

6-4-2020

Double Bare Bones Particle Swarm Algorithm for Solving Flexible Job-shop Scheduling Problem

Yueming Dai

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Minghui Wang

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Wang Chun

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Wang Yan

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Double Bare Bones Particle Swarm Algorithm for Solving Flexible Job-shop Scheduling Problem

Abstract

Abstract: To solve flexible job shop scheduling problem, a double bare bones particle swarm algorithm (DBBPSO) was proposed to minimize the makespan. Combining *Von-neumann bare bones particle swarm optimization algorithm* and chaotic mutation bare bones particle swarm optimization algorithm, *the algorithm DBBPSO used a communication method to cooperate evolution*. This approach could keep balance between global exploration and local exploration, and *a machine selection method was proposed based on the tenet of minimizing makespan*. The proposed algorithm was compared with other algorithms on four benchmark problems and a scheduling optimization example. Simulation results indicate that the improved algorithm has the ability to obtain the optimal solution, and it is more suitable for solving the scheduling problem.

Keywords

flexible job-shop scheduling, double bare bones particle swarm optimization, cooperate evolution, machine selection

Recommended Citation

Dai Yueming, Wang Minghui, Wang Chun, Wang Yan. Double Bare Bones Particle Swarm Algorithm for Solving Flexible Job-shop Scheduling Problem[J]. Journal of System Simulation, 2017, 29(6): 1268-1276.

骨干双粒子群算法求解柔性作业车间调度问题

戴月明, 王明慧, 王春, 王艳

(江南大学教育部物联网技术应用工程研究中心, 江苏 无锡 214122)

摘要: 针对柔性作业车间调度问题, 以最小化最大完工时间为优化目标, 提出了一种骨干双粒子群算法(Double Bare Bones Particle Swarm Algorithm, DBBPSO)。算法结合基于冯诺依曼拓扑结构的改进骨干粒子群算法与基于混沌变异的骨干粒子群算法, 利用种群交流机制使两个种群协同进化, 实现了算法在全局搜索与局部开发之间的平衡, 并提出一种基于最小加工时间的机器选择策略。将所提算法在四个经典算例与一个柔性作业车间调度实例上与其他不同算法进行仿真对比, 结果表明所提算法比其他对比算法具有更好的寻优能力, 更适合解决该类调度问题。

关键词: 柔性作业车间调度; 骨干双粒子群算法; 协同进化; 机器选择策略

中图分类号: TP278

文献标识码: A

文章编号: 1004-731X (2017) 06-1268-09

DOI: 10.16182/j.issn1004731x.joss.201706015

Double Bare Bones Particle Swarm Algorithm for Solving Flexible Job-shop Scheduling Problem

Dai Yueming, Wang Minghui, Wang Chun, Wang Yan

(Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: To solve flexible job shop scheduling problem, a double bare bones particle swarm algorithm (DBBPSO) was proposed to minimize the makespan. Combining Von-neumann bare bones particle swarm optimization algorithm and chaotic mutation bare bones particle swarm optimization algorithm, the algorithm DBBPSO used a communication method to cooperate evolution. This approach could keep balance between global exploration and local exploration, and a machine selection method was proposed based on the tenet of minimizing makespan. The proposed algorithm was compared with other algorithms on four benchmark problems and a scheduling optimization example. Simulation results indicate that the improved algorithm has the ability to obtain the optimal solution, and it is more suitable for solving the scheduling problem.

Keywords: flexible job-shop scheduling; double bare bones particle swarm optimization; cooperate evolution; machine selection

引言

作业车间调度问题(Job-shop Scheduling

Problem, JSP)通过合理的安排工序加工顺序, 使一个或多个性能指标达到最优^[1]。在JSP的基础上, Bucker 等提出了柔性作业车间调度问题(Flexible Job-shop Scheduling Problem, FJSP)^[2]。与JSP不同, FJSP 中每个工件的每道工序可以在一台或多台机器上加工, 因此 FJSP 减少了机器约束, 增加了调度的灵活性, 更加贴近实际生产, 是比JSP更



收稿日期: 2016-08-22 修回日期: 2016-11-06;
基金项目: 国家自然科学基金(61572238), 江苏省杰出青年基金(BK20160001);
作者简介: 戴月明(通讯作者 1964-), 男, 江苏常熟, 硕士, 副教授, 硕导, 研究方向为人工智能和软件工程。

<http://www.china-simulation.com>

• 1268 •

难求解的 NP-hard 优化问题。

目前, 求解 FJSP 常用的方法有遗传算法、粒子群算法、人工蜂群算法以及蛙跳算法等群智能优化算法。作为最常见的一种群智能优化算法, 粒子群算法以其参数少、精度高、收敛快等特点被广泛应用于求解 FJSP 问题。文献[3]采用粒子群算法来解决 FJSP 问题, 其优化目标为最大完工时间最短; Laurent 等^[4]使用了一种带有随机局部搜索的混合粒子群算法来解决 FJSP 问题, 结果表明随机局部搜索显著提高了粒子群算法的性能; 文献[5]将离散粒子群算法与模拟退火算法相结合, 有效解决了粒子群算法早熟收敛问题; 周恺等^[6]在改进量子粒子群算法上, 利用反向学习策略解决 FJSP 问题。

