

柔性车间调度的解空间距离聚类和变邻域搜索粒子群算法^①

杜兆龙, 徐玉斌, 崔志华, 李建伟, 赵俊忠

(太原科技大学 计算机科学与技术学院, 太原 030024)

摘要: 根据柔性车间调度问题提出基于解空间距离聚类和变邻域搜索的粒子群算法. 在粒子群算法基础上采用贪婪策略引入变邻域搜索方式, 即调整关键路径上最大关键工序的机器位置, 调整关键路径上工序相对位置变化, 加强局部搜索能力; 根据机器加工工序的空间距离, 采用 K-means 聚类得到机器加工工序“优良个体”, 加大局部搜索性能. 同时对于粒子群算法速度更新采用局部停滞策略, 保留局部片段相对位置不变特性. 通过实验仿真, 优化算法取得了较好的效果, 与一般的粒子群算法相比较收敛速度迅速且性能良好.

关键词: 柔性车间调度; 变邻域搜索; 空间距离 K-means 聚类; 速度局部停滞

Solution Space Distance Clustering-Variable Neighborhood Search Particle Swarm Optimization for Flexible Job Shop Scheduling Problem

DU Zhao-Long, XU Yu-Bin, CUI Zhi-Hua, LI Jian-Wei, ZHAO Jun-Zhong

(Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: Based on the Flexible Job-Shop Scheduling Problem(FJSP), an improved particle swarm optimization algorithm is proposed, which is based on solution space distance clustering and variable neighborhood search. In this algorithm, a better solution to the problem is that the greedy strategy is adopted to introduce a variable neighborhood search method, adjusting machine location of the biggest key processes on the critical path, adjusting the relative position changes which is on the critical path. According to the space distance of the machining process, the K-means clustering is used to get the “excellent individuals” of machine processing, increasing the local search performance. At the same time, the speed of the particle swarm optimization is updated with the local self-adaptive stagnation strategy, and the relative position of the local segment could be kept unchanged. Through the experimental simulation, the optimization algorithm achieves good effectiveness, and the convergence speed is rapider and the performance is better compared with the general PSO algorithm.

Key words: flexible job shop scheduling problem; variable neighborhood search; space distance K-means clustering; velocity local stagnation

柔性车间调度是车间生产中的一类特殊形式. 所谓的柔性车间调度是指 m 台机器加工 n 件工件, 每个工件每个工序可以在多台机器上加工完成, 最终得到一种调度安排使得 n 种工件加工完成的时间最小的方案^[1,2]. 该类问题因其随着工件和机器规模的扩大, 其求解规模伴随有爆炸性特点. 研究已经证实柔性车间调度问题是一类非确定多项式 NP 难问题^[3]. 因此用直

接查找算法很难获得较优的答案; 然而伴随着启发式搜索算法和群智能算法的发展, 则出现了越来越多的近似求解算法.

针对柔性车间的调度问题, 已有研究主要有: 遗传算法、粒子群算法、蜂群算法、蚁群算法、免疫算法等. 为了增强求解问题的能力, 相关学者进行了一系列研究. 曾建潮、崔志华等^[4]提出一种基于主动调度

① 收稿时间:2016-03-14;收到修改稿时间:2016-04-29 [doi:10.15888/j.cnki.csa.005482]

