

多策略差分进化的元胞多目标粒子群算法

朱大林, 詹 腾, 张 屹, 郑小东

(三峡大学水电机设备设计与维护湖北省重点实验室, 湖北宜昌 443002)

摘 要: 为了增加 Pareto 解集的多样性, 平衡多目标优化的全局搜索和局部寻优的能力, 提出一种多策略差分进化的元胞多目标粒子群算法. 该算法在分析粒子群优化原理基础上, 将元胞自动机理论融入粒子群算法, 研究粒子种群交流结构和信息传递机制. 为了避免粒子飞行速度过快陷入局部收敛, 提出一种限制粒子飞行速度的策略, 并引入一种多策略差分进化选择算子增加对粒子的扰动. 实验证明, 该算法相对于比较算法, 有更好的收敛性和多样性.

关键词: 元胞自动机; 粒子群算法; 多策略差分进化; 速度控制策略; 多目标优化

中图分类号: TP38 **文献标识码:** A **文章编号:** 0372-2112 (2014)09-1831-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2014.09.026

Cellular Multi-Objective Particle Swarm Algorithm Based on Multi-Strategy Differential Evolution

ZHU Da-lin, ZHAN Teng, ZHANG Yi, ZHENG Xiao-dong

(Hubei Key Laboratory of Hydroelectric Machinery Design & Maintenance, China Three Gorges University, Yichang, Hubei 443002, China)

Abstract: In order to strengthen the diversity of Pareto sets obtained by multi-objective optimization algorithms and balance the exploration and exploitation of the algorithms, a cellular multi-objective particle swarm optimization algorithm based on multi-strategy differential evolution (MPSOCell) is proposed. This algorithm is composed by integrating the cellular automata mechanism into the basic particle swarm optimization algorithm, and it is aimed at promoting the communication and information transmission among the particles. To avoid the local convergence caused by the fast flying speed of particles, a strategy used to limit the flying speed is designed; to strengthen the disturbance to the particles, a multi-strategy differential evolution operator is also brought into the algorithm. The experiments demonstrate that MPSOCell has better performance in terms of convergence and diversity.

Key words: cellular automata; particle swarm optimization; multi-strategy differential evolution; speed control strategy; multi-objective optimization

1 引言

粒子群算法 (Particle Swarm Optimization, PSO), 相对于其他算法, 其参数设置更少、实现简单, 且大多数情况下计算效率较高, 因此许多学者采用 PSO 来构建混合多目标进化算法 (Multi-Objective Evolutionary Algorithms, MOEAs) 来解决多目标问题 (Multi-Objective Problems, MOPs). 其中典型算法有, Li^[1]在 NSGA-II 中引入 PSO, 提出一种非支配搜索的粒子群算法 (NSPSO), 该算法类似于 NSGA-II, 将每次迭代的粒子种群同父代种群进行合并, 并依据支配关系和拥挤距离选择出子代种群. NSPSO 在解决 ZDT 系列函数上较 NSGA-II 有较明显的优势; Srinivasan 和 Seow^[2]将 PSO 和 EA 进行融合, 提出一种粒子群进化算法 (Particle Swarm inspired Evolutionary Algorithm, PS-EA), 该算法使用一种自我更新机制去更新每

一个粒子的位置; Santana-Quintero 等^[3]将分散搜索算子引入 PSO 算法中, 得到一种两阶段的混合进化算法; Tsou 等^[4]将局部搜索和聚类机制加入多目标粒子群算法 (Multi-Objective Particle Swarm Optimization, MOPSO) 中, 从而阻止算法陷入局部收敛、提高搜索速度和维持种群多样性; Liu 等^[5]将局部搜索融入 MOPSO 算法, 提出一种多目标文化算法; Wickramasinghe 等^[6]提出一种混合的 MOPSO, 其使用 DE 去搜索全局最优解; Elhossini^[7]等将一种改进的 MOPSO 方法加入 EA 中, 提出了三种混合 EA-PSO 算法; 唐立新和王显鹏^[8]将一种多选择算子引入 MOPSO 框架中, 得到一种混合多目标进化算法.

基于此, 本文提出一种多策略差分进化的元胞粒子群算法 (Cellular Multi-Objective Particle Swarm Based on Multi-Strategy Differential Evolution, MPSOCell), 该算法将元胞自动机的理论应用到 PSO 算法中, 研究 PSO 算法

中种群交流结构和信息传递方式,并且将一种多策略差分进化的选择算子引入算法中,从而加强对粒子的扰动.为了更好的控制粒子的飞行过程,提高算法的性能,本文提出一种限制粒子速度的策略.

2 基本粒子群算法

PSO算法是一种新兴的基于种群智能的启发式全局搜索算法,具有易理解、易实现、全局搜索能力强等特点,大量应用于实际工程领域.假设在 D 维搜索空间中,种群大小为 N ,种群中每个粒子 i 具有如下属性:在第 t 代时粒子所处的位置 $\mathbf{x}_i(t) = (x_{i1}, x_{i2}, \dots, x_{id})$,飞行速度 $\mathbf{v}_i(t) = (v_{i1}, v_{i2}, \dots, v_{id})$,粒子自身的历史最佳位置 $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{id})$,种群到达的最优位置 $\mathbf{p}_g = (p_{g1}, p_{g2}, \dots, p_{gd})$.第 $t+1$ 代的粒子根据下面的公式更新自己的速度和位置

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(p_{ij} - x_{ij}(t)) + c_2r_2(p_{gj} - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

其中: w 为权重因子; c_1, c_2 为学习因子; r_1, r_2 为 $[0, 1]$ 的随机数.式(1)由三个部分组成,第一部分为惯性部分,反映了粒子目前的状态;第二部分是认知部分,反映了粒子对自身历史经验的记忆;第三部分是社会部分,反映了粒子间协同合作与知识共享的种群历史经验.

标准粒子群算法的基本流程介绍如下:

Step 1 在初始化范围内,随机生成粒子种群,包括随机位置和速度,并对种群进行评估.

Step 2 定义初始自身历史最优值 \mathbf{p}_i (一般为初始化粒子本身)和种群历史最优值 \mathbf{p}_g .

