

一种粒子群和改进自适应差分进化混合算法及在生产调度中的应用

周艳平, 蔡素, 李金鹏

(青岛科技大学 信息科学技术学院, 山东 青岛 266061)

摘要: 差分进化算法是一类基于种群的启发式全局搜索技术, 但传统的差分进化算法存在停滞现象, 容易使算法收敛停止; 虽然之后出现了各种版本的自适应差分进化算法, 但没有考虑到当代个体的适应值是否向着最优个体的适应值逼近, 因此提出了一种新型的自适应差分进化算法 FMDE; 考虑到粒子群算法和差分进化算法类似, 为了充分发挥两种算法的特点, 提出了自适应差分进化和粒子群的混合算法 PSO_FMDE; 最后采用测试数据集对性能进行分析; 实验结果表明, 该算法根据进化过程中的搜索进度自适应地确定变异率, 使算法易于跳出局部最优解, 以提高全局搜索能力。PSO_FMDE 算法较单一算法而言, 性能更优, 更易于靠近全局最优解。

关键词: 自适应; 差分进化算法; 变异率; 车间调度

A Hybrid Algorithm Based on Combination of Different Evolution and Particle Swarm Optimization and Its Application on Flow Shop Scheduling Problem

Zhou Yanping, Cai Su, Li Jinpeng

(School of Computer Science, Qingdao University of Science & Technology, Qingdao 266061, China)

Abstract: Differential evolution algorithm is a heuristic global optimization technique based on population. But conventional differential evolution algorithm has a problem of stagnation that can stop the algorithm convergence. Although various versions of adaptive differential evolution algorithms have emerged, it does not take into account whether the fitness of the contemporary individual approximates to the fitness of the optimal individual. An adaptive differential evolution algorithm FMDE is proposed. In order to take advantage of different algorithms, a hybrid optimization algorithm is proposed, based on the combination of different evolution and particle swarm optimization. Finally, the performance is tested and analyzed by using the data of instance. The results show that PSO_FMDE can determine mutation rate adaptively, which enhances the probability of obtaining the global optimum. Comparing with single algorithm, PSO_FMDE algorithm has better performance and is easier to get close to the global optimal solution.

Keywords: differential evolution algorithm; adaptive; mutation operator; flow shop scheduling; particle swarm optimization

0 引言

随着社会的发展, 流水车间调度问题正在引起广泛的关注。流水车间调度问题是指在一定的时间内, 对可用共享资源的分配和生成任务进行合理科学的安排, 从而可以在较短时间内获取较优的调度方案。求解流水车间调度问题的方法有很多, 如人工智能算法、精确求解方法。其中人工智能算法更适合求解车间调度问题, 尤其是差分进化算法和粒子群算法。差分进化算法是可以利用种群个体之间的差异从而逐步进化的一种搜索算法^[1]。该算法是 R. Storn 和 K. Price 为了求解 Chebyshev 多项式而提出的^[2-3], 而粒子群算法是 J. Kennedy 和 R. C. Eberhart 提出的一种进化算法^[4], 通过追随当前搜索到的最优值来寻找全局最

优, 并显示出实际问题的优越性。

本文首先介绍了基于自适应变异算子的差分进化算法, 并且分析了此算法的缺点, 提出一种粒子群算法和改进的自适应差分进化混合算法 PSO_FMDE; 最后通过实验分析了该算法的优异性能。接着描述了流水车间调度问题的模型, 并且通过实验分析了 PSO_FMDE 算法求解流水车间调度问题的突出表现。

1 粒子群和改进自适应差分进化混合算法

1.1 粒子群和改进自适应差分进化混合算法

粒子群算法的基本思想是首先初始化种群, 计算种群中个体的适应值, 然后根据适应值去更新粒子的速度和位置, 当前个体通过与种群中最好个体比较跟踪^[5-6], 从而产生新个体。

对于差分进化算法, 首先会通过随机初始化策略把初始种群进行初始化操作, 然后随机选取两个不同的个体与互不相同的第三个个体通过线性组合的方式产生一个变异后的新个体, 再对新个体通过特定的交叉函数得到交叉后

