

三态协调搜索多目标粒子群优化算法

王学武, 薛立卡, 顾幸生

(华东理工大学 化工过程先进控制和优化技术教育部重点实验室, 上海 200237)

摘要: 提出一种三态协调搜索多目标粒子群优化算法. 该算法提出的三态指导粒子选择策略可以很好地协调算法的局部和全局搜索能力, 且算法改进了传统的外部档案保存机制, 同时引入3种突变因子, 使获得的非劣解具有更好的分散性. 通过对标准测试函数的求解, 并与其他经典多目标优化算法比较, 表明了新算法在收敛性和多样性方面均有较大的优越性. 最后分析了区域划分系数对所提出算法性能的影响.

关键词: 多目标优化; 粒子群优化; 指导粒子选择策略; 搜索能力; 外部档案

中图分类号: TP301

文献标志码: A

Multi-objective particle swarm optimization algorithm based on three status coordinating searching

WANG Xue-wu, XUE Li-ka, GU Xing-sheng

(Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China. Correspondent: WANG Xue-wu, E-mail: wangxuew@ecust.edu.cn)

Abstract: A multi-objective particle swarm optimization algorithm based on three status coordinating searching(TC-MOPSO) is presented. The three status strategy proposed for choosing guides is useful to coordinate local and global search capability. The traditional external archives update mechanism is improved and three kinds of mutation factors are introduced, which contribute to making the Pareto solutions have a better distribution. By solving several standard test functions and comparing with three classical multi-objective optimization algorithms, it is proved that the new algorithm has high competition in terms of convergence and diversity metrics. Finally, the influence of the regionalism coefficient on the performance of the proposed algorithm is analyzed.

Keywords: multi-objective optimization; particle swarm optimization; strategy for choosing guides; search capability; external archives

0 引言

受鸟群捕食活动规律性的启发, Kennedy等^[1]于1995年首次提出了粒子群优化(PSO)算法. 粒子群算法由于其具有收敛速度快和算法结构简单等优点, 自提出以来, 不仅在单目标优化问题中得到了广泛应用, 而且在多目标优化中也受到诸多学者的青睐. 粒子群算法的基础是粒子间的相互合作, 粒子间通过信息的交流产生相应的自组织行为和启发式搜索, 而信息的交流传播方式即拓扑结构, 直接影响到了算法的寻优能力和收敛性^[2]. Kennedy^[3]认为, 在PSO算法中拥有较大的邻居数量在优化简单问题时有较好的效果, 而有较小的邻居数量则在优化复杂问题时有较好

的效果.

在求解多目标优化问题时主要有两个问题^[4]: 一是如何形成适应度函数及选择方案, 使搜索结果逐渐接近 Pareto 前沿; 二是如何保持进化群体的多样性, 使得最终求得的解集在目标空间中具有较好的分布特性, 且分布范围尽可能宽广. 为了同时解决这两个问题, 需要对全局和局部搜索能力进行协调.

在利用粒子群算法求解多目标优化问题中, 不同的指导粒子选择策略会很大程度地影响算法的拓扑结构, 进而影响到算法的局部和全局搜索能力, 最终使算法的优化性能有很大不同. Hu等^[5]提出动态邻域 PSO 算法求解多目标问题, 对两个目标逐步计

收稿日期: 2014-09-14; 修回日期: 2015-01-27.

基金项目: 上海市自然科学基金项目(14ZR1409900); 上海市科委基础研究重点项目(12JC1403400).

作者简介: 王学武(1972—), 男, 副教授, 博士, 从事工业过程建模、控制与优化等研究; 薛立卡(1990—), 男, 硕士生, 从事智能优化算法的研究.

算粒子邻域内适应度较好的粒子,将其作为该粒子的指导粒子.文献[6-7]指导粒子的选取策略是,在所得到的外部档案内的非劣解中随机选取一个.与上述二者相比,一些学者为了均衡各个区域的搜索概率,对指导粒子的选取基于非劣解在目标空间的密度,密度较大的非劣解有较大的概率被选择为指导粒子,Coello等^[8]采用网格法度量密度,Reddy等^[9]采用相邻非劣解间的欧几里得距离求解,向长城等^[10]采用小生境方法作为密度的评价指标.Mostaghim等^[11]提出了Sigma方法选择指导粒子,该方法可以使粒子向着Pareto前沿的方向搜索,以提高算法收敛速度,但与前面几种算法相比,该算法收敛速度的提高势必以牺牲全局搜索能力为代价.