骨干粒子群算法(Bare Bones Particle Swarm Optimization Algorithm, BBPSO)最早由 Kennedy^[7]在粒子群算法基础上提出。它精简了粒子群算法, 取消了粒子的速度属性, 以随机分布的方式更新粒子的位置。其简单的协作式的概率搜索方式能显著的提高粒子搜索效率和精度^[8]。由于算法简洁, 用于调节的参数较少且优化性能良好, BBPSO 算法被广泛应用于经济调度^[9]、故障诊断^[10]、工程优化^[11]等领域。

但是目前 BBPSO 算法主要应用于求解实际的连续优化问题, 其在解决组合优化问题包括 FJSP 问题上还没有相关的应用, 而且传统的 BBPSO 具有易早熟、易陷入局部最优等缺点。因此, 本文针对 FJSP 问题的求解, 基于粒子群交流机制融合两种改进的 BBPSO 算法, 提出一种骨干双粒子群优化算法(Double Bare Bones Particle Swarm Optimization Algorithm, DBBPSO)。所提算法利用基于冯诺依曼拓扑结构的改进骨干粒子群算法(Von Neumann Bare Bones Particle Swarm Optimization Algorithm, VBBPSO)与加入混沌变异机制的骨干粒子群算法(Chaotic Mutation Bare Bones Particle Swarm Optimization Algorithm, CBBPSO)这两个改进的 BBPSO 算法的各自特点对空间进行搜索, 通过引入粒子群交流机制, 使两个

算法在各自搜索过程中相互影响, 协同进化。目的是能让所提算法在求解 FJSP 问题上扩大搜索范围, 避免早熟现象的发生, 加快收敛速度, 提高寻优精度, 得到更好的调度方案。

1 问题描述

柔性作业车间调度问题描述为^[12]:

n 种不同的工件, 在 m 台机器上加工。每个工件有各自的固定加工工艺路线, 每个工艺有不同的可供选择的加工机器子集。不同的加工工序所选择的机器具有差异性, 相同工序加工机器选择的不同, 对应加工时间也会有差异。调度的目标是为每个工件的工序合理地分配加工机器, 并合理的安排各个机器上分配的工序加工顺序, 从而使最大完工时间最小。

对于柔性作业车间调度问题, 工件在加工过程中满足如下约束:

- (1) 同一时刻一台机器只能同时加工一道工序, 某道工序在同一时刻只能在一台机器上加工(资源约束);
- (2) 工件的某道工序一旦开始加工便不能随意中断, 直到该工序加工完成;
- (3) 所有工件具有相同的加工优先级, 它们被加工的概率相同;
- (4) 不同工件的各工序之间没有加工顺序约束, 同一工件的各工序之间有预先确定的加工顺序约束(顺序约束);
- (5) 所有工件在零时刻都可以被加工。

如果工件的每道工序都可在任意一台机器上加工, 则该调度问题为完全柔性作业车间调度问题(Total Flexible Job-shop Scheduling Problem, T-FJSP), 否则为部分柔性作业车间调度问题(Partial Flexible Job-shop Scheduling Problem, P-FJSP)^[13]。

1.1 参数定义

为了后续数学模型和相关知识的描述, 我们在

这里定义如下参数:

- m : 可供加工机器的总数;
 n : 待加工的工件总数;
 Na : 待加工的工序总数;
 $M = \{M_k | 1 \leq k \leq m\}$: 机器集;
 $J = \{J_i | 1 \leq i \leq n\}$: 工件集;
 n_i : 工件 J_i 的工序数;
 $O_i = \{O_{ij} | 1 \leq j \leq n_i\}$: 工件 J_i 的工序集;
 $M_{ij} = \{M_k | X_{ijk} = 1\}$: 工序 O_{ij} 的可用机器子集;
 T_{ijk} : 工序 O_{ij} 在机器 M_k 上的加工时间;
 S_{ijk} : 工序 O_{ij} 在机器 M_k 上开始加工时间;
 C_{ijk} : 工序 O_{ij} 在机器 M_k 上加工结束时间;
 C_j : 每个工件的完工时间;
 $X_{ijk} = \begin{cases} 1, & \text{工件 } O_{ij} \text{ 由设备 } M_k \text{ 加工} \\ 0, & \text{其他} \end{cases};$
 $R_{ijegk} = \begin{cases} 1, & O_{ij} \text{ 与 } O_{eg} \text{ 同在 } M_k \text{ 上加工, } O_{ij} \text{ 先加工} \\ 0, & \text{其他} \end{cases}。$

1.2 数学模型

FJSP 的数学模型及约束条件如下:

$$f = \min(\max_{1 \leq j \leq n} (C_j)) \quad (1)$$

$$\sum_{k \in M_{ij}} X_{ijk} = 1, \forall i, j \quad (2)$$

$$\begin{cases} C_{ijk} - C_{egk} \geq T_{egk} \\ R_{ijegk} = 1 \\ X_{ijk} = X_{egk} = 1 \end{cases} \quad (3)$$

$$\begin{cases} S_{ijk} - C_{i(j-1)h} \geq 0 \\ X_{ijk} = X_{i(j-1)h} = 1 \end{cases} \quad (4)$$

其中: 公式(1)为优化目标即最大完工时间最小, 公式(2)表示每个工序在加工过程中不能被打断, 公式(3)表示每个机器在同一时刻只能同时加工一道工序, 公式(4)表示工件的工序加工顺序约束。