Step 3 在保证粒子的搜索空间内,对粒子群体进行进化操作,不断的更新粒子的速度和位置.

Step 4 对于每个粒子,将其适应度值与自身经历过最好位置的适应度值和将其历史最优适应度值与种群内所经历的最好位置的适应度值进行比较,如果优秀,则替换,否则不替换.

Step 5 判断是否满足终止条件,满足则停止,否则转向 Step 2.

3 算法介绍

对于 PSO 算法,收敛速度快是它的特点,但是收敛速度过快,容易使算法陷入局部最优解,因此从增强算法的多样性、控制粒子飞行速度和加强对粒子扰动三个方面进行了研究,得到一种多策略差分进化的元胞多目标粒子群算法.如下将对新算法这些方面进行探讨分析.

3.1 元胞自动机与粒子群混合机制

将元胞自动机的理论应用到 PSO 算法中,来研究 PSO 算法中种群的交流结构和信息传递与继承机制.将粒子群看作一种元胞自动机模型,每个粒子只与通过确定的邻居进行信息交流,这样信息在种群中的传播

速度相对较慢,有助于保持种群的多样性和探索搜索空间,充分挖掘每个粒子的局部信息.对于种群中的每个粒子,看成是一个元胞个体,元胞的个数即为种群的大小.将元胞放置于一个空间网格中,一般二维的环形网格结构,如图 1(a)所示,黑色表示元胞,网格拓扑结构首尾相连,组成一个环形空间结构.

每个粒子赋予相同的邻居数,如图 1(b)所示,黑色圆形表示为中心元胞,在此使用 Moore 型邻居结构,即每个元胞拥有八个邻居(灰色的网格),且每个元胞限制只能与其邻居元胞之间进行进化交流,这种方式有利于提高算法的全局寻优的能力.同时,邻近元胞的邻居相互重叠(如图 1(b)中黑色方格),此种重叠的方式提供了一种隐性的迁移机制,有利于最优个体在种群中缓慢地扩散,能更持久地保持种群多样性.在初始化阶段,随机产生粒子,然后每一个粒子被按序无重复地安排到网格中.在整个迭代过程中,粒子保持在网格结构中的相对位置不变.

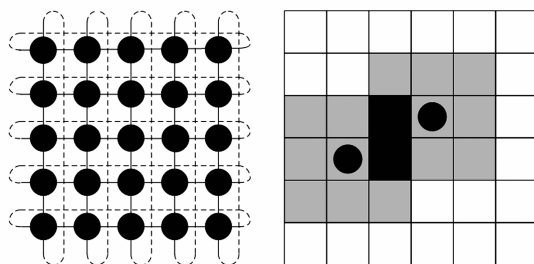


图1 元胞自动机结构示意图

针对粒子群的特点,粒子在每次更新其位置时,需知自身历史最优位置 \mathbf{p}_i 和种群到达的最优位置 \mathbf{p}_g .因此,结合元胞自动机的特点,将粒子自身历史最优位置 \mathbf{p}_i 替换为粒子邻居结构中最优位置 \mathbf{p}_n .假设每个粒子 i 拥有 n 个邻居个体,邻居集合 $S_i(t) = \{x_1(t), x_2(t), \dots, x_i(t), \dots, x_n(t)\}$,则 \mathbf{p}_n 如下所示

Evaluate $S_i(t)$;

rank = Ranking $S_i(t)$;

$\mathbf{p}_n = \text{Sorting}(\text{rank})$;

If ($\mathbf{p}_n(t+1) < \mathbf{p}_n(t)$) then

Update \mathbf{p}_n ;

end if

使用外部文档存储非劣解,因此对于最优个体 \mathbf{p}_g 的选择,步骤如下:先从外部文档中随机选取两个个体,然后通过比较两个个体的拥挤距离,选择拥挤距离大的个体作为 \mathbf{p}_g .

3.2 粒子速度控制策略

PSO算法具有很好的全局搜索性能,但是其局部搜索性能相对较弱,在算法进化后期提高局部搜索能力,

有助于提高算法的收敛性. 如下从提高算法收敛性、加强局部搜索能力和控制粒子飞行速度三个方面进行研究.

(1) 本文采用 1999 年 Clerc 提出的 PSO 收敛因子模型^[9], 实验证明该收敛因子模型能够确保算法的收敛. 收敛因子模型如下所示:

$$v_i(t+1) = k * [wv_i(t) + c_1r_1(p_l - x_i(t)) + c_2r_2(p_g - x_i(t))] \quad (3)$$

$$k = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4 * \varphi} \right|} \quad (4)$$

式中, k 为收敛因子; 如果 $c_1 + c_2 > 4$, 则 $\varphi = c_1 + c_2$; 如果 $c_1 + c_2 < 4$, 则 $\varphi = 0$.

(2) 为了更好地平衡全局搜索和局部搜索, 对影响速度的权重因子 c_1 和 c_2 分别进行线性递增/减. 在算法初期引导算法向 Pareto 前端逼近, 加大“社会认知”部分的影响因子, 减少“邻居认知”部分的影响因子; 而随着进化的不断进行, 种群中非支配个体更加, 为了获得更好的分布性, 算法将加大“邻居认知”部分的影响因子, 减少“社会认知”部分的影响因子. 因此, 影响因子 c_1 、 c_2 的公式为

$$c_1 = (c_{1\max} - c_{1\min}) * \frac{\text{gen}}{\text{MAXGEN}} + c_{1\min} \quad (5)$$

$$c_2 = (c_{2\max} - c_{2\min}) * \frac{\text{MAXGEN} - \text{gen}}{\text{MAXGEN}} + c_{2\min} \quad (6)$$

(3) 粒子在飞行过程中如果飞行速度过快, 将导致算法陷入局部收敛, 而有效地控制粒子的飞行速度有利于提高算法的性能^[10]. 基于此, 对粒子采用限速操作, 公式如下:

$$\begin{cases} v_{i,j}(t) = \frac{2}{3} \Delta, \text{if}(v_{i,j}(t) > \Delta) \\ v_{i,j}(t) = -\frac{1}{3} \Delta, \text{if}(v_{i,j}(t) < -\Delta) \\ v_{i,j}(t) = v_{i,j}(t), \text{else} \end{cases} \quad (7)$$

式(7)中, $\Delta = u_j - l_j$, u_j 、 l_j 分别为决策变量的上下约束值.