以上算法的指导粒子选择方法都是基于目标空间进行选择,当优化问题真实Pareto前沿在决策空间不连贯、非线性时,这些指导粒子策略很大程度上会导致大部分粒子在决策空间远离Pareto前沿的区域搜索,不利于接近Pareto前沿,导致搜索效率较低.虽然文献[12]基于决策变量空间选择非劣解,但在子群迭代过程中,全局最优粒子间无信息交流,不利于全局搜索.因此,本文提出一种新型基于决策空间的指导粒子选择策略(3态指导粒子选择策略),将所有粒子分为3种状态:不受约束状态、激发态、非激发态,根据粒子的状态确定其指导粒子选择策略.新型指导粒子选择策略在增大局部搜索效率的同时,兼顾了粒子间的信息交流以协调全局搜索.另外,多目标粒子群算法在迭代过程中产生的非支配解集通常使用外部档案进行保存,其与NSGA-II^[13]中的精英解集保存策略类似.当非劣解超过预先设定的上限数目时,会剔除密度较大区域的非劣解.但以上算法在剔除多余的粒子时,均采用一次性剔除所有多余非劣解的方法,该方法会将过密区域的非劣解全部剔除,反而导致该区域没有非劣解,不利于获得均匀分布的非劣解.本文改进了传统的外部档案保存机制,以分步剔除多余的非劣解的方法增强所得非劣解在Pareto前沿的均匀分布性.

由于PSO算法在迭代后期可能会陷入局部最优,为了克服这个缺点,很多学者在利用PSO解决多目标优化问题时引入了遗传算法中的突变因子使粒子跳出局部最优^[6-7,14],本文采用3种突变因子来解决这个问题.

综上所述,本文提出的新型多目标PSO算法,在指导粒子选择策略和外部档案保存机制两方面进行改进,引用突变因子,使算法能够协调全局和局部搜索能力,增大在Pareto前沿的搜索概率,同时又保证种群的多样性,最终使所得非劣解尽可能靠近真

实Pareto前沿,并能够在解空间均匀分布.理论上,新算法在求解复杂问题,如决策空间不连贯、非线性问题时,具有更好的优越性.

1 TC-MOPSO 算法

1.1 3态指导粒子选择策略

本文所提出的3态指导粒子选择策略,首先基于决策空间将所有粒子分为受约束粒子和不受约束粒子;然后根据粒子激发机制将受约束粒子分为激发态粒子和非激发态粒子;最后将所有粒子分为3种状态:不受约束状态、激发态、非激发态,不同状态的粒子有不同的指导粒子选择策略.

1.1.1 受约束粒子-不受约束粒子分类

Step 1: 根据前一次迭代所得的非劣解,运用聚类的未加权平均距离法将决策空间划分区域,并编号Region_{*i*}(*i* = 1, 2, ..., *n*),统计区域*i*所含非劣解的数量 nn_i .其中*n*为所分区域的数量,本文设置 $n = \text{ceil}(\text{CR} \times \text{非劣解的数量})$,CR为区域划分系数,ceil函数为返回大于或者等于指定表达式的最小整数.

Step 2: 计算各个区域的中心,有

$$\text{center}_{i,j} = \frac{1}{nn_i} \sum_{s=1}^{nn_i} x_{s,j}. \quad (1)$$

其中: $\text{center}_{i,j}$ 为Region_{*i*}中心的第*j*(*j* = 1, 2, ..., *D*)维变量坐标, $x_{s,j}$ 为Region_{*i*}所含的第*s*个非劣解的第*j*维变量坐标.

Step 3: 计算各个区域中心之间的欧几里得距离,有

$$d_{m,n} = \left(\sum_{j=1}^D (\text{center}_{m,j} - \text{center}_{n,j})^2 \right)^{0.5}. \quad (2)$$

其中: $d_{m,n}$ 为Region_{*m*}、Region_{*n*}之间的距离; $d_i = \min(d_{i,j})$ 为Region_{*i*}的辐射半径, $j = 1, 2, \dots, i-1, i+1, \dots, n$.

Step 4: 计算各个粒子与各个区域中心的欧几里得距离,有

$$dR_{q,i} = \left(\sum_{j=1}^D (x_{q,j} - \text{center}_{i,j})^2 \right)^{0.5}, \quad (3)$$

其中 $dR_{q,i}$ 为粒子*q*到Region_{*i*}中心的距离.找出与粒子*q*距离最小的区域Region_{*t*}, $dR_{q,t} = \min(dR_{q,i}), i = 1, 2, \dots, n$.如果 $dR_{q,t} \leq d_t$,则粒子*q*是Region_{*t*}的受约束粒子,否则粒子*q*为不受约束粒子.

Step 5: 不受约束粒子的指导粒子在所有非劣解目标空间距离排序靠前10%的非劣解中随机选择(全局选择策略),若将所有受约束粒子的指导粒子从其所在区域内的非劣解中随机选择(区域选择策略),则会使一些区域内部粒子过多,其他区域内的粒子相对过少.因此,这些区域的搜索概率相应会降低,应采取

措施平衡各个区域的搜索概率, 本文采取一种粒子激发机制解决该问题.

1.1.2 粒子激发机制

粒子激发机制可以描述为: 当某个区域中的受约束粒子过多时, 该区域中的粒子便会有较大概率被激发, 成为激发态粒子, 将激发态粒子的指导粒子选择策略设为全局选择策略, 以使该粒子跳出该区域. 激发机制可以动态地调整各个非劣解附近的搜索粒子数量, 以平衡各个区域的搜索概率. 但是, 当某个区域中的非劣解数量过少时, 若使非劣解数量和搜索概率一致导致该区域的搜索粒子数量过少, 则使该区域的搜索陷入停滞状态, 因此在非劣解过少的区域中, 应适当增加搜索粒子. 粒子激发机制具体步骤如下.