2 骨干双粒子群优化算法

2.1 骨干粒子群算法

骨干粒子群是一种将粒子群算法精简的算法,

取消了粒子的速度属性, 以随机分布的方式更新粒子的位置。其一般模型可以归纳如下:

$$x_i = \mu(L_i) + \alpha \xi \delta(L_i) \quad (5)$$

式中: x_i 为粒子 i 的进化位置; μ 为搜索中心项, BBPSO 以 μ 为中心进行搜索, 越靠近 μ 的地方粒子落入的概率越大, 随着迭代的进行, 粒子群最终收敛于 μ ; δ 为离散控制项, 控制搜索域中粒子分布方差的大小, δ 越小, 粒子落入搜索中心 μ 附近的区域的概率越大, 一般情况随着迭代的进行, δ 将减小到 0; α 为位置更新公式中唯一的控制参数, α 越大, 粒子群分布越趋向于分散, α 越小, 越趋向于聚拢; ξ 为标准随机变量; L_i 为集合, 表示邻域拓扑结构, 对于局部环状拓扑, 有 $L_i = \{P_{i-1}, P_i, P_{i+1}\}$; 对于全局拓扑, 有 $L_i = \{P_1, P_2, \dots, P_N\}$, P_i 为粒子 i 的历史最优位置, N 为粒子数。

2.2 骨干双粒子群优化算法

2.2.1 基于冯诺依曼拓扑结构的骨干粒子群算法

粒子群之间通过不断交流相互影响从而寻找到最优解, 而拓扑结构则是影响粒子之间交流与算法收敛速度等性能的重要因素之一。冯诺依曼拓扑结构中每个粒子有上、下、左、右四个邻居构成网格形式^[14]。一个粒子找到的最优解只会影响周围的四个邻域粒子, 从而维持其他粒子的多样性, 进而避免粒子陷入早熟, 同时相对其他邻域结构, 收敛速度也得到了保证。

为了改善 BBPSO 易早熟, 易陷入局部最优解等缺点, VBBPSO 在引入冯诺依曼拓扑结构之后, 将 BBPSO 位置更新公式的搜索中心项和离散控制项也做了相应改进。为了使粒子的搜索中心项多样化, 跳出局部最优解, 赋予了搜索中心 μ 一个新的更新公式, 同时引入“兼顾落后粒子 P_{lw} ”这一概念, 扩大搜索范围至整个邻域。 μ_i 以 P_{best_i} 为中心, 沿着 P_{lw_i} 的方向进行适当的偏移, 这一偏移量增加了粒子搜索中心的多样性, 从而避免粒子陷入局部最优。搜索中心项公式具体描述如式(6):

$$\begin{cases} \mu_i = P_{lbest_i} + \beta \sigma_{\mu_i} \\ \sigma_{\mu_i} = P_{lw_i} - P_{lbest_i} \end{cases} \quad (6)$$

式中: P_{lbest_i} 为粒子 i 在邻域 $L(i)$ 中的邻域最优位置, 即邻域内各个粒子的适应度函数最小值对应的粒子的历史最优位置; β 为中心项调节系数; σ_{μ_i} 为搜索中心 μ_i 在邻域 $L(i)$ 中的偏差, P_{lw_i} 为粒子 i 的邻域 $L(i)$ 中适应度值最大的历史最优位置相对落后的粒子位置。

为避免算法陷入局部最优, 增强算法局部开发能力, 仿照 Rifaie 等^[15]在离散控制项中加入了一个系数 α 来控制离散项, 并使用局部最优位置与落后粒子位置的差值作为方差。改进后的离散控制项更新公式如下:

$$\delta_i = |P_{lbest_i} - P_{lw_i}| \quad (7)$$

VBBPSO 算法以冯诺依曼拓扑结构为邻域, 添加中心项调节系数, 引入“兼顾落后粒子”这一概念, 扩大搜索范围, 增加群体在邻域空间内寻优搜索能力, 改进算法中心项与离散项更新公式, 最终粒子位置更新公式为:

$$\begin{cases} \mathbf{x}_i = \mu(L_i) + \alpha \xi \delta(L_i); \\ \mu(L_i) = P_{lbest_i} + \beta (P_{lw_i} - P_{lbest_i}); \\ \delta(L_i) = |P_{lbest_i} - P_{lw_i}|; \\ \xi \sim N(0,1); \end{cases} \quad (8)$$

其中: ξ 满足标准高斯分布 $N(0,1)$ 。

2.2.2 基于混沌变异的骨干粒子群算法

在粒子群算法中, 随着迭代的增加, 粒子逐渐向个体历史最优点和全局最优点点靠拢, 种群的多样性逐渐下降, 算法容易陷入局部最优。为了增加种群的多样性, 避免算法陷入早熟, 引入混沌变异因子来跳出局部最优, 增加全局寻优的搜索范围。

混沌运动经常出现在非线性系统中^[16], 其能在一定范围内按照一定规律不重复地遍历所有状态, 且具有初始敏感性、伪随机性、遍历性等特点^[17]。在 CBBPSO 算法中选用描述生物种群演变规律的简化 Logistic 映射模型, 并用全局最优值的停滞迭代代数作为变异的触发条件, 当全局最优值