3.3 多策略差分进化交叉算子

借鉴在求解单目标问题时, 多策略差分^[11-14]有很好的效果, 基于此提出一种多差分协同进化的选择算子. 该策略一者能够增强对粒子群的扰动; 二者能够加强算法的局部寻优的能力, 提高算法的多样性.

差分进化(Differential Evolution, DE)存在多种进化模式, 其一般表示形式为 DE/ $x/y/z$, 其中 x 表示差分算子的基准个体的选择, 一般有随机选取(rand)还是当前种群最优个体选取(best); y 表示参与重组的差异个体的选择个数; z 表示重组选择的方式. 下面将给出 5 种常见的 DE 进化模式:

(1) Rand/1/bin 模式, 其重组公式为

$$V_{i,G} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G}) \quad (8)$$

(2) Rand/2/bin 模式, 其重组公式为

$$V_{i,G} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G}) + F(X_{r_4,G} - X_{r_5,G}) \quad (9)$$

(3) Best/1/bin 模式, 其重组公式为

$$V_{i,G} = X_{\text{best},G} + F(X_{r_1,G} - X_{r_2,G}) \quad (10)$$

(4) Best/2/bin 模式, 其重组公式为

$$V_{i,G} = X_{\text{best},G} + F(X_{r_1,G} - X_{r_2,G}) + F(X_{r_3,G} - X_{r_4,G}) \quad (11)$$

(5) Rand-to-best/1/bin 模式, 其重组公式为

$$V_{i,G} = X_{i,G} + F(X_{\text{best},G} - X_{i,G}) + F(X_{r_1,G} - X_{r_2,G}) \quad (12)$$

其中: $V_{i,G}$ 代表基准个体 i , G 代表进化代数, $X_{\text{best},G}$ 表示当代最优秀个体, $X_{r_1,G}$ 、 $X_{r_2,G}$ 、 $X_{r_3,G}$ 、 $X_{r_4,G}$ 、 $X_{r_5,G}$ 和 $X_{i,G}$ 代表随机选择的个体, F 表示差分控制参数. 上面的 5 种进化模式有各自的特点, 对各种不同进化模式进行分析如下:

DE/rand/1/bin 模式和 DE/rand/2/bin 模式, 其优点是全局探索能力, 容易跳出局部最优解, 但是收敛速度较慢; DE/best/1/bin 模式 DE/best/2/bin 模式, 收敛速度快, 但是全局探索的能力相当较弱, 易于陷入局部收敛; DE/rand-to-best/1/bin 模式, 能很好全局探索和局部寻优能很好保持平衡, 但鲁棒性不好. 鉴于上述各种 DE 进化策略各自有其特有的优势和不足, 不能一概而论哪种进化模式更优秀, 哪种进化模式相对较弱, 但是各类进化模式存在共同的特性, 即采取的重组方式都是基准个体与其差异个体之间的线性组合. 本文将 DE/rand/1/bin、DE/best/1/bin 和 DE/rand-to-best/1/bin 三种进化策略引入至交叉算子中, 让三种进化策略相互配合, 相互促进, 协同进化.

结合元胞空间结构的特点, 在选择进化个体时也有其特点. 本文采用八邻居结构, 分别对三种模式所需个体进行选择. 对于 DE/rand/1/bin 进化模式, 须选择三个个体, 分别为一个中心元胞个体和两个邻居个体; 对于 DE/best/1/bin 进化模式, 三个个体分别来源为: 两个差分个体来自邻居个体, 最优个体从外部存档中随机选择一个个体; 对于 DE/rand-to-best/1/bin 进化模式, 四个个体分别为: 一个中心个体, 两个邻居个体, 最优个体的选择同 DE/best/1/bin.

基于这种多策略进化差分的交叉算子, 使得三种进化策略都有相同的概率产生新的粒子, 这种方式一方面能够对种群中的粒子产生扰动, 增强种群多样性, 另一方面在执行某一进化策略时必然存在其他进化策略生成的个体参与进化, 是一种隐式协同进行. 在时间复杂度上, 这个选择算子和一般的 DE 算子相同.

3.4 约束处理方法

对于无约束的问题, 其可行解的个体之间的优劣比

较,首先将种群中的个体进行非劣排序,每个个体被赋予秩,根据秩的大小进行分层,秩值小的个体优于秩值大的个体.对于同一层的个体,则根据其拥挤距离大小进行判断,拥挤距离大的个体优于拥挤距离小的个体.

而对于含有约束条件的多目标优化问题,则需要对违反约束的不可行解进行处理,常见的处理方式有罚函数法和多目标法^[15].本文采用 Deb^[16]的多目标法处理违反约束的个体.具体如下:

(1)当在两个比较的个体中,一个个体为可行解,另外一个个体为不可行解时,选择可行解;

(2)当两个比较的个体均为可行解时,选择非支配的个体占优;

(3)当两个比较的个体均为不可行解时,选择违反约束条件程度小的个体;

3.5 算法流程

综合 3.1~3.3 节在经典的元胞遗传算法上三方面改进,MPSCell 算法的算法流程如下所示:

Step 1 对粒子群进行随机初始化,将粒子分布于二维网格结构中,并计算各粒子对应的目标向量,将其中的非劣解放入外部文档.

Step 2 评估每个粒子的邻居粒子,确定粒子的初始邻居结构最优值 p_n 和种群最优值 p_g .

Step 3 根据新粒子飞行策略,改变粒子的速度和位置.对生成的粒子群进行一定概率的扰动,进行多策略差分进化的选择操作,其中包括对粒子的选择和交叉操作.

Step 4 对子代粒子群进行评估,替换操作,将其中的非劣解存入外部文档中,并更新 p_n 和 p_g .

Step 5 判断是否满足终止条件,满足则输出非支配解集,否则转向 Step 2.