Step 1: 统计各个区域拥有的受约束粒子个数 np_i .

Step 2: 计算各个区域的粒子激发值, 有

$$\text{ratio}_i = \begin{cases} 4 \times \frac{nn_i}{np_i} \times \frac{N}{R}, & nn_i < 3; \\ \frac{nn_i}{np_i} \times \frac{N}{R}, & \text{otherwise.} \end{cases} \quad (4)$$

其中: np_i 为 Region_i 所含受约束粒子的个数, N 为种群规模, R 为外部档案上限

Step 3: 对于每个受约束粒子, 若 $\text{rand} < \text{ratio}_i$ (rand 为 $0 \sim 1$ 之间的随机数), 则该粒子属于非激发态粒子, 否则该粒子被激发成为激发态粒子. 非激发态粒子指导粒子策略为区域选择策略, 激发态粒子指导粒子策略为全局选择策略.

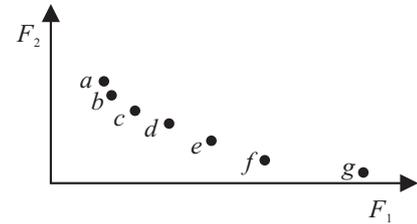
综上所述, 3 态指导粒子选择策略为: 首先将所有粒子分为受约束粒子和不受约束粒子两类, 远离所有区域的粒子为不受约束粒子, 靠近某个区域的粒子为受约束粒子; 然后运用粒子激发机制将受约束粒子分为激发态和非激发态粒子两类, 激发态和不受约束的粒子指导粒子选择策略为全局选择策略, 非激发态粒子的指导粒子选择策略为区域选择策略.

1.2 改进外部档案保存机制

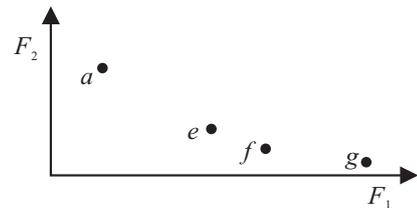
传统的外部档案保存机制为: 计算非劣解之间的距离, 当外部档案规模超过最大规模时, 依据非劣解的距离, 一次性剔除所有密度过大的多余非劣解. 但是, 当部分区域的非劣解分布过于密集时, 该区域的所有非劣解将会全部被剔除, 从而导致所得非劣解分布均匀性变差.

针对传统外部档案保存机制的缺点, 本文首先基于目标空间计算相邻非劣解间的距离, 将非劣解与其相邻的两个解的平均距离作为其密度值, 平均距离越小代表其密度越大. 当外部档案规模超过最大规模时, 依据非劣解的密度值多次剔除多余的非劣解, 每次仅

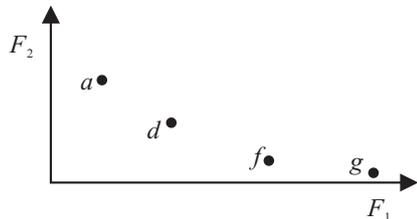
剔除密度最大的非劣解, 每次剔除后重新计算非劣解的密度. 在重新计算非劣解密度时, 由于只有与被剔除非劣解相邻的非劣解密度发生改变, 只需重新计算与被剔除非劣解相邻的非劣解密度即可, 而不需要每次计算所有剩余非劣解的密度, 这样可以很大程度地减小计算复杂度, 同时尽可能使保留的非劣解在目标空间均匀分布.



(a) 初始非劣解分布



(b) 传统外部档案保存机制所得结果



(c) 改进外部档案保存机制所得结果

图 1 外部档案保存机制对比

图 1(a) 为 7 个非劣解 a, b, c, d, e, f, g 在目标空间的分布. 假设根据外部档案保存机制保留 4 个非劣解, 则采用传统和改进后的外部档案保存机制保留非劣解的过程如下.

1) 根据传统外部档案保存机制保留非劣解过程如下.

Step 1: 计算所有非劣解的密度;

Step 2: 剔除密度最大的 3 个非劣解 b, c, d , 最终剩余非劣解为 a, e, f, g , 如图 1(b) 所示.

2) 根据改进外部档案保存机制保留非劣解过程如下.

Step 1: 计算所有非劣解的密度;

Step 2: 剔除密度最大的非劣解 b , 更新之前与非劣解 b 相邻的非劣解 a, c 的密度;

Step 3: 剔除剩余 6 个非劣解 a, c, d, e, f, g 中密度最大的非劣解 c , 更新之前与非劣解 c 相邻的非劣解 a, d 的密度;

Step 4: 剔除剩余 5 个非劣解 a, d, e, f, g 中密度

最大的非劣解 e , 最终剩余非劣解为 a, d, f, g , 如图 1(c) 所示.

通过比较图 1(b) 和图 1(c) 可以看出, 改进后的外部档案保存机制明显增强了非劣解在目标空间的均匀分布性.