收稿日期: 2019-02-20; 修回日期: 2019-03-19。

基金项目: 国家自然科学基金项目(61402246)。

作者简介: 周艳平(1976-), 男, 山东临沂人, 硕士生导师, 副教授, 主要从事智能优化算法、生产计划与生产调度方向的研究。

的个体，最后将新个体与当前最优个体比较，从而选出一代的最优个体。因为标准的差分进化算法变异因子是恒定的，所以在种群初期个体较多进化相对较慢，种群后期个体较少进化相对较快。所以颜学峰等人提出了一种具有自适应变异算子的差分进化算法 MDE^[7]，根据算法搜索进度自适应的变异算子如下：

$$\lambda = e^{1-\frac{G}{G_m}}, F = F_0 \cdot 2^\lambda \quad (1)$$

其中： F_0 为变异算子； G_m 为最大进化代数； G 为当前进化代数。

该算法在种群进化初期具有较大的变异因子值 $F = 2F_0$ ，因此初期个体变化较快，个体多样；随着算法的进展变异率逐渐降低，到了算法的后期变异率已经接近于 F_0 ，从而避免了最优解的破坏，保留了良好的信息，提高了搜索到全局最优解的概率。该算法和大多数算法一样，虽然具有了种群初期进化较快，种群后期进化较慢的特性^[8-9]，但是没有考虑到当代个体的适应值是否向着最优个体的适应值逼近，即个体适应值是否朝着最优解的方向发展。假设当代个体经过变异、交叉后，其适应值并未向着最优个体的适应值逼近，偏离了正确的迭代方向，此时恰巧变异算子却变小了，导致整个子代个体近乎速度放缓，不易快速收敛到全局最优值^[10-11]。因此本文提出了一种新的自适应差分进化算子 FMDE，其中变异算子 F 的定义为：

$$F = \begin{cases} F_0 \cdot 2^\lambda + e^{(|f_i(pre)+f_i(suf)| - |f_i(pre)-f_i(suf)|)}, \\ |f_i(pre) - f(\min)| < |f_i(suf) - f(\min)| \\ F_0 \cdot 2^\lambda, qita \end{cases}$$

$$\lambda = e^{1-\frac{G}{G_m}} \quad (2)$$

式中， $f_i(pre)$ 代表上一代个体的函数值， $f(\min)$ 代表当代的目标函数最优值， $f_i(suf)$ 代表当代个体的函数值。如果当代个体适应值与当代最优值的差的绝对值比上一代个体适应值与最优值的差的绝对值要小，说明当代的此个体偏离了最优值的进化方向，这时采用适当增加变异算子 F ，稍微加快进化速度，但又考虑到变异算子的取值范围，所以采取添加一个单调递减的指数函数的策略^[12]。

粒子群算法在优化求解过程中早起具有较快的收敛速度，在后期容易陷入局部最优，难以获得全局最优解。种群差分进化算法会从当前最优个体和经过变异、交叉后产生的新个体中选择一个较优个体作为当前最优个体。只有当适应值优于父代个体使才选为子代，否则父代直接进入下一代，该机制增加了算法的收敛速度。但该算法有陷入局部最优解的缺点，并且粒子群算法和该算法有一定的相似度，因此本文考虑将两种算法结合，首先让两种算法分别迭代，接着为了避免混合算法的迭代后期出现停滞现象，引入一种随机跳出机制，通过一个随机函数使出现停滞现象的个体变成一个新个体，这样就跳出了局部最优，实现如下：