停滞迭代超过 5 代时, 采用下列公式更新全局最优值和历史最优值:

$$\begin{cases} a = 4 \times a \cdot (1 - a); \\ b = 4 \times b \cdot (1 - b); \\ P_i = P_i \cdot a \\ G = G \cdot b \end{cases} \quad (9)$$

式中: G 为全局最优位置, a 和 b 为初始值, 取值为随机数混沌变量。

CBBPSO 的位置更新公式为:

$$x_i = \frac{P_i + G}{2} + \frac{\xi(P_i - G)}{2} \quad (10)$$

式中: 进化中心项为粒子 i 的历史最优位置 P_i 与群体全局最优位置 G 的中点; 离散控制项为历史最优位置 P_i 与全局最优位置 G 的偏差。

2.2.3 粒子群交流机制

VBBPSO 与 CBBPSO 两个粒子群同时按照各自不同的轨迹进行寻优, 在每个单独的种群中, 粒子在个体历史最优与全局最优的共同作用下不断更新位置。两个单独的种群在每次迭代结束后进行交流, 引入粒子群交流机制, 将两个种群的全局最优点点相互比较, 选取适应度值最好的一个最优值作为两个种群共同的全局最优。粒子群交流机制能有效的将两个种群信息进行交流, 从而使两个种群相互引导, 协同进化。

2.2.4 骨干双粒子群优化算法步骤

本文所提 DBBPSO 流程如算法 1 所示:

算法 1: DBBPSO 具体步骤

步骤 1: 随机初始化 VBBPSO 与 CBBPSO 内粒子初始位置, 其中 VBBPSO 中为每个粒子构造拓扑结构;

步骤 2: 按照公式(8)和公式(10)分别更新 VBBPSO 与 CBBPSO 种群中粒子的位置;

步骤 3: 计算粒子的适应度值, 并更新每个粒子的历史最优值与各个种群的全局最优值;

步骤 4: CBBPSO 混沌变异;

步骤 5: 判断 CBBPSO 种群中全局最优值的停滞迭代代数是否大于 5:

若是, 则按照公式(9)更新 CBBPSO 种群内粒子历史最优位置与全局最优位置;

若否, 则直接执行下一步;

步骤 6: 粒子群进行交流, 比较 VBBPSO 与 CBBPSO 的全局最优值的适应度值大小, 并选取两者中最优的一个, 将其作为较差最优值种群的新的全局最优;

步骤 7: 重复步骤 2~6 直至迭代次数满足最大迭代次数。

VBBPSO 与 CBBPSO 拥有各自的特性, VBBPSO 善于在局部范围内细化搜索, 提高搜索精度, 而 CBBPSO 则善于在全局范围内搜索, 能跳出局部最优, 增大寻找到全局最优的概率。DBBPSO 在融合 VBBPSO 与 CBBPSO 各自特点的基础上, 引入粒子群交流机制, 使两个种群在迭代的过程中既遵循各自种群内信息交流的规律, 又吸收另一个种群带来的群体经验来改进各自种群, 两个种群相互引导, 协同进化。改进后的 DBBPSO 不仅能扩大搜索范围, 避免早熟现象的发生, 还具有较快的收敛速度和较高的寻优精度。

3 骨干双粒子群优化算法求解柔性作业车间调度问题

3.1 粒子的编码与解码

为了解决柔性作业车间调度问题, 需要对粒子按照一定规律进行编码与解码操作, 从而使 DBBPSO 算法应用在求解 FJSP 问题上。本文用两段 N_a (待加工的工序总数) 维的向量表示粒子^[18], 第一段向量记录各工件工序加工顺序, 记做工序向量 $X_{process}[Na]$, 工件出现的次数表示该工序是相应工件的第几道工序; 第二段向量记录对应各个工序选择的加工机器号, 记做机器向量 $X_{machine}[Na]$ 。

随后在迭代过程中, 粒子根据 DBBPSO 算法规则更新粒子的位置, 对每个粒子的两段向量按照更新位置从小到大排列, 得到粒子新的排序后的向量。每个粒子代表一个可能的调度方案, 排序后的

粒子向量根据工序向量 $X_{process}[Na]$ 中各个工序的排列顺序确定工序的加工顺序, 而对应的机器向量 $X_{machine}[Na]$ 的机器号, 则表示对应工序选择的加工机器。根据得到的调度方案, 可以得出每个调度方案的最大完工时间, 从而得到适应度值, 按照 DBBPSO 规则进行下一步运算。

3.2 机器选择策略

在上一节中, 任意粒子根据每一维更新后的位置大小排序得到工序向量 $X_{process}[Na]$, 从而确定调度方案中工序的加工顺序, 在这之后, 还需要对每个工序加工的机器进行选择。针对最大完工时间最小这一调度目标, 本文设计了一种机器选择策略。

从公式(1)中可以看出, 为了让最大完工时间最小, 我们应该尽量选择对应工序加工时间较少的机器进行加工。根据 1.2 节中的约束条件, 假设第一道工序 O_1 选择机器 M_1 进行加工, 且加工时间为 T_1 , 下道工序 O_2 如果仍然选择机器 M_1 进行加工, 且对应加工时间为 T_2 , 那么 O_2 的实际完工时间为 T_1+T_2 , 在这之后加工的工序以此类推, 每个工序的实际加工完工时间大于等于所选机器加工工序的时间之和。