1.3 突变因子

虽然 PSO 算法收敛速度快, 但在搜索后期可能会陷入局部极值, 因此很多文献均在粒子群算法中加入突变因子, 以增强所得解的多样性. 本文加入 3 种突变因子, 分别为变范围突变因子、固定范围突变因子和突变为非劣解. 分别依据概率进行突变: 若随机数 $\text{rand}_1 < 0.15$ (rand_1 为 $0 \sim 1$ 的随机数), 则粒子发生突变; 若 $\text{rand}_2 < 0.2$ (rand_2 为 $0 \sim 1$ 的随机数), 则粒子为变范围突变; 若 rand_2 在 $[0.2, 0.9]$ 之间, 则粒子突变为非劣解; 若 $\text{rand}_2 > 0.9$, 则粒子为固定范围突变.

1.4 TC-MOPSO 算法步骤

根据以上算法的描述, 一种新的基于 PSO 的多目标优化算法详细步骤如下.

Step 1: 初始化参数, 种群规模 N 、迭代次数 T_{\max} 、外部档案上限 R 、收缩因子、学习因子、惯性权重、区域划分系数 CR 等. 初始化种群各个粒子的位置、速度和个体历史最优值.

Step 2: 运用 Pareto 支配方法, 找出非劣解集并将其储存在外部档案中, 计算非劣解基于目标空间的距离.

Step 3: 为每个粒子寻找指导粒子. 如果非劣解数量不大于 20, 则在非劣解集中随机选择一个非劣解作为其指导粒子; 如果非劣解数量大于 20, 则运用 3 态指导粒子选择策略为其寻找指导粒子.

Step 4: 运用公式更新每个粒子的位置和速度, 评估粒子的适应度, 根据适应度更新粒子的个体历史最优.

Step 5: 将外部档案的非劣解和当前迭代的粒子放入一个暂时的档案中, 运用 Pareto 支配方法, 找出暂时档案中的非劣解集, 并将其储存在外部档案中, 计算非劣解的密度. 若外部档案规模超出上限, 则运用本文提出的外部档案保存机制保留 R 个非劣解.

Step 6: 引入突变因子, 在当前的迭代粒子中依概率产生突变.

Step 7: 增加迭代次数 t , 若 $t < T_{\max}$, 则转至 Step 4, 否则终止迭代, 输出外部档案中的解作为最终解.

2 算法性能测试

本文选择 TC-MOPSO 算法与 CMOPSO 算法^[8]、

EM-MOPSO^[6]、NSGA-II^[13] 三个多目标优化算法进行比较, 对 7 个标准测试函数进行求解, 通过 GD、SC、 Δ 和运行时间 t 等 4 个性能评价指标分析新算法 TC-MOPSO 的性能, 并分析区域划分系数 CR 对算法性能的影响.

2.1 性能评价准则

学者们对多目标优化算法的性能测试提出了各种评价准则, 本文运用其中 3 个评价准则评价算法的优劣, 同时加入算法的运行时间对算法进行评价.

1) GD (代距). 该标准用来描述算法所获得的非劣解与问题的真实 Pareto 前端之间的距离, 有

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}. \quad (5)$$

其中: n 为算法所得非劣解的个数, d_i 为第 i 个解到真实 Pareto 最优解集的最小欧几里得距离 (在目标向量空间). 若 $GD = 0$, 则表示所得非劣解均属于 Pareto 最优解集, 该指标反映算法所得优化解集与 Pareto 最优解的逼近程度.

2) SC (两集覆盖度). 该指标表示得到的非劣解能被 Pareto 前端所支配的比率, 从另一个角度描述所得非劣解逼近 Pareto 前端的程度, 有

$$SC(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X' : a' \leq a''\}|}{|X''|}. \quad (6)$$

3) Δ (多样性指标). 该指标描述所得到解的分布范围的多样性, 不仅评估了所得解的均匀分布程度, 也评估了真实 Pareto 前沿的跨度, 能有效地评估所得解在真实 Pareto 前沿的分布情况, 有

$$\Delta = \frac{\sum_{m=1}^M h_m + \sum_{i=1}^{N-1} |h_i - \bar{h}|}{\sum_{m=1}^M h_m + (N-1)\bar{h}}. \quad (7)$$

其中: N 为所得非劣解的个数, h_i 为所得的非劣解与相邻非劣解之间的欧几里得距离, \bar{h} 为 h_i 的均值, h_m 为第 m 维目标空间中所得到的非劣解末端与真实 Pareto 前沿末端的距离.

4) t (运行时间). 该指标为算法迭代最大次数所需要的时间, 可在一定程度上反应算法的计算复杂度.

2.2 测试函数

本文所用测试函数如表 1 所示. 7 个测试函数的真实 Pareto 前沿取点个数依次为 862、871、448、659、218、500、136. 以上测试函数数据均取自 <http://www.cs.cinvestav.mx/emooobook/>.