4 仿真实验及其分析

4.1 测试函数

为了验证 MPSOCell 算法的在多目标优化问题上的求解性能,选取 WFG 和 DTLZ 系列基准测试函数;为了求证在解决多目标约束问题上的能力,选取了 Osyczka2、Tanaka、ConstrEx、Srinivas 测试函数.将其与包括 OMOPSO^[17]、SMP SO^[10]、NSGA-II^[18]、SPEA2^[19] 和 MOCcell^[20] 五种目前比较优异的算法进行比较分析.

4.2 算法性能测试、对比与分析

为了评估算法的性能,采用三个性能指标:覆盖性指标 HV(Hypervolume)^[21]、分布指标 SP(Spread)^[18]和收敛性指标 IGD(Inverted Generational Distance)^[22]三个性能指标进行度量.

算法的参数设置:所有算法的初始种群大小 N 均设置为 100,最大函数评价次数均设置为 250 代.其中,MPSCell 的外部文档大小设置为 100,反馈个体数目设置为 20,其中 $w, r_1, r_2 \in (0, 1), c_{1\max} = 3.0, c_{1\min} = 1.5, c_{2\max} = 3.0, c_{2\min} = 1.5$.

对于差分选择算子的概率 CR 的选择,文献[23]认为 CR 值应在[0.3,0.9]范围,且 CR 越大,每一维选择变异矢量的概率越大,从而使基础矢量相对于目标矢量受到更大的扰动,降低算法的收敛速度,多样性更好,反之,小的 CR 概率容易使种群快速失去多样性.为了增大对粒子的扰动,在实验反复验算下,本文选用的 DE 进化参数设置为:CR = 0.8, F = 0.5.对于其他的算法中的控制参数设置都源于原文献^[10,17~20].这六种算法分布对测试函数进行 50 次独立运行计算,统计 HV、SP 和 IGD 的平均值(括号外)及标准差(括号内).对于计算的结果,如表 1~3 所示.

表 1 IGD 性能指标的计算结果

IGD	MPSOCell	NSGA-II	SPEA2	MOCcell	SMP SO	OMOPSO
WFG1	2.80E-04(9.8E-05)	7.24E-04(5.2E-04)	1.62E-03(5.2E-04)	1.13E-03(5.6E-04)	4.04E-03(1.4E-04)	3.80E-03(3.4E-04)
WFG2	5.73E-05(6.5E-07)	1.41E-03(1.1E-03)	1.54E-03(1.1E-03)	1.45E-03(1.1E-03)	8.97E-05(1.2E-05)	6.88E-05(3.5E-06)
WFG3	6.14E-05(4.4E-07)	8.80E-05(4.8E-06)	6.74E-05(1.8E-06)	6.86E-05(4.1E-06)	6.69E-05(1.3E-06)	6.35E-05(6.8E-07)
WFG4	1.20E-04(1.8E-05)	1.05E-04(4.8E-06)	8.82E-05(2.3E-06)	8.19E-05(2.1E-06)	2.68E-04(3.0E-05)	2.27E-04(2.6E-05)
WFG5	4.04E-05(2.1E-06)	5.40E-05(3.9E-06)	4.46E-05(2.5E-06)	3.83E-05(2.1E-06)	4.11E-05(1.8E-06)	4.08E-05(1.9E-06)
WFG6	7.16E-05(9.5E-07)	1.73E-04(1.0E-04)	1.47E-04(7.7E-05)	1.36E-04(8.1E-05)	7.77E-05(2.1E-06)	7.44E-05(1.2E-06)
WFG7	5.72E-05(3.9E-07)	7.97E-05(4.6E-06)	6.60E-05(1.4E-06)	5.93E-05(7.6E-07)	6.10E-05(1.1E-06)	5.87E-05(7.3E-07)
WFG8	3.22E-03(2.3E-04)	2.84E-03(7.1E-04)	3.13E-03(4.7E-04)	2.72E-03(8.6E-04)	2.90E-03(1.9E-04)	3.33E-03(1.8E-04)
WFG9	1.12E-04(4.0E-06)	1.51E-04(9.7E-06)	1.27E-04(6.4E-06)	1.34E-04(2.0E-05)	1.42E-04(4.4E-06)	1.31E-04(3.3E-06)
DTLZ1	9.99E-02(6.7E-02)	2.89E-03(3.6E-03)	9.65E-04(1.1E-03)	4.01E-02(2.4E-02)	9.29E-04(4.0E-05)	3.91E-01(1.3E-01)
DTLZ2	7.23E-04(2.1E-05)	7.89E-04(3.8E-05)	5.89E-04(6.5E-06)	8.03E-04(4.3E-05)	7.83E-04(2.4E-05)	7.61E-04(2.3E-05)
DTLZ3	2.07E+00(7.0E-01)	1.12E-01(6.4E-02)	1.40E-01(7.3E-02)	7.78E-01(2.5E-01)	3.54E-03(3.1E-03)	2.91E+00(6.5E-01)
DTLZ4	8.81E-04(1.3E-03)	7.46E-04(3.2E-05)	2.56E-03(3.2E-03)	1.20E-03(1.9E-03)	1.15E-03(6.7E-04)	9.62E-04(5.4E-04)
DTLZ5	6.26E-05(1.0E-06)	9.11E-05(4.7E-06)	6.99E-05(2.1E-06)	7.36E-05(4.6E-06)	6.47E-05(2.0E-06)	6.66E-05(1.8E-06)
DTLZ6	1.14E-03(3.7E-03)	1.20E-02(1.1E-03)	1.10E-02(7.5E-04)	4.58E-02(2.3E-03)	2.77E-04(1.6E-03)	5.49E-05(1.4E-06)
DTLZ7	8.90E-04(1.1E-04)	7.52E-04(6.7E-05)	7.85E-04(1.0E-03)	9.76E-04(8.6E-05)	8.57E-04(1.0E-04)	8.54E-04(9.2E-05)
Osyczka2	2.99E-03(1.5E-03)	6.31E-04(5.1E-04)	6.85E-04(9.6E-04)	1.95E-03(1.3E-03)	2.33E-03(1.0E-03)	2.09E-03(9.9E-04)
Tanaka	6.38E-05(2.3E-06)	1.01E-04(1.0E-05)	1.22E-04(2.6E-05)	8.99E-05(1.6E-05)	1.40E-04(1.2E-05)	1.08E-04(1.0E-05)
ConstrEx	4.57E-05(8.7E-07)	7.02E-05(5.6E-06)	4.69E-05(8.2E-07)	1.43E-04(1.1E-05)	4.66E-05(9.8E-07)	4.66E-05(1.1E-06)
Srinivas	3.73E-05(9.5E-07)	5.53E-05(3.7E-06)	3.73E-05(2.6E-07)	4.05E-05(7.7E-07)	3.74E-05(3.5E-07)	3.73E-05(4.2E-07)