$$x_i^{t+1+M} = X_{\min} - rand(0,1) \cdot (X_{\max} - X_{\min}) \quad (3)$$

其中： M 表示可以允许停滞的最大迭代次数， $(X_{\min},$

$X_{\max})$ 是可以允许的搜索范围。

因此，本文提出的混合算法 PSO _ FMDE 实现步骤如下：

算法 PSO _ FMDE 算法

输入：基本的输入参数，包括 R_1 、 R_2 、 c_1 、 c_2 、 v_{max} 、 $iter_{max}$ 、 F_0 、 G_m 、 G 等

输出：最优个体最优值

1) 对两个种群 POP^{PSO} 和 POP^{FMDE} 进行随机初始化，而且位于不同区域。

2) 开始迭代，根据公式 (1) ~ (3) 对 POP^{PSO} 种群中所有个体进行位置和速度更新；

3) 利用改进的自适应差分进化算法 FMDE 对种群中所有个体执行变异、交叉、选择操作。

4) 分别选择当前迭代中，POP^{PSO} 种群以及 POP^{FMDE} 种群中的最佳个体。

5) 选择两个种群中较优的个体，并判断个体是否出现停滞现象，陷入局部最优，如果陷入局部最优，则执行步骤 6)，否则执行步骤 7)；

6) 根据公式 (6) 更新个体位置，然后重复计算每个种群中个体适应值，选择较优的个体；

7) 判断当前是否达到最大迭代次数或者全局最优位置满足最小界限，如果满足退出条件，输出最终结果并退出，否则返回步骤 2)；

步骤 5 中判断当前的结果陷入局部最优，出现停滞现象的方法如下：

$$\frac{f(x_i^{t-1})}{f(x_i^t)} = 1 \quad (4)$$

如果上一次迭代和本次迭代的个体适应值相同即出现了局部最优现象。如果 $f(x_i^t) = f(x_i^{t-1}) = \dots = f(x_i^{t-M})$ ，其中 M 代表可以允许停滞的最大迭代次数，如果连续 M 次迭代出现个体适应值相等，那么就出现停滞现象，陷入局部最优，需要利用公式 (3) 在被搜索空间中找一新位置代替，从而跳出局部最优解。

1.2 函数优化测试

为了测试和解释 PSO _ FMDE 算法在求解全局最优问题的性能和优越性，将该算法分别与 MDE、FMDE 进行了对比，然后对以下的数学优化问题进行寻优^[13]。

$$f(x) = \sum_{i=1}^D x_i^2 + 10\cos(2\pi x_i) + 10$$

$$-5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, 10 \quad (5)$$

优化问题的目标函数在 10 维可行域内有 $2^{10} - 1$ 个局部最优解，差分进化算法可以求解这一问题。为了保证实验的有效性，3 种算法的参数尽可能一致，其中种群大小为 $N_p = 100$ ，最大进化代数是 800。交叉参数 $C_R = 0.1$ ，加速常数 $C_1 = 0.1$ ， $C_2 = 0.2$ 惯性权重参数 $\omega = 0.9$ ，允许的最大停滞次数 $numpause = 10$ 等。

为了研究变异因子对算法的性能影响, 变异因子 F 分别取值 0.1, 0.5, 0.7, 1.5, 2。图 1、图 2、图 3 分别展示了 MDE、FMDE、PSO_FMDE 三种算法在不同的变异率情况下, 最优值与迭代次数的关系。图 x 轴代表迭代的次数, y 轴代表最优值。

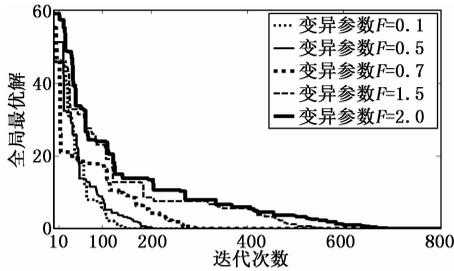


图 1 MDE 算法变异参数实验

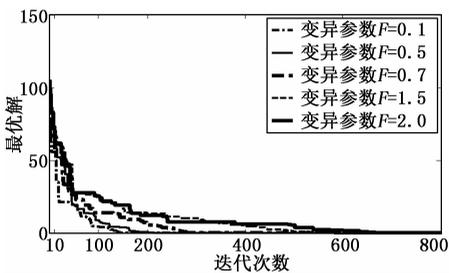


图 2 FMDE 算法变异参数实验

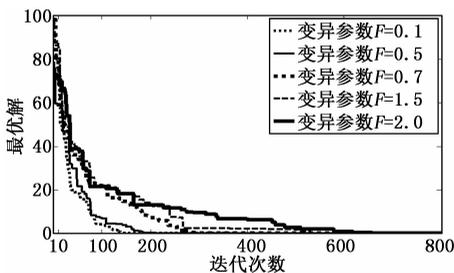


图 3 PSO_FMDE 算法变异参数实验

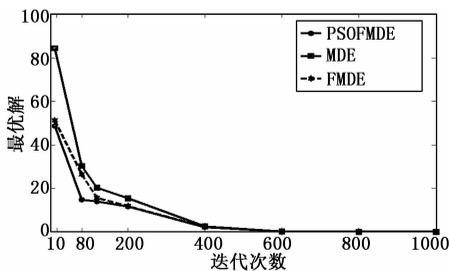


图 4 3 种算法的性能比较

从三幅图可以看出随着变异参数的增加, 会需要更多的迭代次数才能达到最终的全局最优解 (即进化结束), 比如 $F=0.1$ 时只需要 100 次迭代就能进化完成, $F=0.7$ 时需要大约 250 次能完成进化, 但是 $F=2$ 时会需要将近 450 次才能完成进化, 性能逐渐减小。同时也能看出, 在固定的迭代次数时, 变异率越小, 最优解的值就更靠近全局最优解。