编码的遗传算法,实现将所有工序以不可重复的自然数进行编码,优点是编码唯一且可行.孔飞^[5]提出的基于双层粒子群优化算法 ITLPSO,以工件号为工序编码,相同工件的工序编码相同,同时采用惯性权重 w 凸凹函数变化及其停滞阻滞策略,在一定程度上加速收敛性能和搜索规模.董蓉等^[6]则提出一种以析取图模型为基础的混合 GA-CAO 算法,在遗传算法结果基础上利用蚁群算法寻得较优的解.徐华等^[7]提出的 DPSO 算法使用基于 NEH 的最短用时分解策略避免早熟和提高全局搜索能力.谢志强等^[8]提出的动态关键路径策略和短用时调度策略实现以纵向为主兼顾横向的双向调度优化.赵诗奎^[9]提出基于空闲时间的邻域结构给出最大限度查询关键工序相关机器空闲时间的方法,移动关键工序到空闲时间位置实现搜索优化.张国辉^[10]对 VNS-变邻域搜索进行了总结,强调利用某种算法为 VNS 提供较好质量的初始解集.高永超^[11]参考的 Big Valley 分布理论和骨架尺寸的研究与讨论,其中 Boose 推断 TSP 等类似问题的最优解分布于一个大谷内.孙元凯等^[12]提出的解空间距离矩阵、衡量解空间规模的特征数和分隔因子、决定局部解空间崎岖形状的加工时间分布形式,给出问题难易程度的衡量标准.已有研究表明,单一算法在全局或者局部搜索过程中都存在一定问题.同时对于柔性车间调度问题本身特性的研究也有助于算法性能的提高.

由于柔性车间调度问题复杂性,本文采用一种基于解空间距离聚类和变邻域搜索策略的 PSO 算法,附加速度停滞控制,在调度过程中不断优化机器工序位置,以得到近似最优解.

1 问题描述

柔性车间调度问题,是指具有 m 台机器,加工 n 件工件的调度优化问题.加工工件的机器集合为 $M = \{m_1, m_2, m_3, \dots, m_m\}$, m_i 为第 i 台机器, $i = 1, 2, 3, \dots, m$. 工件的集合 $P = \{p_1, p_2, p_3, \dots, p_n\}$, 为第 j 件工件, $j = 1, 2, 3, \dots, n$. 每个工件的工序数和加工次序都是确定的,对应于每个工件的工序集合,如工件 p_j 对应的工序数为 n_j , 故对应的工序集 $op_j = \{op_{j1}, op_{j2}, op_{j3}, \dots, op_{jn_j}\}$, 而所有工件对应的工序数集合 $OP = \{n_1, n_2, n_3, \dots, n_n\}$, n_j 对应第 j 个工件 p_j 的总工序数. 工件 p_j 的工序集 op_j 在机器集上加

工时间矩阵为 t_j , 大小为 $n_j * m$ 的矩阵. 如果工序不能在机器上加工,设置相应的时间设置为 0 标记. 时间矩阵集合 $Time = \{t_1; t_2; t_3; \dots; t_n\}$. 目标函数为最后一个被加工完的工件时间的最小值, 即:

$$makespan = \min(C_{\max}), C_{\max} = \max_{1 \leq j \leq n} \{C_j\}$$

其中 C_j 为第 j 个工件加工完成时的时间.

同时在建模过程中假设:

- 1) 同工件同时刻至多一道工序在某台机器加工;
- 2) 同时刻某台机器至多加工一工件的一道工序;
- 3) 工件开始加工后必须完成该工件所有工序;
- 4) 工件加工按照确定的加工工序次序进行;
- 5) 每一工序只能在一台机器上加工,不能选择多台机器.

2 基于解空间和邻域搜索策略粒子群算法

2.1 微粒群算法

Kennedy 和 Eberhart 共同提出的微粒群算法^[13],主要是针对于鸟类群体行为的觅食或者飞向栖息地的过程的建模.在飞行进化过程中主要受个体速度惯性部分、个体认知部分、群体社会认知三部分影响.标准的微粒群算法包括速度和位移两个进化公式:

$$v_j(t+1) = w * v_j(t) + c_1 r_1 (p_j(t) - x_j(t)) + c_2 r_2 (p_g(t) - x_j(t)) \quad (1)$$

$$x_j(t+1) = x_j(t) + v_j(t+1) \quad (2)$$

$v_j(t+1)$ 为第 $(t+1)$ 代第 j 个微粒的速度, $x_j(t+1)$ 为第 $(t+1)$ 代第 j 个微粒的位置, w 为相应的惯性权重, c_1 为个体认知系数, c_2 为社会系数, r_1, r_2 为服从均匀分布的随机数; $p_j(t)$ 为第 t 代第 j 个微粒的个体历史最优位置, $p_g(t)$ 为第 t 代的群体历史最优位置.