本文提出的机器选择策略, 则是在确定工序的加工顺序之后, 按照工序的加工顺序在每个工序的可选机器子集中选择对应的完工时间最小的机器作为加工机器。同时, 为了增加机器选择的多样性, 增加变异因子 $\gamma=0.02$, 对应的变异工序的机器随机选择。具体机器选择策略基本流程可概括为算法 2:

算法 2: 机器选择策略基本流程

步骤 1: 初始化参数, 制作所有工序对应的每个机器的加工时间表, 加工时间表初始值为对应机器加工该工序的加工时间;

根据工序加工顺序, 对第 $i(1 \leq i \leq N_a)$ 个工序的加工机器进行选择:

步骤 2: 生成随机数 p , 若 p 大于变异因子 γ 则执行步骤 3, 若 p 小于变异因子 γ 则执行步骤 4;

步骤 3: 对照加工时间表, 选择该工序的机器子集中加工时间最小的机器进行加工, 在机器向量 $X_{\text{machine}}[Na]$ 中记录对应加工机器号, 加工时间为 T_i , 执行步骤 5;

步骤 4: 随机选择该工序的机器子集中任意一台机器进行加工, 在机器向量 $X_{\text{machine}}[Na]$ 中记录对应加工机器号, 加工时间为 T_i , 执行步骤 5;

步骤 5: 在加工时间表中, 对所选机器对应的所有工序加上加工时间 T_i , 执行步骤 6;

步骤 6: $i=i+1$, 返回步骤 2。

4 仿真实验

4.1 Kacem 经典算例实验

为了验证所提 DBBPSO 算法求解 FJSP 的有效性, 选取 Kacem^[19] 中的 4 个 FJSP 算例进行仿真实验。实验平台为: Windows2007 Intel®Core™ i5-4200U 4.9 GHz, 4 GB RAM, 算法采用 Matlab 编程语言实现。其中 DBBPSO 算法设置如下:

每个种群规模 $P_i=50$, 总体种群规模 $P=100$, $\alpha=0.65$, $\beta=0.3$, rows=5, cols=10, 最大迭代次数 Gen=300, 每个算例独立运行 10 次。4 个算例分别为问题 4×5(T-FJSP), 8×8(P-FJSP), 10×7(T-FJSP), 10×10(T-FJSP), 其中“问题 $n \times m$ ”表示 n 个工件, m 台机器。

DBBPSO 算法与文献[20]提出的分散式粒子群算法(dPSO), 文献[21]的 GENACE 算法以及文献[5]提出的基于邻域搜索策略的改进粒子群算法(IPSO)的最优值进行比较, 结果如表 1 所示, 其中“--”表示原文中没有给出相应数据, 每个问题最好结果均用粗体标出。

表 1 Kacem 算例测试最优值比较
Tab. 1 Best comparison of Kacem

问题 $n \times m$	文献[19]	GENACE	dPSO	IPSO	DBBPSO
4×5	16	11	11	11	11
8×8	16	--	14	17	14
10×7	15	12	11	--	11
10×10	7	7	7	8	7

从表 1 可以看出, DBBPSO 算法所获得的最优解在 4 个不同 FJSP 问题上明显优于或等于其他 4 个对比算法所得到的最优解, 表明 DBBPSO 在 FJSP 问题中有较好的寻优能力, 解的质量有较大的提高。

对每个算例独立运行 10 次, 将每次得到的最优解求平均值, 并与文献[5]的 IPSO 算法、文献[22-23]分别提出的改进遗传算法实验测试的平均值进行对比实验, 实验结果记录在表 2 中。

表 2 Kacem 算例测试平均值比较
Tab. 2 Average comparison of Kacem

问题 $n \times m$	文献[23]	文献[22]	IPSO	DBBPSO
4×5	--	--	11	11
8×8	17.9	16.2	14.4	14.35
10×7	--	--	11	11
10×10	9.1	7.8	7.2	7

从表 2 中可以看出 DBBPSO 对于 4×5、10×7 和 10×10 问题均能全部稳定寻找到最优值, 而对于 8×8 问题其平均值明显优于其它 3 个算法, 这说明 DBBPSO 具有较高的稳定性和可靠性。综合表 1 与表 2, DBBPSO 在解决 FJSP 问题上无论是解的质量还是求解的稳定性均较好, 寻优能力较强且稳定性较高。