图 4 展示了不同算法在相同变异因子下的性能, 此时 $iter_{max}=1000, F=1.5$, 通过结果可以看出, 在迭代初期 (前 200 次迭代), PSO_FMDE 算法能够更快地靠近全局最优解 0, 其次是 FMDE 算法。当迭代次数固定时, PSO_FMDE 算法到 MDE 算法靠近全局最优解地速度依次降低, 也就是说从 PSO_FMDE 算法到 MDE 算法的性能逐渐下降, 表明 PSO_FMDE 算法使群体的平均适应度和最优适应度和其他算法相比都有适当的提高。

2 PSO_FMDE 算法求解流水车间调度问题

2.1 流水车间调度问题模型

问题描述: 如果要在 m 台机器上利用 n 个工件进行流水加工, 每个工件需要经过 m 道工序, 每个工件在每台机器上只加工一次, 每台机器在某一时刻只能加工一个工件, 而且各个工件在机器上所需要的加工时间和准备时间已知, 且各个工序都有相应的约束条件, 以使工件完工时间最小。

令 c_{ik} 代表工件 i 在设备 k 上的完成时间。 t_{ik} 代表工件 i 在设备 k 上的加工时间, 且 $c_{ik} > 0, i, j=1, 2, \dots, n, k=1, 2, \dots, m$ 。该流水车间调度问题的目标函数为:

$$f(x) = \min \max_{1 \leq k \leq m, 1 \leq i \leq n} c_{ik} \quad (6)$$

约束条件为:

- 1) 设备 h 先于设备 k 加工工件 $i: c_{ik} - t_{ik} \geq c_{ih};$
- 2) 工件 i 先于工件 j 在设备 k 上加工: $c_{jk} - t_{jk} \geq c_{ji}。$

2.2 PSO_FMDE 求解流水车间调度问题的步骤

差分进化算法求解流水车间调度问题和遗传算法求解车间调度问题类似^[14-15], 主要分为以下几个步骤:

- 1) 设置差分进化算法的参数, 包括最大迭代次数、变异因子等。
- 2) 根据流水车间调度问题, 随机初始化两个种群。
- 3) 根据交叉变异或者速度位置更新得到临时种群, 对临时种群进行评价。
- 4) 选择较优个体, 并判断是否陷入局部最优, 如果是则求解公式 (3)。

5) 并判断满足终止迭代条件。如果满足, 根据最优个体得到流水车间调度问题的最佳调度方案; 否则进行进化操作, 转步骤 3)。

2.3 仿真实验

采用标准 TA 类^[16] 车间调度问题进行实验测试, 设置粒子群算法和改进自适应差分进化的混合算法的种群规模是 40, 最大进化次数是 800。数据集分别选择 20 工件 5 机器、50 工件 5 机器以及 100 工件 5 机器, 在上述 3 个数据集上进行流水车间调度问题的实验。实验采用 Pascal 语言编程实现流水车间调度优化算法。

3 种算法 MDE、FMDE、PSO_FMDE 分别对 3 种不同规模的数据最进行 makespan 目标最小化, 比较几种算法的性能。在测试实验过程中, 为了实验结果科学及更具说服力, 3 种算法分别运行 10 次, 然后分析 10 次结果的最优值 (makespan) 和平均值, 几种不同规模数据的优化结果

如表 1 所示。

表 1 不同规模的不同算法的最优值

PS	MDE	FMDE	PSO_FMDE	Best
20 * 5	1362/1363/1387 /1363/1368	1324/1332/1323 /1315/1318	1286/1297/1291 /1295/1297	1286
	1360/1358/1363 /1360/1362	1338/1320/1332 /1328/1321	1297/1289/1288 /1294/1290	
	1364.5	1325.1	1292.4	
	2839/2837/2887 /2886/2881	2792/2802/2803 /2815/2801	2759/2777/2740 /2752/2769	
2867/2874/2879 /2853/2875	2803/2815/2808 /2807/2799	2767/2750/2764 /2774/2739		
2867.8	2805	2759.1		
100 * 5	5543/5564/5544 /5530/5545	5510/5521/5515 /5512/5511	5512/5498/5517 /5507/5508	5494
	5556/5569/5535 /5541/5544	5507/5516/5515 /5524/5511	5507/5494/5505 /5507/5505	
	5547.1	5514.2	5506	

车间调度问题，将最小化最大完成时间作为调度方案的最优评价指标，因为该算法加快了全局搜索速度和跳出局部最优解的进度，所以具有一定的实践意义。

参考文献:

[1] 周艳平, 顾幸生. 差分进化算法研究进展 [J]. 化工自动化及仪表, 2007, 34 (3): 1-5.

[2] Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces [J]. Global Optimization, 1995, 23 (1): 341-359.

