.52	7.82	11.0	29.0	0.00	7.34	6.45	25.5	1.20	8.60	8.01	27.7
DP4	6.53	1.01	17.3	17.7	3.45	0.00	10.5	15.4	4.33	1.04	14.1	16.6
DP5	2.93	7.17	13.6	28.3	0.00	5.77	7.72	25.3	1.6	7.60	9.71	27.1
DP6	3.17	9.54	11.0	30.9	0.00	7.47	5.42	27.2	1.02	9.40	7.74	28.6
DP7	6.93	1.28	15.7	20.9	0.00	1.06	10.5	18.7	2.81	2.19	11.9	19.8
DP8	5.26	10.2	14.8	29.3	0.00	8.90	10.5	27.7	2.60	10.1	11.9	28.5
DP9	3.51	13.1	14.1	30.4	0.00	10.1	10.1	28.2	1.67	12.5	12.4	29.2
DP10	6.52	4.38	14.8	21.8	0.00	1.44	9.04	20.1	2.20	2.90	12.4	21.0
DP11	4.57	11.0	15.4	29.4	0.00	10.1	8.85	24.6	2.31	11.6	11.7	27.9
DP12	5.68	11.9	13.2	31.2	0.00	11.1	8.99	29.2	2.67	12.0	11.6	30.5

表 3 4种算法的运算时间

Table 3 The running time of four algorithms

实例	运行时间/s				实例	运行时间/s			
	SFLA	HGA	TLBO	IMBO		SFLA	HGA	TLBO	IMBO
MK1	2.07	3.88	3.66	1.56	MK15	19.6	29.0	21.2	12.3
MK2	2.45	4.72	4.39	1.57	DP1	8.80	19.0	19.1	5.70
MK3	6.05	10.3	9.96	3.50	DP2	12.6	21.5	19.5	8.70
MK4	3.24	5.65	5.57	1.82	DP3	13.2	21.6	21.8	8.71
MK5	5.18	10.3	9.70	3.25	DP4	10.7	20.7	22.7	7.73
MK6	5.76	9.13	8.63	3.50	DP5	10.7	21.8	19.0	7.70
MK7	5.74	9.56	9.09	3.46	DP6	13.9	19.2	19.1	9.69
MK8	6.57	12.6	12.0	3.76	DP7	18.7	33.1	29.7	12.8
MK9	15.0	23.8	25.7	10.4	DP8	22.3	29.9	28.9	14.4
MK10	16.8	24.4	22.9	10.3	DP9	25.5	30.3	38.3	14.6
MK11	9.76	21.2	19.8	7.79	DP10	21.6	37.3	32.9	14.5
MK12	13.5	23.8	22.3	9.27	DP11	23.3	36.5	35.4	14.4
MK13	16.0	24.8	23.8	9.22	DP12	25.2	37.5	37.7	14.8
MK14	16.2	27.7	25.5	9.94					



(a)　　(b)

图 3 SFLA, TLBO, HGA和IMBO关于实例MK3和DP6的收敛曲线

Fig. 3 The convergence curves of SFLA, TLBO, HGA and IMBO on MK3 and DP6

SFLA的优良性能主要在于它的模因组搜索在组内最好解而不是最差解上进行, 局部搜索和全局搜索的协调优化, 以及基于种群和记忆的模因组构建策略, 而种群重组的取消简化了算法步骤, 使其运算时间更短. IMBO 中邻域结构的设计只针对当前解进行局部搜索, 没有考虑不同解之间的信息交换, 因而没有获得较好的搜索性能. HGA和TLBO虽然也都考虑了局部搜索和全局搜索的融合, 但实验结果显示它们的性能没有SFLA 优异. 综上所述, SFLA是求解低碳FJSP一种具有较强竞争力的优化方法.

6 结论(Conclusions)

柔性作业车间环境下低碳调度问题研究具有非常重要的现实意义. 本文针对低碳FJSP, 在对其3个子问题单独编码后, 运用一种新的蛙跳算法SFLA对问题进行求解. 该算法加入了记忆, 并在种群划分和组内搜索阶段采用全新的方式和策略以提高解的质量. 最后, 通过实验验证了SFLA算法的有效性和可行性.

利用SFLA研究低碳FJSP可能是一个很好的思路. 为了满足严格的节能环保要求, 作者仍将关注各种环境下的低碳车间调度问题. 同时作者将继续深入研究SFLA以及其他群体智能算法的改进或混合策略, 为所研究问题提供更好的算法理论支持.

参考文献(References):

[1] LIU G S, ZHANG B X, YANG D D, et al. A branch-and-bound algorithm for minimizing the energy consumption in the PFS problem [J]. *Mathematical Problems in Engineering*, 2013, 2013(2): 388 – 400.

[2] DAI M, TANG D B, GIRET A, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm [J]. *Robotics and Computer-Integrated Manufacturing*, 2013, 29(2): 418 – 429.

[3] LUO H, DU B, HUANG G Q, et al. Hybrid flow shop scheduling considering machine electricity consumption cost [J]. *International Journal of Production Economics*, 2013, 146(2): 423 – 439.

[4] LIN W W, YU D Y, ZHANG C Y, et al. A multi-objective teaching-learning-based optimization algorithm to schedu-ling in turning process for minimizing makespan and carbon footprint [J]. *Journal of Cleaner Production*, 2015, 101: 337 – 347

[5] NAGASAWA K, IKEDA Y, IROHARA T. Robust flow shop scheduling with random processing times for reduction of peak power consumption [J]. *Simulation Modeling Practice & Theory*, 2015, 59(1): 102 – 113.