图 1~4 分别为 DBBPSO 对于 Kacem 的 4 个经典算例仿真所得最优解的调度甘特图。

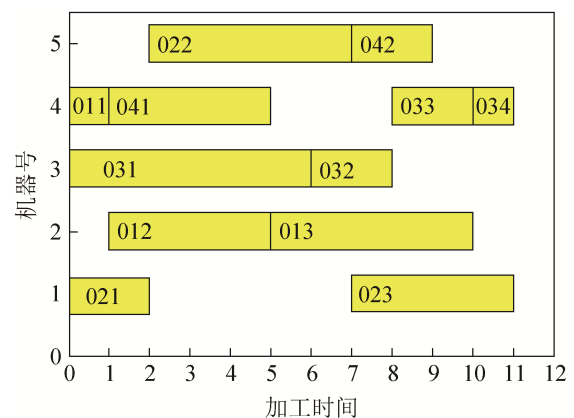


图 1 4×5 算例甘特图

Fig. 1 Gantt of 4×5 Examples

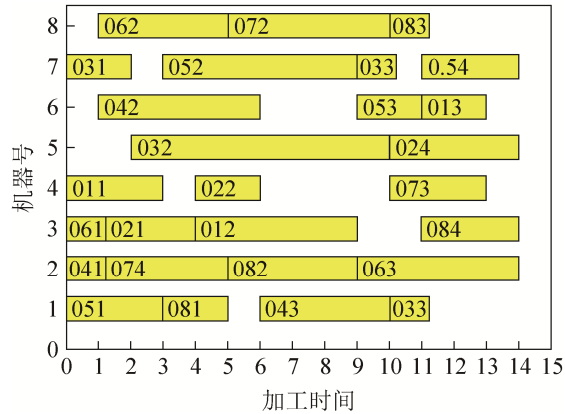


图 2 8×8 算例甘特图
Fig. 2 Gantt of 8×8 Examples

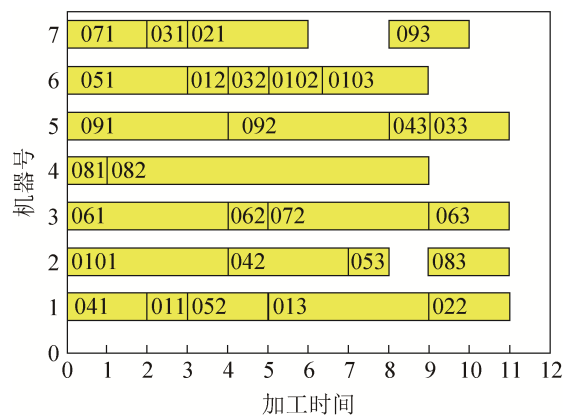


图 3 10×7 算例甘特图
Fig. 3 Gantt of 10×7 examples

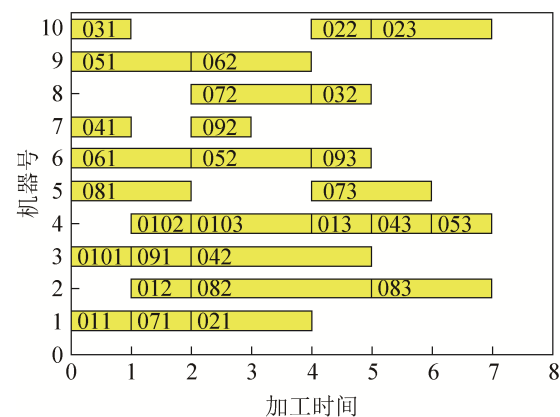


图 4 10×10 算例甘特图
Fig. 4 Gantt of 10×10 examples

4.2 车间实例模型实验

为进一步验证所提算法的有效性和优越性,选取了一个车间实例模型进行仿真实验。实验模型采用 8 台机器,6 个工件,共 26 道工序,工件各工

序所对应的机器加工时间如表 3 所示。

表 3 各工序在对应机器上加工时间
Tab. 3 The time of processes on machine

工件	工序	加工时间 t							
		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
J_1	O_{11}	--	12	--	--	14	--	20	--
	O_{12}	18	--	--	19	--	--	11	--
	O_{13}	--	--	14	9	--	17	--	--
	O_{14}	--	11	--	--	9	--	--	12
	O_{15}	15	--	--	8	--	--	--	18
J_2	O_{21}	--	--	12	19	14	--	--	--
	O_{22}	8	--	--	9	--	--	15	-
	O_{23}	16	7	--	--	--	9	--	--
	O_{31}	--	--	--	11	10	--	--	13
	O_{32}	--	--	12	18	--	--	14	--
J_3	O_{33}	9	--	--	15	7	--	--	--
	O_{34}	--	12	--	--	--	5	--	9
	O_{35}	3	--	--	4	--	8	--	--
	O_{41}	--	19	--	--	7	--	13	--
	O_{42}	--	--	8	--	11	--	--	16
J_4	O_{43}	--	11	--	8	--	--	18	--
	O_{44}	6	--	--	--	--	--	14	9
	O_{51}	--	--	--	22	--	12	17	--
	O_{52}	--	18	--	--	11	--	--	9
	O_{53}	9	--	12	--	--	--	--	7
J_6	O_{61}	--	11	--	--	--	9	14	--
	O_{62}	8	--	--	--	6	--	--	9
	O_{63}	--	--	11	--	17	--	--	18
	O_{64}	5	--	--	15	--	--	7	--
	O_{65}	--	11	--	--	--	8	--	7
	O_{66}	19	--	--	7	--	15	--	--

将 DBBPSO 与文献[18]提出的改进双层粒子群 ITLPSO 算法,文献[6]的改进量子粒子群 OBL-QPSOB 算法进行对比实验。实验环境与上一实验环境相同,其中每个算法的种群规模均为 50,迭代次数 4 000,实验独立运行 30 次,得到的平均收敛曲线如图 5 所示。记录各个算法得到的最优值(best)、平均值(mean),同时将运行 30 次得到的结果进行显著水平为 0.05 的 Wilcoxon 秩和检验,其中有显著性差异的结果用“+”表示,无显著性差异的结果用“-”表示,没有进行检验的用“=”表示,具体结果记录在表 4 中。