表 1 实验所用测试函数

test problems	objective functions definition	variables definition
Kita: 决策空间 Pareto 前沿连贯、线性	$\max F = (F_1(x), F_2(x))$ $F_1 = -x^2 + y, F_2 = \frac{1}{2}x + y + 1$	$n = 2, x, y \geq 0, 0 \geq \frac{1}{6}x + y - \frac{13}{2}$ $0 \geq \frac{1}{2}x + y - \frac{15}{2}, 0 \geq 5x + y - 30$
Kursawe(l): 决策空间 Pareto 前沿不连贯、非线性	$\min F = (F_1(x), F_2(x))$ $F_1(x) = \sum_{i=1}^{n-1} (-10e^{0.2\sqrt{x_i^2+x_{i+1}^2}})$ $F_2(x) = \sum_{i=1}^n (x_i ^{0.8} + 5 \sin(x_i)^3)$	$n = 3$ $-5 \leq x_1, x_2, x_3 \leq 5$
Tanaka: 决策空间 Pareto 前沿连贯、非线性	$\min F = (F_1(x), F_2(x))$ $F_1(x, y) = x, F_2(x, y) = y$	$n = 2, 0 < x, y < \pi$ $0 \geq -x^2 - y^2 + 1 + 0.1 \cos(16 \arctan(x/y))$ $\frac{1}{2} \geq \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2$
deb(2): 决策空间 Pareto 前沿不连贯、线性	$\min F = (F_1(x), F_2(x))$ $F_1(x) = x_1, F_2(x, g) = g(x)h(x)$ $g(x) = 1 - 10x_2, h(x) = 1 - \left(\frac{F_1}{g(x)}\right)^2 - \frac{F_1}{g(x)} \sin(12\pi F_1)$	$n = 2$ $0 \leq x_i \leq 1, i = 1, 2$
Binh(1): 决策空间 Pareto 前沿连贯、线性	$\min F = (F_1(x, y), F_2(x, y))$ $F_1(x, y) = x^2 + y^2, F_2(x, y) = (x - 5)^2 + (y - 5)^2$	$n = 2$ $-5 \leq x, y \leq 10$
ZDT2: 决策空间 Pareto 前沿连贯、非线性	$\min F = (F_1(x), F_2(x))$ $F_1(x) = x_1, F_2(x, g) = g(x)\left(1 - \left(\frac{F_1}{g(x)}\right)^2\right)$ $g(x) = 1 - \frac{9}{n-1} \sum_{i=2}^n x_i$	$n = 30$ $0 \leq x_i \leq 1, i = 1, 2, \dots, 30$
ZDT3: 决策空间 Pareto 前沿不连贯、非线性	$\min F = (F_1(x), F_2(x)), F_1(x) = x_1$ $F_2(x, g) = g(x)\left(1 - \sqrt{\frac{F_1}{g(x)}} - \frac{F_1}{g(x)} \sin(10\pi F_1)\right)$ $g(x) = 1 - \frac{9}{n-1} \sum_{i=2}^n x_i$	$n = 30$ $0 \leq x_i \leq 1, i = 1, 2, \dots, 30$

2.3 4种算法性能对比

2.3.1 参数设置

4种算法的种群规模均为100, TC-MOPSO、CMOPSO、EM-MOPSO的外部档案规模为100, NSGA-II的精英解集规模为100. 4种算法在求解高维测试函数ZDT2、ZDT3时迭代500次, 求解其余测试函数时迭代100次. TC-MOPSO的区域划分系数为0.1, 学习因子 C_1 为1.0, 学习因子 C_2 随着迭代从0.5线性递增到1.5, 收缩因子为0.9, 惯性权重为0.4. CMOPSO学习因子 C_1 、 C_2 为1.0, 惯性权重为0.4, 分格数为30, 突变因子的参数与原文保持一致. EM-MOPSO的学习因子 C_1 、 C_2 分别为1.0、0.5, 收缩因子为0.9, 惯性权重为0.4, 在突变过程中, 突变粒子数为15, P_{em} 为0.2, 突变范围mutScale随着迭代从0.2线性递减到0.01. NSGA-II为实数编码, 交叉和突变因

子 η_c 、 η_m 为20, 突变时规模为原始种群规模的一半.

4种算法运算编程采用相同的编程习惯, 每个测试问题随机独立运行20次, 统计各个评价指标的均值和方差. 测试环境为: CPU Intel(R)Core(TM) i5-3210M CPU@2.50 GHz; 内存4.00 GB, 操作系统64位 Windows 7, 运行平台 Matlab 2014a.

2.3.2 实验结果和分析

实验所得算法评价性能指标数据结果见表2~表5, 黑体表示最优值. 表2和表3显示各算法的收敛性, 表4数据表示各算法的分散性, 表5显示各算法的运行时间. 此外, 选择2个较为典型的测试函数Kursawe(l)(决策空间不连贯、非线性)和Kursawe(l)(高维、决策空间不连贯、非线性), 通过比较4种算法对这两个测试函数求得的非劣解在目标空间上的分布, 可显示出各算法的收敛性和分散性, 见图2和图3.