针对车间生产调度,首先是工序和机器的编码问题.机器按照机器顺序编码.工序编码是按照某工件当前出现次数表示该工件的工序数的编码方式.如工件数 2、机器数 3,每个工件都包含 3 道工序,下面矩阵中第一行表示工序排列,第二行表示对应工序加工所在机器排列.矩阵第一列中工件号为 1,且第一次出现,代表工件 1 的第一道工序,机器号为 2;第四列中工件号也为 1,但是工件号 1 到当前位置已出现 3 次,故代表工件 1 的第三道工序,机器号为 1.

$$\begin{bmatrix} process \\ machine \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 2 \\ 2 & 3 & 2 & 1 & 3 & 1 \end{bmatrix}$$

2.2 速度局部停滞控制

由 Big Valley 分布理论和骨架尺寸的研究与讨论, Boose 推断 TSP 类似问题的最优解可能分布于一个大谷内, 因此对于调度方面问题也可以考虑对于局部进行特殊搜索.

定义 1. 种群中个体对于本次进化, 其中某段或者某几片段的进化速度强制控制为 0, 本次操作完成后, 在下次的操作中再以极小概率选择部分片段再次进行强制控制, 这种控制称为速度局部停滞控制.

未选择控制部分, 其工序位置可以自由交换. 如果所有位置自由交换都出现障碍, 对于搜索则会陷入停滞. 用稀疏矩阵 $s(i,:)$ 和概率均匀分布矩阵 $u(i,:)$, 共同对于工序变化速度进行选择, 当前第 i 个个体的速度停滞控制公式为:

$$v'(p_{i,:}) = s(i,:) * u(i,:) * v(p_{i,:}) \quad (3)$$

假设之前的速度和位置分别为

$$v(p_{i,:}) = [0.12 \quad 0.32 \quad 0.19 \quad -0.14 \quad 0.59 \quad -0.84]$$

$$p_i = [1 \quad 2 \quad 1 \quad 1 \quad 2 \quad 2]$$

变化后的速度为

$$v'(p_{i,:}) = [0 \quad 0.16 \quad 0 \quad 0 \quad 0.531 \quad 0]$$

而新的位置更新得到 $p' = [1 \quad 2.16 \quad 1 \quad 1 \quad 2.531 \quad 2]$, 对于重新排序的到 p' , 第一行为位置数值, 第二行为排序序号

$$\begin{bmatrix} 1 & 1 & 1 & 2 & 2.16 & 2.531 \\ 1 & 3 & 4 & 6 & 2 & 5 \end{bmatrix}$$

按照新位置的大小重新排列得到 $[1 \quad 1 \quad 1 \quad 2 \quad 2]$, 同时修正对应的机器.

定义 2. 由工件号重复多次出现的编码方式造成对工件工序排列时, 相同或者相邻工件的工序容易凑到一起“扎堆”, 较远的工件号与其交流出现一定障碍, 需要归一化操作, 以消除由于编码产生的工件工序“扎堆”问题, 这一操作称为工序编码归一化.

如果工件 1 各个工序的编码都为 1, 工件 8 各个工序编码都为 8, 这时假设工件 8 的工序变化为 $process_8 = 8 + v$, 与工件号为 1, $process_1 = 1 + v_1$, 如果对得到的工序位置进行排序的话, 其位置交流可能就会出现障碍, 导致工件编号越小其工序越容易出现现在调度的前面, 工件编号越大其工序越容易出现后面. 为了解决这个问题, 在进行粒子群算法位置更

新前对当前工件工序排列进行归一化, 从而消除掉工件号编码的问题. 这里仅仅对工序编号进行归一化, 而不对工序变化速度进行处理, 可以保留各个工件工序变化的特殊性, 而且速度在一定程度上是个体惯性、个体和群体三者的影响结合体. 工序编码归一化函数:

$$EFP = v * \cos\left(\frac{2\pi * p}{jobN} + \frac{\pi^2}{2 * jobN}\right) \quad (4)$$

其中 p 为工序编码矩阵, $jobN$ 为工件数, v 为工序的速度矩阵, EFP 为归一化后的工序.