[6] MANSOURI S A, AKTAS E, BESIKC U. Green scheduling of a two-machine flowshop: trade-off between makespan and energy consumption [J]. *European Journal of Opera-tional Research*, 2016, 248(3): 772 – 788.

[7] DING J Y, SONG S J, WU C. Carbon-efficient scheduling of flow shops by multi-objective optimization [J]. *European Journal of Operational Research*, 2016, 248(3): 758 – 771.

[8] FANG K, UHAN N, ZHAO F, et al. A new approach to scheduling in manufacturing for power consumption an carbon footprint reduction [J]. *Journal of Manufacturing Systems*, 2011, 30(2): 234 – 240

[9] TANG D B, DAI M. Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem [J]. *Chinese Journal of Mechan-ical Engineering*, 2015, 28(5): 1 – 8.

[10] LIU Y, TIWARI A. An investigation into minimizing total energy consumption and total completion time in a flexible job shop for recycling carbon fiber reinforced polymer [C] *//The 22nd CIRP conference on Life Cycle Engineering*. [s.l.]: [s.n.], 2015, 29: 722 – 727.

[11] HE Y, LI Y, WU T, et al. An energy-responsive optimiza-tion method for machine tool selection and operation sequence in flexible machining job shops [J]. *Journal of Cleaner Production*. 2015, 87(3): 245 – 254.

[12] LEI D M, GUO X P. An effective neighborhood search for scheduling in dual resource constrained interval job shop with environmental objective [J]. *International Journal of Production Economics*, 2015, 159(1): 296 – 303.

[13] ZHANG R, CHIONG R. Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption [J]. *Journal of Clearner Production*, 2016, 112(4): 3361 – 3375

[14] JIANG Zengqiang, ZUO Le. Multi-objective flexible job-shop scheduling based on low-carbon strategy [J]. *Computer Intergrated Manufacturing Systems*, 2015, 21(4): 1023 – 1031.
(蒋增强, 左乐. 低碳策略下的多目标柔性作业车间调度 [J]. 计算机集成制造系统, 2015, 21(4): 1023 – 1031.)

[15] TANG Lili. Improved migrating birds optimization algo-rithm to solve low-carbon scheduling problem [J]. *Computer Engineering and Applications*, 2016, 52(17): 166 – 171.
(唐立力. 求解低碳调度问题的改进型候鸟优化算法 [J]. 计算机工程与应用, 2016, 52(17): 166 – 171.)

[16] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using the shuffled frog leaping algorithm [J]. *Journal of Water Resources Planning and Management*, 2003, 129(3): 210 – 225

[17] PAN Yuxiang, PAN Quanke, SANG Hongyan. Hybrid discrete shuffled frogleaping algorithm for lot-streaming flowshop schedul ing problem [J]. *Computer Intergrated Manufacturing Systems*, 2010, 16(6): 1265 – 1271.
(潘玉霞, 潘全科, 桑红燕. 批量流水车间调度问题的混合离散蛙跳算法 [J]. 计算机集成制造系统, 2010, 16(6): 1265 – 1271.)

[18] PAN Q K, WANG L, GAO L, et al. An effective shuffled frog leaping algorithm for lot-streaming flow shop scheduling problem [J]. *International Journal of Advanced Manufacturing Technology*, 2011, 52(5/6/7/8): 699 – 713.

[19] LI Jianjun, YU Bin, CHEN Wuping. Improvement and simulation for shuffled frog leaping algorithm [J]. *Journal of System Simulation*, 2014, 26(4): 755 – 760.
(李建军, 郁滨, 陈武平. 混合蛙跳算法的改进与仿真 [J]. 系统仿真学报, 2014, 26(4): 755 – 760.)

[20] LEI D M, GUO X P. A shuffled frog-leaping algorithm for hybrid flow scheduling with two agents [J]. *Expert Systems with Applications*, 2015, 42(23): 9333 – 9339.

[21] LEI D M, GUO X P. A shuffled frog-leaping algorithm for job shop scheduling with outsourcing options [J]. *International Journal of Production Research*, 2016, 54(16): 4793 – 4804.

[22] ZHAO Min, ZHANG Weiguo, YU Lizhong. Carbon emissions from energy consumption in shanghai city [J]. *Research of Environmental Sciences*, 2009, 22(8): 984 – 989.
(赵敏, 张卫国, 俞立中. 上海市能源消费碳排放分析 [J]. 环境科学研究, 2009, 22(8): 984 – 989.)

[23] BRANDIMARTE P. Routing and scheduling in a flexible jobshop by tabu search [J]. *Annals of Operations Research*, 1993, 41(3): 157 – 183.

[24] DAUZÈRE-PÉRÉS S, PAULLI J. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search [J]. *Annals of Operations Research*, 1997, 70(1): 281 – 306.

[25] XU Y, WANG L, WANG S Y, et al. An effective teaching-learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time [J]. *Neurocomputing*, 2015, 148(1): 260 – 268.

[26] LI X Y, GAO L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem [J]. *International Journal of Production Economics*, 2016, 174: 93 – 110.

作者简介:

艾子义 (1993–), 女, 硕士研究生, 研究领域为智能系统优化与控制, E-mail: aiziyi1026@163.com;

雷德明 (1968–), 男, 博士, 教授, 博士生导师, 研究领域为智能系统优化与控制, E-mail: deminglei11@163.com.