表 2 SP 性能指标的计算结果

SP	MPSOCell	NSGA-II	SPEA2	MOCcell	SMPSO	OMOPSO
WFG1	9.29E-01(1.5E-01)	7.20E-01(3.5E-02)	6.71E-01(5.8E-02)	8.54E-01(2.0E-01)	1.02E+00(4.7E-02)	1.10E+00(1.0E-01)
WFG2	7.59E-01(8.6E-04)	7.92E-01(1.1E-02)	7.62E-01(5.0E-03)	7.66E-01(6.6E-03)	8.03E-01(2.0E-02)	7.73E-01(6.7E-03)
WFG3	3.47E-02(8.6E-03)	3.68E-01(3.3E-02)	1.54E-01(1.2E-02)	7.28E-02(1.1E-02)	7.00E-02(8.1E-03)	5.58E-02(6.6E-03)
WFG4	2.59E-01(3.9E-02)	3.76E-01(3.1E-02)	2.49E-01(2.0E-02)	1.39E-01(1.6E-02)	4.66E-01(4.0E-02)	4.11E-01(3.8E-02)
WFG5	1.50E-01(1.5E-02)	4.16E-01(3.0E-02)	2.73E-01(1.6E-02)	1.36E-01(1.3E-02)	1.57E-01(1.8E-02)	1.52E-01(1.6E-02)
WFG6	1.14E-01(1.5E-02)	3.88E-01(3.1E-02)	2.53E-01(2.0E-02)	1.71E-01(4.9E-02)	1.51E-01(1.8E-02)	1.27E-01(1.4E-02)
WFG7	1.20E-01(1.5E-02)	3.78E-01(2.9E-02)	2.50E-01(1.8E-02)	1.27E-01(1.4E-02)	1.50E-01(1.2E-02)	1.33E-01(1.3E-02)
WFG8	5.45E-01(8.2E-02)	6.30E-01(5.0E-02)	6.30E-01(7.1E-02)	6.17E-01(9.0E-02)	7.01E-01(6.3E-02)	5.66E-01(6.6E-02)
WFG9	1.64E-01(1.5E-02)	3.93E-01(2.4E-02)	2.77E-01(1.7E-02)	1.78E-01(1.5E-02)	2.10E-01(1.8E-02)	1.96E-01(1.5E-02)
DTLZ1	7.32E-01(7.0E-02)	9.39E-01(2.4E-01)	8.49E-01(2.7E-01)	1.15E+00(1.2E-01)	6.72E-01(4.2E-02)	7.09E-01(5.9E-02)
DTLZ2	6.28E-01(4.1E-02)	7.12E-01(4.0E-02)	5.35E-01(2.8E-02)	6.91E-01(4.8E-02)	6.34E-01(4.2E-02)	6.22E-01(3.5E-02)
DTLZ3	6.88E-01(6.5E-02)	1.04E+00(1.3E-01)	1.25E+00(1.0E-01)	1.01E+00(6.4E-02)	8.31E-01(3.3E-01)	7.28E-01(9.6E-02)
DTLZ4	6.36E-01(6.6E-02)	6.65E-01(4.0E-02)	5.23E-01(1.9E-01)	6.53E-01(8.5E-02)	6.82E-01(8.1E-02)	6.73E-01(9.6E-02)
DTLZ5	1.11E-01(1.5E-02)	4.39E-01(3.6E-02)	2.39E-01(4.3E-02)	2.74E-01(8.1E-02)	1.67E-01(5.6E-02)	2.02E-01(4.3E-02)
DTLZ6	2.62E-01(2.2E-01)	8.18E-01(5.8E-02)	5.78E-01(3.4E-02)	7.41E-01(4.3E-02)	1.43E-01(9.5E-02)	1.29E-01(2.9E-02)
DTLZ7	7.06E-01(4.2E-02)	7.36E-01(4.6E-02)	5.88E-01(4.7E-02)	7.14E-01(4.1E-02)	7.05E-01(4.4E-02)	7.00E-01(4.9E-02)
Oszycza2	7.30E-01(1.1E-01)	5.63E-01(7.1E-02)	4.35E-01(1.1E-01)	7.12E-01(1.2E-01)	9.41E-01(1.4E-01)	9.31E-01(1.7E-01)
Tanaka	6.20E-01(2.1E-02)	8.07E-01(2.8E-02)	8.11E-01(3.5E-02)	7.51E-01(3.7E-02)	8.03E-01(3.2E-02)	7.24E-01(3.6E-02)
ConstrEx	8.07E-02(1.4E-02)	4.66E-01(4.1E-02)	1.30E-01(1.1E-02)	5.14E-01(1.9E-02)	1.20E-01(1.1E-02)	1.22E-01(1.3E-02)
Srinivas	6.92E-02(1.1E-02)	4.04E-01(3.0E-02)	7.03E-02(1.1E-02)	1.69E-01(1.3E-02)	8.38E-02(1.1E-02)	7.89E-02(1.2E-02)