表 2 各算法GD比较

算法	GD	Kita	Kursawe(l)	Tanaka	deb(2)	Binh(1)	ZDT2	ZDT3
CMOPSO	均值	0.000 84	0.016 01	0.000 76	0.000 24	0.006 49	0.258 71	0.197 61
	方差	0.000 37	0.013 94	0.000 51	0.000 27	0.006 50	0.061 10	0.029 26
EM-MOPSO	均值	0.026 72	0.015 73	0.000 76	0.000 17	0.006 49	0.000 53	0.000 72
	方差	0.032 82	0.017 68	0.000 10	0.000 02	0.000 73	0.000 05	0.000 06
NSGA-II	均值	0.023 29	0.083 68	0.000 59	0.000 16	0.005 95	0.026 87	0.024 07
	方差	0.049 41	0.072 94	0.000 55	0.000 03	0.000 69	0.011 02	0.004 64
TC-MOPSO	均值	0.013 61	0.001 59	0.000 44	0.000 17	0.004 95	0.000 22	0.000 63
	方差	0.024 26	0.000 15	0.000 07	0.000 02	0.000 44	0.000 02	0.000 04

表3 各算法SC比较

算法	SC	Kita	Kursawe(l)	Tanaka	deb(2)	Binh(1)	ZDT2	ZDT3
CMOPSO	均值	0.888 00	0.903 00	0.798 90	0.694 50	0.238 00	1.000 00	1.000 00
	方差	0.127 84	0.046 12	0.035 50	0.196 19	0.269 01	0.000 0	0.000 0
EM-MOPSO	均值	0.768 50	0.977 48	0.876 01	0.362 00	0.490 00	0.902 00	0.757 96
	方差	0.053 93	0.018 01	0.029 82	0.065 18	0.038 80	0.036 51	0.080 60
NSGA-II	均值	0.706 00	0.822 00	0.529 50	0.027 50	0.483 50	1.000 00	1.000 00
	方差	0.046 39	0.135 09	0.100 18	0.021 24	0.047 82	0.000 0	0.000 0
TC-MOPSO	均值	0.601 50	0.525 50	0.609 50	0.464 50	0.361 00	0.595 00	0.315 00
	方差	0.059 94	0.058 71	0.062 78	0.121 63	0.057 48	0.052 47	0.140 28

表4 各算法 Δ 比较

算法	Δ	Kita	Kursawe(l)	Tanaka	deb(2)	Binh(1)	ZDT2	ZDT3
CMOPSO	均值	0.801 09	0.727 86	0.832 48	0.890 78	0.772 06	0.965 90	0.934 23
	方差	0.110 36	0.057 68	0.027 46	0.047 54	0.068 83	0.013 57	0.028 22
EM-MOPSO	均值	0.565 52	0.615 34	0.790 60	0.556 04	0.355 03	0.260 34	0.576 75
	方差	0.174 91	0.049 22	0.028 57	0.017 42	0.022 45	0.019 55	0.067 77
NSGA-II	均值	0.610 17	0.634 83	0.850 99	0.667 19	0.418 44	0.751 31	0.753 69
	方差	0.169 83	0.118 54	0.251 33	0.087 61	0.024 71	0.126 16	0.038 86
TC-MOPSO	均值	0.391 81	0.296 74	0.686 65	0.488 08	0.186 01	0.163 57	0.469 32
	方差	0.136 63	0.011 79	0.026 15	0.014 21	0.015 15	0.010 87	0.061 63

表5 各算法 t 比较

算法	t	Kita	Kursawe(l)	Tanaka	deb(2)	Binh(1)	ZDT2	ZDT3
CMOPSO	均值	2.657	1.112	1.302	1.411	3.555	9.874	5.986
	方差	0.458	0.103	0.051	0.081	0.205	2.106	0.916
EM-MOPSO	均值	0.733	0.671	0.934	0.586	0.612	1.933	2.154
	方差	0.034	0.031	0.043	0.031	0.055	0.126	0.032
NSGA-II	均值	23.243	17.823	28.673	18.828	16.616	55.352	53.584
	方差	0.576	0.894	2.774	1.012	0.272	3.870	1.067
TC-MOPSO	均值	1.280	1.280	1.458	1.044	1.339	3.584	4.170
	方差	0.158	0.066	0.024	0.032	0.117	0.077	0.218

表2的GD和表3的SC共同反映了4种算法在求解不同测试问题时算法的收敛性能。TC-MOPSO与其他两种MOPSO算法相比,在求解真实Pareto前沿变量空间不连贯、非线性的测试函数时,前者逼近真实Pareto前沿的能力远好于后两者,如求解Kursawe(l)、Tanaka、ZDT2、ZDT3。在求解简单问题时,当真实Pareto前沿变量空间线性时,三者的收敛性能相差不大。如Kita、Binh(1)、deb(2),在求解这样的测试函数时,利用传统的指导粒子选择策略可以求得较好的非劣解。TC-MOPSO在与NSGA-II比较时,前者的优点在于能够协调全局和局部搜索能力,后者因全局搜索能力过弱而易陷入局部最优;在求解真实Pareto前沿变量空间不连贯、非线性的测试函数时,如Kursawe(l)、Tanaka、ZDT2、ZDT3,前者能够更好地收敛到真实的真实Pareto前沿,后者不能协调好全局和局部搜索能力,因而易陷入局部最优;在求解

deb(2)时,前者求解收敛性略低于后者,原因在于这类问题并没有多个局部最优值,就搜索前沿的概率而言,NSGA-II的性能要优于TC-MOPSO,因此其收敛性能略优于TC-MOPSO。图2和图3也表明TC-MOPSO算法的收敛性要优于另外3种算法。