2.3 解空间 K-means 聚类

定义 3. 如果将每个解都视为高维空间中的一个点, 并且在高维空间中对于这些多维向量排列, 每个解都会与部分解或无效解相邻, 由此构成的高维向量集合, 或者说是高维点阵集合, 称为问题的原空间^[12].

定义 4. 原空间中由可行解构成的集合称之为问题的解空间^[12].

定义 5. 对于车间生产调度问题而言, 取任意解 s 和 t , 定义两解之间的距离^[12]:

$$H(s,t) = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=j+1}^n XOR(PREC_{i,s}(j,k), PREC_{i,t}(j,k)) \quad (5)$$

其中 $PREC_{i,s}(j,k)$ 表示解 s 中第 i 台机器上工序 j 和工序 k 之间的加工顺序, 如果 j 在 k 的前面则 $PREC_{i,s}(j,k) = 0$, 否则 $PREC_{i,s}(j,k) = 1$; XOR 为逻辑异或运算符. 有 $n = \sum_{j=1}^n n_j$ 是所有工件工序数之和, 在计算 $H(s,t)$ 的时候采用的是工序的自然数编码.

公式(5)衡量两个解之间距离, 作为种群中某个解与当前最优解的距离计算公式. 通过研究发现当出现新的最优解时当前种群所有个体与最优解的距离的平均值可能会发生较大的波动, 之后又会趋于稳定直到下次出现新的最优值.

对于如何衡量单个个体的性能, 以种群中某个解在所有机器上工序的逆序数和的形式表示, 表示当前解的所有机器上的逆序数的总和, 如下

$$RN = \sum_i \sum_{j,k} p_i(j,k) \quad (6)$$

在机器 i 上, $p_i(j,k)$ 为工序 j,k 逆序数表示, 如果工序 j 在 k 的前面, 那么 $p_i(j,k) = 0$; 如果工序 j 在 k 的后面, 则 $p_i(j,k) = 1$.

定义 6. 不同工件的工序, 如工件 1 的第 2 道工序, 和工件 2,3,4,5 等的第 2 道工序, 可以互称为顺位工序.

如果同一工件的工序, 在同一个机器上加工, 因

为加工顺序问题不会存在逆序数;但是不同工件的工序,有可能产生逆序排列.采用每台机器上工序逆序数的分析,可以确认当前机器上不同工件间相同顺位工序被加工的次序,如果相同顺位工序排列的很紧凑、位置很靠近,则应该是一种较好的安排.

通过计算当前解在每台机器上工序的逆序数,然后进行聚类分析可以衡量当前解的工件加工的工序对于寻找最优过程的影响.

2.4 变邻域搜索策略

基于变邻域搜索的邻域动作是由关键路径决定的,目前有 Nowicki 和 Smutnicki 提出的 N5 结构; Balas 和 Vazacopoulos 提出的 N6 结构,张超勇提出的 N7 结构.在搜索可行解的基础上采用机器调整和关键工序位置移动策略.首先利用整个工序的排列顺序,搜寻到相应的关键路径.关键路径可能有多条,仅选择其中的一条进行调整.其次从关键路径中找到相应的最大花费工序,简称最大关键工序.每次进行最大关键工序的机器位置调整,同时在关键路径上进行工序位置的移动.

机器策略:

(1) 对于最大关键工序进行机器调整,选择该工序花费尽可能小的机器进行调整;

(2) 关键路径中工件的首道工序没在初始时刻开始加工的调整到某台机器的近似开始时刻;

(3) 关键路径中最后一道工序,如果存在可以调整机器使得调整后整体时间缩短则调整.