表 3 HV 性能指标的计算结果

HV	MPSOCell	NSGA-II	SPEA2	MOCcell	SMPSO	OMOPSO
WFG1	5.94E-01(1.8E-02)	5.30E-01(7.4E-02)	4.05E-01(8.8E-02)	4.47E-01(8.6E-02)	1.20E-01(1.0E-02)	1.33E-01(2.3E-02)
WFG2	5.58E-01(2.7E-05)	5.54E-01(2.3E-03)	5.54E-01(2.3E-03)	5.53E-01(2.4E-03)	5.54E-01(6.6E-04)	5.56E-01(2.8E-04)
WFG3	4.95E-01(3.5E-05)	4.92E-01(5.0E-04)	4.93E-01(4.9E-04)	4.92E-01(8.8E-04)	4.93E-01(3.1E-04)	4.94E-01(2.0E-04)
WFG4	2.06E-01(9.8E-04)	2.09E-01(3.3E-04)	2.10E-01(2.9E-04)	2.09E-01(4.5E-04)	1.95E-01(1.6E-03)	1.98E-01(1.1E-03)
WFG5	2.12E-01(2.0E-05)	2.10E-01(2.6E-04)	2.11E-01(1.1E-04)	2.12E-01(6.7E-05)	2.12E-01(1.3E-04)	2.12E-01(3.3E-05)
WFG6	2.11E-01(3.7E-05)	1.99E-01(9.7E-03)	2.01E-01(7.4E-03)	2.02E-01(7.3E-03)	2.09E-01(3.1E-04)	2.10E-01(1.7E-04)
WFG7	2.11E-01(2.8E-05)	2.09E-01(3.0E-04)	2.10E-01(1.6E-04)	2.10E-01(2.5E-04)	2.09E-01(2.3E-04)	2.10E-01(1.3E-04)
WFG8	2.12E-01(1.7E-03)	2.14E-01(1.3E-02)	2.10E-01(5.5E-03)	2.11E-01(1.6E-02)	2.11E-01(1.9E-03)	2.09E-01(2.0E-03)
WFG9	2.18E-01(7.6E-04)	2.15E-01(1.3E-03)	2.16E-01(1.1E-03)	2.14E-01(2.1E-03)	2.13E-01(3.9E-04)	2.14E-01(3.2E-04)
DTLZ1	4.65E-03(1.9E-02)	5.93E-01(2.6E-01)	7.49E-01(9.7E-02)	1.40E-02(8.0E-02)	7.39E-01(5.7E-03)	0.00E+00(0.0E+00)
DTLZ2	4.02E-01(5.1E-03)	4.03E-01(5.7E-03)	4.32E-01(1.7E-03)	3.92E-01(6.9E-03)	3.79E-01(6.8E-03)	3.84E-01(4.4E-03)
DTLZ3	—	—	—	—	3.45E-01(9.0E-02)	—
DTLZ4	3.99E-01(5.7E-02)	4.04E-01(5.4E-03)	3.49E-01(1.3E-01)	3.80E-01(8.2E-02)	3.85E-01(1.4E-02)	3.86E-01(1.1E-02)
DTLZ5	9.35E-02(2.5E-05)	9.22E-02(1.9E-04)	9.26E-02(1.5E-04)	9.22E-02(2.5E-04)	9.32E-02(9.6E-05)	9.29E-02(1.3E-04)
DTLZ6	8.44E-02(2.6E-02)	0.00E+00(0.0E+00)	0.00E+00(0.0E+00)	0.00E+00(0.0E+00)	9.16E-02(1.3E-02)	9.35E-02(5.3E-05)
DTLZ7	2.85E-01(8.6E-03)	3.07E-01(4.2E-03)	3.17E-01(9.9E-03)	2.40E-01(1.4E-02)	3.03E-01(6.5E-03)	2.87E-01(6.7E-03)
Oszycza2	6.57E-01(1.0E-01)	7.49E-01(4.4E-03)	7.39E-01(4.9E-02)	7.16E-01(6.0E-02)	6.82E-01(4.6E-02)	7.06E-01(7.3E-03)
Tanaka	3.05E-01(1.8E-04)	3.04E-01(3.9E-04)	3.04E-01(6.3E-04)	3.05E-01(4.0E-04)	3.00E-01(8.9E-04)	3.02E-01(5.7E-04)
ConstrEx	7.78E-01(5.7E-05)	7.75E-01(3.2E-04)	7.76E-01(2.0E-04)	7.76E-01(2.3E-04)	7.76E-01(2.0E-04)	7.76E-01(1.7E-04)
Srinivas	5.36E-01(4.9E-05)	5.33E-01(3.8E-04)	5.36E-01(6.1E-05)	5.35E-01(1.9E-04)	5.36E-01(6.2E-05)	5.36E-01(7.3E-05)

从表 1 中 IGD 性能指标可得, MPSOCell 获得 9 个最优值, 次优值为 3 个; NSGA-II 获得 3 个最优值, 3 个次优值; SPEA2 获得 2 个最优值, 5 个次优值; MOCcell 获得 3 个最优值, 1 个次优值; SMPSO 获得 2 个最优值, 3 个次优值; OMOPSO 获得 1 个最优值, 5 个次优值. 实验数据说明, 在收敛性方面, MPSOCell 总体上相对于 NSGA-II、SPEA2 和 MOCcell 三种进化算法有较明显的优势, 尤其在解决 WFG 系列问题上, 有较为明显的优势. MPSOCell 相对于 SMPSO 和 OMOPSO 两种粒子群算法在解决 WFG 系列和含有约束问题上占有绝对的优势, 在解决 DTLZ 系列问题上, SMPSO 略微占优. MPSOCell 收敛性方面的优势, 主要原因如下: 首先是 PSO 算法本身全

局搜索能力较强. 其次, 元胞拓扑结构可以降低算法的选择压, 能够很好地平衡全局搜索和局部寻优, 有利于种群跳出局部最优解. 第三, 使用 Clerc^[9] 提出的 PSO 收敛因子模型同时, 考虑了在算法初期种群最优解更有利算法进化, 增大了种群最优解的影响因子, 在后期为了加强算法的局部搜索, 逐渐提高邻居最优个体的作用, 增大了邻居最优个体的影响因子, 这种策略更有利于提高算法收敛性.