通过对表4数据的分析可知,TC-MOPSO在求解各个测试函数时,算法的分散性指标 Δ 均明显优于其他算法。由此可以看出,新算法采用的外部档案保存策略在使获得的非劣解Pareto前沿均匀分布方面显示出了强大的优越性。由图2和图3也可以看出,TC-MOPSO可以使所得到的非劣解均匀分布在真实Pareto前沿。

表5数据显示,3种MOPSO算法的时间远小于NSGA-II,在3种算法中,CMOPSO和TC-MOPSO相差不大,两者的时间略大于EM-MOPSO。可以说,新算法的复杂度与其他两种粒子群算法相比差别不大。

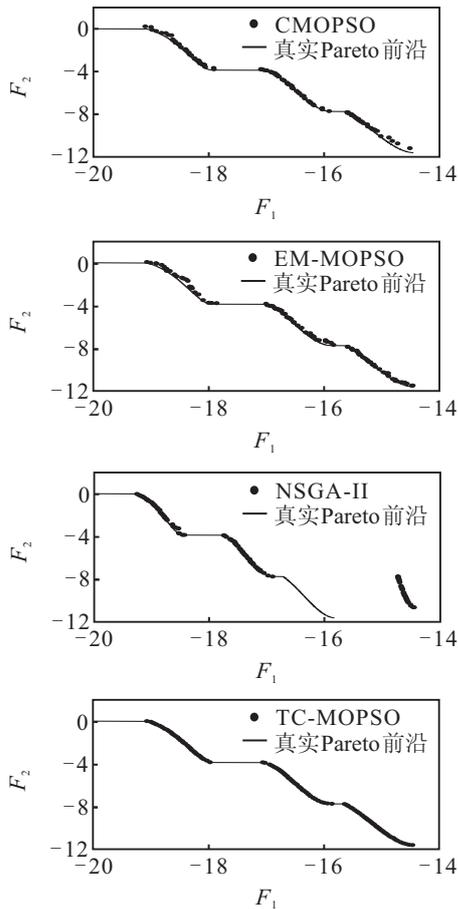


图 2 4 种算法求解 Kursawe(l) 所得的非劣解

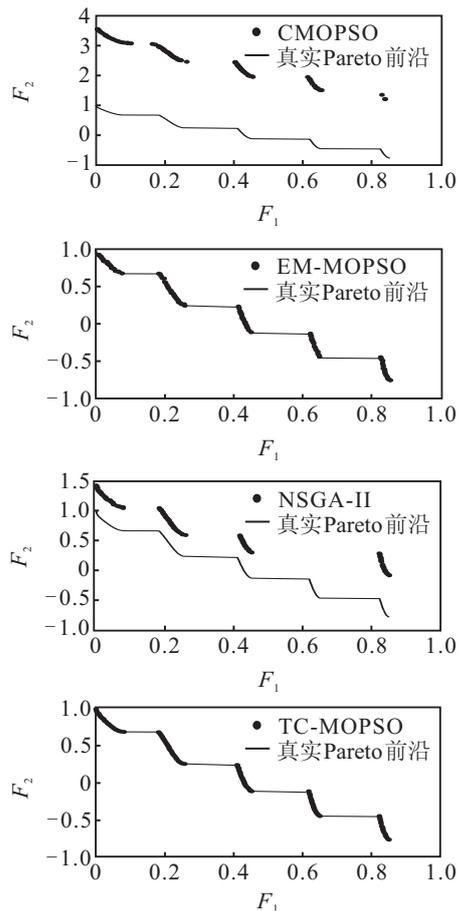


图 3 4 种算法求解 ZDT3 所得的非劣解

2.3.3 实验结论

由以上分析可以得出结论：新算法在增大局部搜索能力的同时协调了全局搜索能力，在求解真实 Pareto 前沿变量空间不连贯、非线性和高维优化问题等复杂问题时，收敛性有很大的提高；改进的外部档案保存机制提高了所得非劣解的分散性，同时仅以较小的算法复杂度为代价便可获得较好的优化性能。

2.4 区域划分系数 CR 分析

以上实验中，TC-MOPSO 的区域划分系数 CR 均设置为 0.1，CR 决定了算法对全局和局部搜索的协调能力，选择合适的 CR 可以使算法达到较好的优化性能，因此有必要分析 CR 对算法产生的影响。此处以测试函数 Kursawe(l) 为例，分析当 CR 变化时，TC-MOPSO 的优化性能变化情况。实验分析 CR 时，其他参数不变，参数设置参考第 2.3.1 节。实验中，CR 设置从 0.01 递增到 1，间隔为 0.01，CR 每改变一次，算法独立运行 20 次。算法的 4 个性能指标均值（进行归一化处理）随 CR 变化情况见图 4。