关键路径上工序调整:关键路径上两道工序为 v_w, \dots, v_i, v_i 可以调整到 v_w 之前,除了该处位置的调整,其它工序的相对位置不受影响,但关键路径的时间被缩短,只要满足以下条件之一:

- (1) 如果 v_i 为该道工件的首道工序,可以直接移动;
- (2) 如果 v_i 的前道工序 v_{i-1} ,其完成的时间要比 v_w 工序的完成时间早.

3 基于解空间和变邻域搜索的粒子群算法

3.1 初始化 PSO 种群和 W 的变化策略

初始化 PSO 种群,个体数为 N,维数为所有工件的工序数之和,种群个体随机产生. PSO 惯性参数 w 采用凸凹函数形式:

$$w = (a - b) \left(\frac{n}{N}\right)^2 + 2(b - a) \frac{n}{N} + b, w \in [a, b] \quad (7)$$

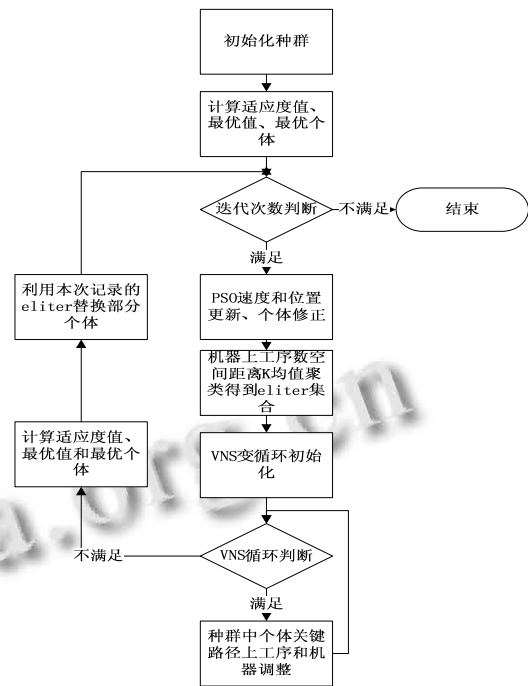


图1 算法流程图

3.2 粒子群速度和位置更新

对于机器更新,假设之前机器位置如下所示 $m(t) = [2 \ 3 \ 2 \ 1 \ 3 \ 1]$,按照之前介绍的粒子群进化公式,可以得到 $m'(t+1) = [3.5 \ 2.4 \ 2.2 \ 0.8 \ 2.4 \ 2.1]$ 向上取整则 $m'(t+1) = [4 \ 3 \ 3 \ 1 \ 3 \ 2]$,4超出了机器编码的范围,可以修正为编号为1,则更新后的新的机器编码为 $m(t+1) = [1 \ 3 \ 3 \ 1 \ 3 \ 2]$.

稀疏矩阵 $s(i,:)$ 和概率均匀分布矩阵 $u(i,:)$; 粒子群速度 $v(p_{i,:})$, 位置 p_i 等数值如下:

$$p_i = [1 \ 2 \ 1 \ 1 \ 2 \ 2]$$

$$s(i,:) = [1 \ 0 \ 1 \ 0 \ 1 \ 0]$$

$$u(i,:) = [-0.3 \ 0.5 \ 0.4 \ 0.3 \ 0.9 \ -0.2]$$

$$v(p_{i,:}) = [0.12 \ 0.32 \ 0.19 \ -0.14 \ 0.59 \ -0.84]$$

根据公式(3)当前第 i 个个体的速度更新为

$$v'(p_{i,:}) = [-0.036 \ 0 \ 0.076 \ 0 \ 0.531 \ 0]$$

根据公式(4)归一化工序编码得 EFp 如下

$$EFp = [-0.0280 \ 0 \ 0.0592 \ 0 \ -0.4129 \ 0]$$

EFp 代替 p_i , 由公式(2)更新工序位置得

$$p_i' = [-0.064 \ 0 \ 0.1352 \ 0 \ 0.1181 \ 0]$$

对得到的工序位置进行升序排列

$$\begin{bmatrix} -0.064 & 0 & 0 & 0 & 0.1181 & 0.1352 \\ 1 & 2 & 4 & 6 & 5 & 3 \end{bmatrix}$$