从表 2 中 SPREAD 性能指标可得, MPSOCell 获得 11 个最优值, 次优值为 2 个; NSGA-II 仅获得 3 个次优值; SPEA2 获得 5 个最优值, 3 个次优值; MOCcell 获得 2 个最优值, 2 个次优值; SMPSO 获得 1 个最优值, 3 个次优值;

OMOPSO 获得 1 个最优值, 8 个次优值. 由实验数据说明, MPSOCell 算法整体上相对于其他 5 种算法, 多样性有很明显的优势, 尤其在解决 WFG 系列问题和带约束问题上, 能很好的保持种群的多样性. 在解决 DTLZ 系列问题时, SPEA2 算法比 MPSOCell 略微占优. 文献[24]大量研究和实验发现差分进化算法在维护种群多样性及搜索能力方面功能较强, MPSOCell 算法采用 3 差分策略协同进化更有利于保持算法的多样性, 且能够加大对粒子的扰动. 采用控制粒子飞行速度机制, 有利于粒子进行全局搜索, 保持种群的多样性. 最后, 元胞拓扑结构, 有利于最优解在种群中缓慢的扩散, 加强了算法的局部寻优能力.

从表 3 中 HV 性能指标可得, MPSOCell 获得最优值为 10 个, 次优值为 3 个; NSGA-II 最优值获得 3 个, 次优值为 4 个; SPEA2 获得最优值为 4 个, 次优值为 4 个; MOCeII 仅获得 1 个次优值; SMPSO 获得最优值为 2 个, 次优值为 3 个; OMOPSO 获得最优值为 1 个, 次优值为 4

个. 由实验数据说明, 在覆盖率方面, MPSOCell 表现最佳, 在解决 WFG 系列问题和带约束问题上有很大的优势. 在解决 DTLZ 系列问题时, SPEA2 覆盖性最优.

为了能够更加直观地说明问题, 各算法在解决 WFG2 问题时的非劣解集的情况, 如图 2 所示. 所有算法的非劣解集来源于运行 50 次中, HV 指标最大值的一次结果. 在解决 WFG2 问题时, MPSOCell 和 OMOPSO 具有较好的收敛性和均匀性保持能力, SMPSO 得到的非支配解集分布不够均匀, NSGA-II、SPEA2 和 MOCeII 未能完全找到所有的 Pareto 最优解. 图 2 与表 1~3 中所对应的结果相符.

从上表和图的结果对比可知, 在 2 目标测试函数 WFG、3 目标测试函数 DTLZ 和 4 个带约束的测试函数的求解效果上, MPSOCell 算法相对 NSGA-II、SPEA2、MOCeII、SMPSO 和 OMOPSO 算法具有明显优势, 不仅能达到最好的收敛性和多样性, 还能获得分布更均匀、覆盖范围更广的近似 Pareto 前端.

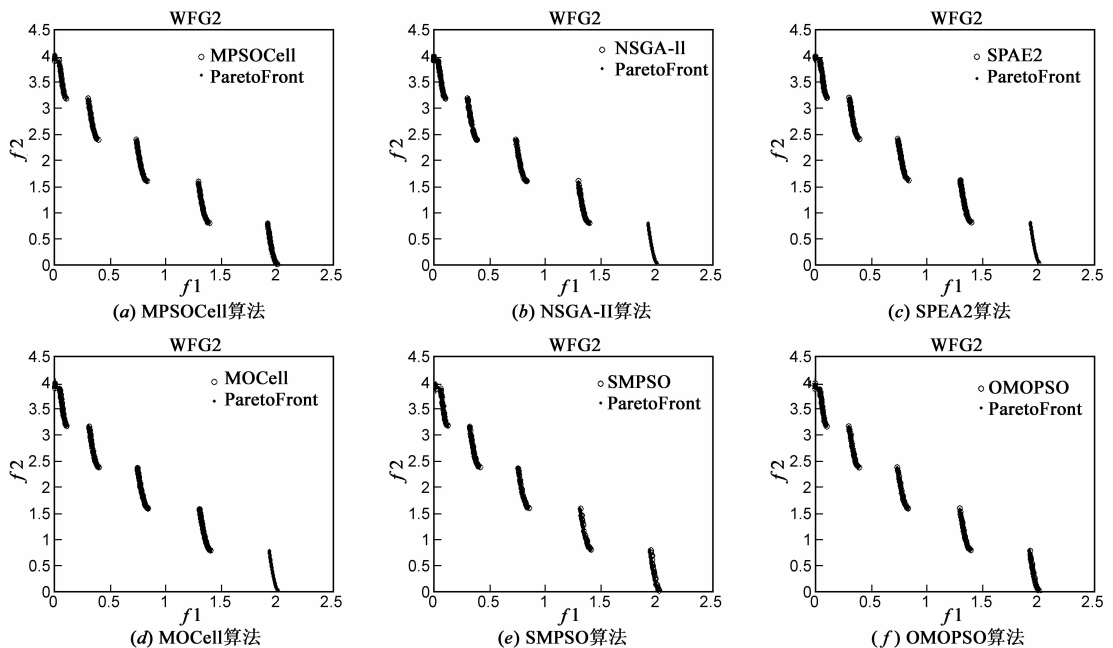


图2 WFG2问题的Pareto最优前沿面比较

5 总结

本文提出一种多策略差分进化的元胞多目标粒子群算法 MPSOCell. 首先, 分析粒子群算法基本原理, 将元胞自动机理论融入粒子群算法中, 对种群中每个粒子分配邻居, 从而隐性的将种群划分为一个个小种群; 其次, 针对粒子群的特点, 将个体“自身”历史最优值变为“小种群”历史最优值, 这种方式促进算法局部寻优能力; 第三, 为了避免粒子在进化过程中飞行速度过快, 定义了一种速度控制策略, 从而加强算法的全局寻

优; 最后, 为了增强对粒子的扰动, 引入一种多策略差分进化的选择算子, 同时该算子对算法的多样性也有一定程度的提高. 通过 20 个基准多目标测试函数进行测试, 实验结果表明, 提出的 MPSOCell 算法在整体性能相对于 3 种进化算法 (NSGA-II、SPEA2、MOCeII) 和 2 种多目标粒子群算法 (OMOPSO、SMPSO), 有更好的收敛性和多样性, 而且在解的覆盖率上得到一定程度的提高. 此外, 对于 MPSOCell 算法如何将其运用至实际工程应用中, 将是进一步的方向.