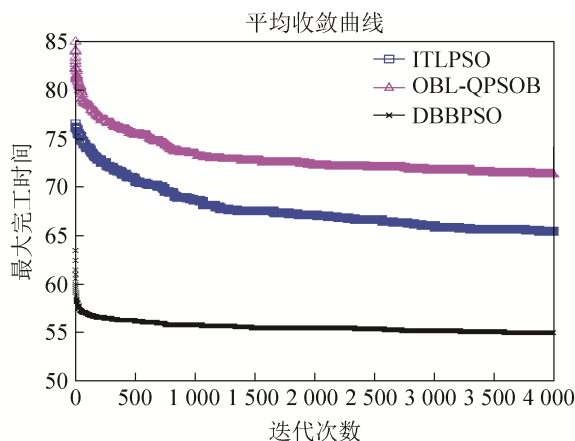


图 5 3 种算法求解 FJSP 的平均收敛曲线图

Fig. 5 The average convergence curve of 3 algorithms for solving FJSP

表 4 3 种算法求解 FJSP 结果比较

Tab. 4 The results of 3 algorithms for solving FJSP

	ITLPSO	OBL-QPSOB	DBBPSO
best	60	67	53
mean	64.7	70.733 3	53.833 3
Wilcoxon	+	+	=

从图 5 中可以明显看出, DBBPSO 算法收敛速度和收敛精度均优于其他对比算法, 同时表 4 中的数据也可以表明 DBBPSO 算法解的质量和稳定性均为最优。Wilcoxon 秩和检验也表明 DBBPSO 算法在求解 FJSP 问题上显著优于其他对比算法。最后, 图 6 显示一组最优解对应的调度甘特图。

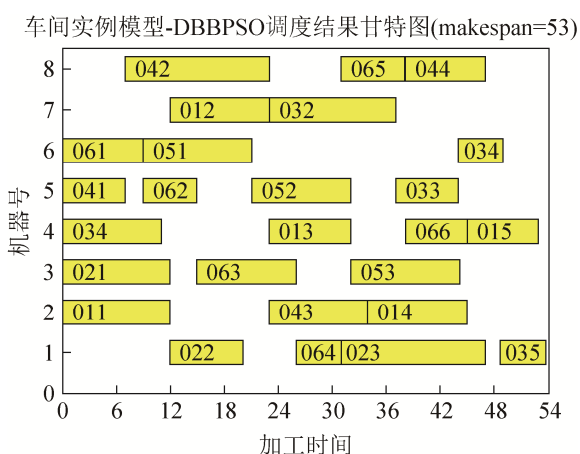


图 6 车间实例模型调度结果甘特图

Fig. 6 Gantt of examples in workshop

5 结论

本文针对 PSO、BBPSO 算法易早熟、易陷入局部最优等缺点, 利用冯诺依曼拓扑结构与混沌变异改进了两个 BBPSO 算法, 并通过种群交流机制使他们协同进化, 进而改善 BBPSO 算法性能, 并将改进算法应用到解决 FJSP 问题上。在解决 FJSP 问题过程中, 为实现最大完工时间最小这一优化目标, 提出了一种基于最小加工时间的机器选择策略。通过对 Kacem 的 4 个经典算例与一个车间实例模型的仿真测试表明, 本文所提的 DBBPSO 算法在求解 FJSP 问题时无论是解的质量还是稳定性均有较好的表现。

本文所提算法优化的 FJSP 模型是单目标的, 将 BBPSO 算法应用到求解多目标 FJSP 问题将是下一步研究的方向。

参考文献:

- [1] 王万良, 吴启迪. 生产调度智能算法及其应用 [M]. 北京: 科学出版社, 2007: 12-13. (Wang Wanliang, Wu Qidi. Intelligent Scheduling Algorithm for Production and Application [M]. Beijing, China: Science Press, 2007: 12-13.)
- [2] Brucker P, Schlie R. Job-shop scheduling with multi purpose machines [J]. Computing (S0010-485X), 1990, 45(4): 369-375.
- [3] Girish B S, Jawahar N. A particle swarm optimization algorithm for flexible jobshop scheduling problem [C]// IEEE International Conference on Automation Science and Engineering. USA: IEEE Press, 2009: 298-303.
- [4] Deroussi L. A Hybrid PSO Applied to the Flexible Job Shop with Transport [M]// Swarm Intelligence Based Optimization. Germany: Springer International Publishing, 2014: 115-122.
- [5] 张静, 王万良, 徐新黎, 等. 基于改进粒子群算法求解柔性作业车间批量调度问题 [J]. 控制与决策, 2012, 27(4): 513-518. (Zhang Jing, Wang Wanliang, Xu Xinli, et al. Improved Particle Swarm Algorithm for Batch Splitting Flexible Job Shop Scheduling [J]. Control and Decision, 2012, 27(4): 513-518.)
- [6] 周恺, 纪志成. 关于柔性作业车间调度问题的仿真研究 [J]. 计算机仿真, 2016, 33(3): 282-287. (Zhou Kai, Ji Zhicheng. Simulation Research on Flexible Job Shop