按照新位置顺序重新排列工序为 $[1 \ 2 \ 1 \ 2 \ 2 \ 1]$,同时修正对应的机器.

3.3 空间距离 K 均值聚类

根据空间距离计算的逆序数、机器加工工序数进行聚类。逆序数则是每台机器上工序的逆序数，种群中每个个体的逆序数为一个向量，整个种群则构成一个逆序数矩阵。机器加工工序数也是种群中每个个体由每台机器上加工工序数组合成一个向量。采用 K-means 聚类算法，聚类结果数位 N1，迭代次数为 Niter，去掉重复个体。通过聚类可以将优良的个体替代掉“年老”个体，将优良个体保持至下一代进行进化，提高优良个体的局部搜索范围。

3.4 VNS 变邻域搜索

初始化 VNS 搜索的最大次数，函数为

$$F(n) = N * (1 - e^{-\frac{n}{N}}) \quad (8)$$

其中 n 为当次循环，N 为循环最大次数。

在循环过程中首先搜索每个个体一条关键路径。在求解关键路径的同时，将关键路径上最大关键工序查找出来。将关键路径上最大关键工序进行相应调整，每次只调整(1)(2)(3)中的一种情况；然后再根据关键路径上前后工序位置进行移动；至此完成一次工序和机器的内部调整，接着开始下次 VNS 循环。

4 实验分析

4.1 实验设计

为了验证实验效果，采用 Intel 双核处理器 I5-5200U@2.20GHZ，内存 4GB，64 为操作系统，MATLAB 为实验仿真工具，采用数据集为 Kacem、Hurink^[14]、Brandimarte^[14]等。

算法的参数设置：种群中个体数 N=60，惯性参数 W 的范围[0.1,0.9]，c1=1.5,c2=1.5,r1,r2 采用范围在(0,1)之间的随机数，惯性权重更新采用上文介绍的抛物线形式，整体迭代次数 N=100，当前迭代次数为 n，因此每次 VNS 邻域搜索的次数最大值设置 count <= 20 限制。

4.2 仿真结果

基于速度停滞控制、空间距离聚类及变邻域搜索的粒子群算法简称为 SVNS-vPSO。对于 Kacem 基准问题的 10*10 样例，其迭代过程中的 H(s,t)距离均值变化曲线如图 2 所示，在当次种群迭代中寻得较优值时会发生 H(s,t)均值跳动反应。因此搜索的结果极可能存在于这些跳动位置处种群局部最优个体身上。

对于 Kacem 基准问题的 8*8 算例，SVNS-vPSO 算法生成的最优调度甘特图如图 3。图 4 为 Kacem 问题的本算法收敛曲线。表 1 为本算法下 Kacem 各个实例的具体运行数据。对于图 4 和表 1 表明算法能够在