参考文献

- [1] LI X D. A non-dominated sorting particle swarm optimizer for multiobjective optimization [A]. James A F. Genetic and Evolutionary Computation—GECCO [C]. Berlin: Springer, 2003. 37 – 48.
- [2] SRINIVASAN D, SEOW T H. Particle swarm inspired evolutionary algorithm (ps-ea) for multi-criteria optimization problems [A]. Ajith A. Evolutionary Computation [C]. London: Springer, 2003. 2292 – 2297.
- [3] SANTANA-QUINTERO L V, RAMIREZ N, et al. A multi-objective particle swarm optimizer hybridized with scatter search [A]. Alexander G. MICAI2006: Advances in Artificial Intelligence [C]. Berlin Heidelberg: Springer, 2006. 294 – 304.
- [4] TSOU C S, FANG H H, CHANG H H, et al. An improved particle swarm Pareto optimizer with local search and clustering [A]. Wang T D. Proceedings of the 6th International Conference on Simulated Evolution and Learning [C]. Berlin Heidelberg: Springer, 2006. 400 – 407.
- [5] LIUD S, TAN K C, GOH C K, HO W K. A multi-objective memetic algorithm based on particle swarm optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2007, 37(1): 42 – 50.
- [6] WICKRAMASINGHE U, LI X D. Choosing leaders for multi-objective PSO algorithms using differential evolution [A]. Green D. Proceedings of the 7th International Conference on Simulated Evolution and Learning [C]. Australia: Springer, 2008. 249 – 258.
- [7] ELHOSSINI A, AREIBI S, DONY R. Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization [J]. Evolutionary Computation, 2010, 18(1): 127 – 156.
- [8] TANG L X, WANG X P. A hybrid multi-objective evolutionary algorithm for multi-objective optimization problems [J]. IEEE Transactions on Evolutionary Computation, 2013, 17(1): 20 – 45.
- [9] CLERC M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization [A]. Institute of Electrical and Electronics Engineers Computer Society: Evolutionary Computation [C]. Washington DC: Elsevier, 1999. 1951 – 1957.
- [10] NEBRO A J, DURILLO J J, GARCÍA-NIETO J, COELLO C A, et al. SMP SO: A new PSO-based metaheuristic for multiobjective optimization [A]. Institute of Electrical and Electronics Engineers Computer Society: IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making [C]. Nashville, TN: Elsevier, 2009. 66 – 73.
- [11] QIN A K, SUGANTHAN P N. Self-adaptive differential evolution algorithm for numerical optimization [A]. Institute of Electrical and Electronics Engineers Computer Society: IEEE Congress on Evolutionary Computation (CEC 2005) [C]. Berlin Heidelberg: Elsevier, 2005. 1785 – 1791.
- [12] HUANG V L, QIN A K, SUSANTHAN P N. Self-adaptive differential evolution algorithm for constrained real-parameter optimization [A]. Institute of Electrical and Electronics Engineers Computer Society: 2006 IEEE Congress on Evolutionary Computation [C]. Vancouver, BC: Elsevier, 2006. 17 – 24.
- [13] QIN A. K, HUANG V L, SUSANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398 – 417.
- [14] 贺毅朝, 王熙照, 刘坤起, 王彦祺. 差分演化的收敛性分析与算法改进 [J]. 软件学报, 2010, 21(5): 875-885.
HE Yi-chao, WANG Xi-zhao, LIU Kun-qi, et al. Convergent analysis and algorithmic improvement of differential evolution [J]. Journal of Software, 2010, 21(5): 875 – 885. (in Chinese)
- [15] 王勇, 蔡自兴, 周育人, 肖赤心. 约束优化进化算法 [J]. 软件学报, 2009, 20(1): 11 – 29.
WANG Yong, CAI Zi-xing, ZHOU Yu-ren, XIAO Chi-xin. Constrained optimization evolutionary algorithms [J]. Journal of Software, 2009, 20(1): 11 – 29. (in Chinese)
- [16] DEB K. An efficient constraint handling method for genetic algorithms [J]. Computation Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311 – 338.
- [17] MARGARITA R S, COELLO C A. Improving PSO-Based multi-objective optimization using crowding, mutation and E-Dominance [A]. Zitzler E. Third International Conference on Evolutionary Multi-Criterion Optimization [C]. Berlin Heidelberg: Springer, 2005. 505 – 519.
- [18] DEB K, PRATAP A, MEYARIVAN T. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182 – 197.
- [19] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257 – 271.
- [20] NEBRO A J, DURILLO J J, LUNA F, et al. MOCe ll: A cellular genetic algorithm for multi-objective optimization [J]. International Journal of Intelligent Systems, 2009, 24(7): 726 – 746.
- [21] VAN VELD HUIZEN D A, LAMONT G B. Multi-objective Evolutionary Algorithm Research: A History and Analysis [R]. USA: Department of Electrical and Computer Engineering, the Air Force Institute of Technology, 1998.
- [22] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach [J]. IEEE Transactions on Evolutionary Computation,

1999,3(4):257-271.

- [23] NOBAKHTI A, WANG H. A simple self-adaptive differential evolution algorithm with application on the ALSTOM gasifier [J]. *Applied Soft Computing*, 2008, 8(1): 350-370.
- [24] 孟红云, 张小华, 刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. *计算机学报*, 2008, 31(2): 228-

235.

MENG Hong-yun, ZHANG Xiao-hua, LIU San-yang. A differential evolution based on double population for constrained multi-objective optimization problem [J]. *Chinese Journal of Computers*, 2008, 31(2): 228-235. (in Chinese)

作者简介



朱大林 男, 1957年10月生于湖北宜昌, 三峡大学机械与动力学院教授, 主要研究方向: 系统优化设计、机械及结构可靠性.

E-mail: dlzhu@ctgu.edu.cn



詹腾 男, 1988年2月生于湖北黄冈, 三峡大学硕士研究生, 主要研究方向: 智能优化算法, 工程优化设计等.

E-mail: zhanteng1988@gmail.com



张屹(通信作者) 男, 1976年12月生于甘肃兰州, 三峡大学机械与动力学院副教授, 博士, 主要研究方向: 智能制造、系统优化设计等.

E-mail: jxzhangyi1976@126.com



郑小东 男, 1989年6月生于湖北孝感, 三峡大学硕士研究生, 主要研究方向智能优化算法.