[3] Price K V. An introduction to differential evolution [J]. New Ideas in Optimization, 1999: 79-108.

[4] Kennedy J, Eberhart R. Particle swarm optimization [A]. IEEE International Conference on Neural Networks [C]. 2002: 1942-1948.

[5] Bonyadi M R, Michalewicz Z. Particle swarm optimization for single objective continuous space problems: a review [J]. Evolutionary Computation, 2017, 25 (1): 1-54.

[6] 王 植. 基于差分进化的混沌量子粒子群优化算法 [J]. 计算机与现代化, 2017, 8 (8): 22-25.

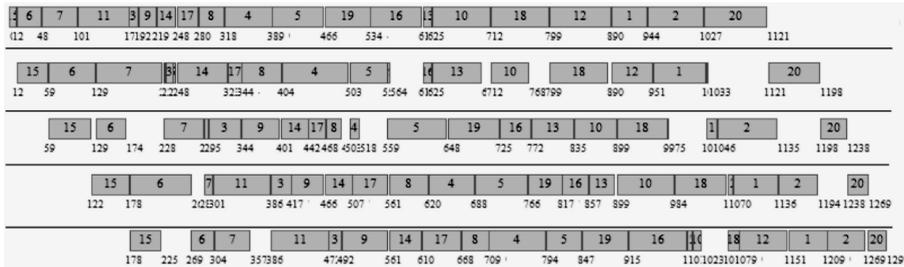


图 5 20 * 5 PSO _ FMDE 算法的最优调度方案

对于同一个数据规模单从最优值来看，PSO _ FMDE 算法比其他算法的最优值更理想。为了进一步分析算法性能，将算法平均值 (Avg) 的差作为评价指标。在 3 种不同规模的数据集 (20 * 5、50 * 5、100 * 5) 下，PSO _ FMDE 算法的平均值与 FMDE、MDE 算法的平均值差值分别是 72.1、108.7、121.1。从差值可以看出，在相同机器数量时，随着工件数的增加 PSO _ FMDE 算法较其他算法的优化强度也随之增加，说明 PSO _ FMDE 算法随着数据规模越大就会越有效。因此，PSO _ FMDE 算法有着较好的有效性以及稳定性。

图 5 给出了使用 PSO _ FMDE 算法对规模为 20 * 5 的调度集进行优化时得到的第一次计算结果的甘特图。

3 结语

考虑到粒子群算法和改进的自适应差分进化算法的异同，将两种算法进行结合，提出了基于粒子群和改进的自适应差分进化的混合算法。因为进化过程中可能会出现局部最优的问题，因此本文采用随机个体替换停滞个体的方法，以便快速跳出局部最优解区域。通过多种算法的性能对比，较单一算法相比，该有更优更靠近最优解的优势，具有一定的实用价值。为了用 PSO _ FMDE 算法求解流水

[7] 颜学峰, 余 娟, 钱 锋. 基于改进差分进化算法的超临界水氧化动力学参数估计 [J]. 华东理工大学学报, 2006, 32 (1): 94-97.

[8] Qin A K, Suganthan P N. Self-adaptive differential evolution algorithm for numerical optimization [A]. Proceedings of the 2005 IEEE Congress on Evolutionary [C]. 2005, pp: 1785-1791.

[9] 刘龙龙, 颜七笙. 一种新的改进差分进化算法 [J]. 江西科学, 2017, 35 (4): 486-490.

[10] 陈 皓, 张洁杨, 清 萍. 基于混合搜索的多种群人工蜂群算法 [J]. 计算机应用, 2017, 37 (10): 2773-2779, 2786.

[11] 翁志远, 方 杰, 孔 敏, 等. 改进差分进化算法的作业车间调度问题优化策略 [J]. 控制工程, 2017, 24 (6): 1282-1285.

[12] Zhang J, Sanderson A. Jade: adaptive differential evolution with optional external archive [J]. IEEE Transactions on Evolutionary Computation, 2009, 13 (5): 945-958.

[13] Fan H Y, Lampinen J. A trigonometric mutation operation to differential evolution [J]. Journal of Global Optimization, 2003, 27 (1): 105-129.

[14] 王 悦. 遗传算法在函数优化中的应用研究 [J]. 计算机技术与应用, 2016, 24 (10): 74-76.

[15] Corus D, Oliveto P S. Standard steady state genetic algorithms can hillclimb faster than mutation-only evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation, 2017: 99.

[16] 王 凌. 车间调度及其遗传算法 [M]. 北京: 清华大学出版社, 2003.