较短时间与迭代次数内获得最优解。

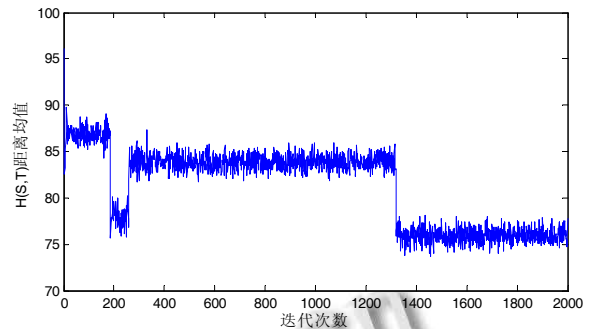


图 2 Kacem 10*10 的 H(s,t)均值变化曲线

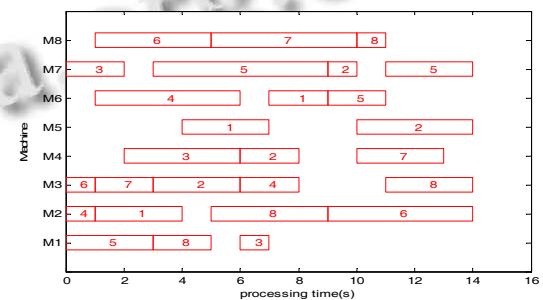


图 3 Kacem 8*8 调度优化甘特图

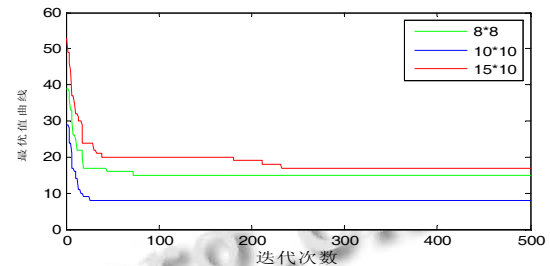


图 4 Kacem 数据的算法收敛曲线

表 2 中对于 PSO 算法、ITLPSO 双层粒子群算法^[5]、AL+CGA(局部最小化分配模型的进化算法)、改进 GA、GA-ACO 遗传蚁群算法，其中后三种算法参数设置按董蓉文中^[6]参数形式，最后搜索的结果如表 2，由数据可见 SVNS-vPSO 算法的搜索最优和平均值较一般的粒子群算法较好，但是在平均计算时间和平均结果上和 AL+CGA、改进 GA 算法比较而言相似甚至有些数据优于它们，但是较 GA-ACO 算法稍差些。

表 1 Kacem 各实例的计算结果统计

实例	工件数	机器数	工序数	平均计算时间/s	最优解	平均解
1	8	8	27	37.5895	14	15.996
2	10	10	30	137.3046	7	7.833
3	15	10	56	256.2888	15	17.533

表 2 Kacem 基准问题各方法运行 30 次结果比较

方法	8*8		10*10		15*10	
	最优	平均	最优	平均	最优	平均
本文方法	14	15.996	7	7.833	15	17.533
PSO	27	-	20	-	35	-
ITLPSO	25	-	19	-	33	-
AL+CGA	15	-	7	-	23	-
改进 GA	14	16.2	7	7.8	-	-
GA-ACO	14	15.3	7	7.6	11	12.7

表 3 和表 4 中数据集来自于参考引用^[14]网页中的 FJSPLIB 文件. 这里选用 Hurink 数据集和 Brandimarte 数据集中数据进行测试. 表中 n 代表工件数, m 代表机器数, C**表示当前的最优值, Cmin 为本文算法获得的最优值, RD(%)为本文算法最优值与 C**的相对误差, 其具体公式为:

$$RD(\%) = \frac{C_{\min} - C^{**}}{C_{\min}} * 100 \quad (9)$$

表 3 Hurink 数据集^[14]运行结果

问题	n,m	总工序数	C**	SVNS-vPSO	
				Cmin	RD(%)
LA01	10,5	50	666	666	0
LA06	15,5	75	926	926	0
FT06	6,6	36	55	56	1.79
FT10	10,10	100	930	1079	13.80
LA11	20,5	100	1222	1230	0.65
LA16	10,10	100	945	1045	9.57
LA21	15,10	150	1046	1251	16.39
LA26	20,10	200	1218	1498	18.69

表 4 Brandimarte 数据集^[14]运行结果

问题	n,m	总工序数	C**	SVNS-vPSO	
				Cmin	RD(%)
MK01	10,6	55	37	48	22.92
MK02	10,6	58	26	39	33.33
MK03	15,8	150	204	252	19.05
MK04	15,8	90	60	85	29.41
MK05	15,4	106	173	214	19.16
MK08	20,10	225	523	615	14.96

由表 3 和表 4 测试结果表明, 该算法取得一定效果. 对于 Hurink 数据, 当工序数达到一定程度后, 相对误差开始增大; 对于 Brandimarte 类型数据, 其相对误差值高些.