- Scheduling Problem [J]. Journal of System Simulation (S1004-731X), 2016, 33(3): 282-287.)
- [7] Kennedy J. Bare bones particle swarms [C]// Proc of the Swarm Intelligence Symposium. USA: IEEE, 2003: 80-87.
- [8] 雷阳, 李树荣, 张强, 等. 基于骨干粒子群的混合遗传算法及其应用 [J]. 计算机工程与应用, 2010, 46(36): 7-10. (Lei Yang, Li Shurong, Zhang Qiang, et al. Novel hybrid GA based on bare bones PSO and its application [J]. Computer Engineering and Applications, 2010, 46(36): 7-10.)
- [9] Zhang Y, Gong D, Geng N, et al. Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects [J]. Applied Soft Computing, 2014, 18(C): 248-260.
- [10] 史丽萍, 王攀攀, 胡泳军, 等. 基于骨干微粒群算法和支持向量机的电机转子断条故障诊断 [J]. 电工技术学报, 2014, 29(1): 147-155. (Shi Liping, Wang Panpan, Hu Yongjun, et al. Broken Rotor Bar Fault Diagnosis of Induction Motors based on Bare-Bone Particle Swarm Optimization and Support Vector Machine [J]. Transactions of China Electrotechnical Society, 2014, 29(1): 147-155.)
- [11] Yao J, Han D. Improved barebones particle swarm optimization with neighborhood search and its application on ship design [J]. Mathematical Problems in Engineering (S1024-123X), 2013, 12(7): 1-13.
- [12] 高亮, 张国辉, 王晓娟. 柔性作业车间调度智能算法及其应用 [M]. 武汉: 华中科技大学出版社, 2012: 13-15. (Gao Liang, Zhang Guohui, Wang Xiaojuan. Flexible Job Shop Scheduling Intelligent Algorithm and Its Application [M]. Wuhan, China: Huazhong University of Science and Technology Press, 2012: 13-15.)
- [13] 李莉. 柔性作业车间调度中的群智能优化算法研究 [D]. 哈尔滨: 东北林业大学, 2011. (Li Li. Research on Group Intelligent Optimization Algorithm in Flexible Job Shop Scheduling [D]. Harbin, China: Northeast Forestry University, 2011.)
- [14] 陈自郁, 何中市, 张程. 基于冯诺依曼邻居的粒子群多阈值分割算法 [J]. 计算机应用研究, 2009, 26(5): 1979-2000. (Chen Zi-yu, He Zhong-shi, Zhang Cheng. Image multi-level thresholds based on PSO with Von Neumann neighborhood [J]. Application Research of Computers, 2009, 26(5): 1979-2000.)
- [15] Rifaie M, Blackwell T. Bare bones particle swarms with jumps [C]// Algorithmic Number Theory Symposium. Brussels, Belgium: Springer, 2012: 49-60.
- [16] LU H J. A new optimization algorithm based on Chaos [J]. Journal of Zhejiang University Science A (S1673-565X), 2006, 7(4): 539-542.
- [17] 吴定海, 张培林, 李胜, 等. 基于混沌变异的自适应双粒子群优化 [J]. 控制与决策, 2011, 26(7): 1083-1095. (Wu Dinghai, Zhang Peilin, Li Sheng, et al. Adaptive Double Particle Swarms Optimization Algorithm Based on Chaotic Mutation [J]. Control and Decision, 2011, 26(7): 1083-1095.)
- [18] 孔飞, 吴定会, 纪志成. 基于双层粒子群优化算法的柔性作业车间调度优化 [J]. 计算机应用, 2015, 35(2): 476-480. (Kong Fei, Wu Dinghui, Ji Zhicheng. Flexible Job-shop Scheduling Optimization Based on Two-layer Particle Swarm Optimization Algorithm [J]. Computer Application, 2015, 35(2): 476-480.)
- [19] Kacem I, Hammadi S, Borne P. Approach by localization and multi-objective evolutionary optimization for flexible job shop scheduling problems [J]. IEEE Trans on Systems, Man and Cybernetics, Part C: Applications and Reviews (S1094-6977), 2002, 32(1): 1-13.
- [20] Maroua N, Abdelghani B, Abderezak J. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem [J]. Journal of Intelligent Manufacturing (S0956-5515), 2014, 24(3), 1-13.
- [21] Ho N B, Tay J C. GENACE: An efficient cultural algorithm for solving the flexible job-shop problem [C]// Proc of the Congress on Evolutionary Computation (CEC), Portland. USA: IEEE, 2004: 1759-1766.
- [22] 张超勇, 饶运清, 李培根. 柔性作业车间调度问题的两级遗传算法 [J]. 机械工程学报, 2007, 43(4): 119-124. (Zhang Chaoyong, Rao Yunqing, Li Peigen. Two - level Genetic Algorithm for Flexible Job Shop Scheduling Problem [J]. Journal of Mechanical Engineering, 2007, 43(4): 119-124.)
- [23] 张超勇, 刘琼, 邱浩波, 等. 考虑加工成本和时间的柔性作业车间调度问题研究 [J]. 机械科学与技术, 2009, 28(8): 1005-1011. (Zhang Chaoyong, Liu Qiong, Qiu Haobo, et al. Research on Flexible Job Shop Scheduling Problem Considering Processing Cost and Time [J]. Mechanical Science and Technology, 2009, 28(8): 1005-1011.)