实验仿真结果表明, 当工序数少时如 Kacem 问题, 可以快速搜索到有效的结果; 当工序数增大后其效果将会出现一定偏差, 如 Hurink 和 Brandimarte 的数据. 算法中对工序和机器分别编码, 然后进行 PSO 算法速度和位置更新, 但进化过程中两者只有一个约束关系, 即工序不能够在此机器上加工. 也就是说工序和机器的进化两者近似于彼此独立, 缺少进化的有效协同,

无法形成统一整体. 对工序和机器进行整体编码的方式, 可以解决类似的问题, 但其设计方法则更困难.

5 结语

本文对于柔性车间调度问题提出了 SVNS-vPSO 算法. 即通过解空间距离 K 均值聚类、贪婪变邻域搜索、粒子群停滞控制等策略提高了优化算法的收敛速度和搜索结果. 通过研究发现, 工序与机器之间彼此近似孤立进化, 将影响整体的搜索结果. 下一步对工序和机器进化进行协同研究, 或者是采用工序和机器整体编码的方式^[15]研究. 同时对于机器使用率和整体加工时间等多目标优化问题可以通过基于空间距离、机器逆序数等因素进行优化研究.

参考文献

- 1 Brucker P, Schlie R. Job-Shop scheduling with multipurpose machines. Computing, 1990, 45(4): 369-375.
- 2 Mati Y, Rezg N, Xie XL. An intergrated greedy heuristic for a flexible job shop scheduling problem. Proc. of IEEE International Conference on Systems, Man, and Cyberneticcs. Washington, D.C., USA. IEEE. 2001. 2534-2539.
- 3 Blazewica J, Finke G, Haaopt G, et al. New trends in machine scheduling. European Journal of Operational Research, 1988, 37(3): 303-317.
- 4 曾建潮, 崔志华, 等. 自然计算. 长沙: 国防工业出版社, 2012.
- 5 孔飞, 吴定会, 等. 基于双层粒子群优化算法的柔性作业车间调度优化. 计算机应用, 2015, 35(2): 476-480.
- 6 董蓉, 何卫平. 求解 FJSP 的混合遗传-蚁群算法. 计算机集成制造系统 2012, 18(11): 2492-2501.
- 7 徐华, 张庭. 改进离散粒子群算法求解柔性流水车间调度问题. 计算机应用, 2015, 35(5): 1342-1347, 1352.
- 8 谢志强, 杨静等. 基于工序集的动态关键路径多产品制造调度算法. 计算机学报, 2011, 34(2): 406-412.
- 9 赵诗奎, 方水良, 等. 作业车间调度的空闲时间邻域搜索遗传算法. 计算机集成制造系统, 2014, 20(8): 1930-1940.
- 10 张国辉. 柔性作业车间调度方法研究[博士学位论文]. 上海: 华中科技大学, 2009.
- 11 高永超. 智能优化算法的性能及搜索空间研究[博士学位论文]. 济南: 山东大学, 2007.
- 12 孙元凯, 刘民, 等. 调度问题及其解空间的特征分析. 电子学报, 2001, 29(8): 1042-1045.
- 13 Eberhart R, Kennedy J. New optimizer using particle swarm theory. Proc. of the Sixth International Symposium on Micro Machine and Human Science, MHS'95. IEEE, Piscataway, NJ, USA. 1995. 39
- 14 <http://people.idsia.ch/~monaldo/fjsp.html>.
- 15 Singh MR, Mahapatra SS. A quantum hehaved particle swarm optimization for flexible job shop scheduling. Computer & Industrial Engineering. 2016, 93: 36-44.