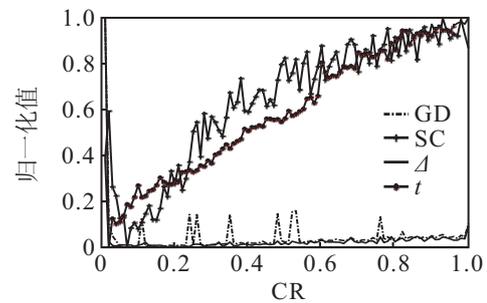


图 4 区域划分系数 CR 变化对算法性能的影响

由图 4 可见，CR 对算法的性能有非常大的影响。当 CR 为 0.01 时，算法不对决策空间进行区域划分，TC-MOPSO 的 3 态指导粒子策略失效，退化为 EM-MOPSO 的指导粒子选择策略，因此优化效果较差。随着 CR 的增加，局部搜索能力增加，全局搜索能力减弱。在 CR 大于 0.02 后，CR 的变化对 GD 和 Δ 的影响不大，而 SC 和 t 在 CR 变化的整个过程中都比较敏感。

区域划分系数 CR 对算法的收敛性有很大影响，因为该参数决定了算法的搜索能力。当 CR 在 0.1 附近时，收敛性能最好。当 CR 大于 0.5 时，由于式 (4) 的原因，粒子激发机制效果减弱，算法的优化收敛性能降低。当 CR 过大时，过多的区域划分不利于粒子间的信息交流，不利于算法在 Pareto 前沿的搜索。因此，CR 应与测试函数 Kursawe(l) 的真实 Pareto 前沿在决策空间的连贯区域数量相匹配，才能达到较好的优化效果。由于决定算法分散性的是外部档案保存机制，当 CR 大于 0.02 时，该参数的变化对算法分散性 Δ 指标的影响不大。对于 t 的影响，主要在于随着 CR 的增加，算法在指导粒子选择策略中算复杂度有所增加。

综上所述, 区域划分系数 CR 很大程度上决定了算法的性能, 对于测试函数 Kursawe(I), 当区域划分系数 CR 在 0.05~0.15 之间时, 算法 TC-MOPSO 的 4 个评价指标均较为满意。

3 结 论

本文提出了一种 3 态协调搜索多目标粒子群优化算法. 该算法使不受约束粒子、激发态粒子、非激发态粒子 3 种粒子互相协调, 动态转化. 粒子激发机制平衡各个区域的搜索概率; 非激发态粒子增大粒子在 Pareto 前沿附近的搜索概率; 受约束粒子和激发态粒子的信息传递方式的拓扑结构为全互联结构, 有助于整个群体的信息传递, 增加种群的多样性, 因此算法兼顾了局部和全局搜索能力. 同时, 为了使获得的非劣解在 Pareto 前沿均匀分布, 对传统的外部档案保存机制进行改进, 当外部档案规模超过上限时, 分步剔除较密集区域的非劣解. 另外, 为了克服粒子群算法后期易陷入局部最优, 引入了 3 种突变因子. 运用新算法对 7 个标准测试函数进行求解, 并与 CMOPSO、EM-MOPSO、NSGA-II 算法进行对比. 结果显示, 新算法有更好的收敛性和多样性, 同时算法的复杂度也相对较小, 特别是在求解复杂优化问题(高维、解的决策空间不连贯、非线性)时, 新算法的优越性更加明显. 最后, 分析了区域划分系数变化对算法性能所产生的影响。

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Perth: IEEE Service Cneter, 1995: 1942-1948.
- [2] 潘峰, 李位星, 高琪. 粒子群优化算法多目标优化[M]. 北京: 北京理工大学出版社, 2013: 23-30.
(Pan F, Li W X, Gao Q. Particle swarm optimization multi-objectives optimization[M]. Beijing: Beijing Institute Technology Press, 2013: 23-30.)
- [3] Kennedy J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance[C]. Proc of the 1999 Congress on Evolutionary Computation. Washington DC: IEEE Press, 1999: 1933-1938.
- [4] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.
- [5] Hu Xiaohui, Eberhart R. Multiobjective optimization using dynamic neighborhood particle swarm optimization[C]. Proc of the 2002 Congress on Evolutionary Computation. Honolulu, 2002: 1677-1681.
- [6] Carlo R Raquel, Prospero C Naval. An effective use of crowding distance in multiobjective particle swarm optimization[C]. Proc of the 2005 Conf on Genetic and Evolutionary Computation. New York: ACM, 2005: 257-264.
- [7] Li Xiaodong. Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function[C]. The 6th Annual Genetic and Evolutionary Computation Conf. Berlin: Springer Heidelberg, 2004: 117-128.
- [8] Coello C Ac, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.
- [9] Reddy M Janga, Kumar D Nagesh. An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design[J]. Engineering Optimization, 2007, 39(1): 49-68.
- [10] 向长城, 黄席樾, 杨祖元, 等. 小生境粒子群优化算法[J]. 计算机工程与应用, 2007, 43(15): 41-43.
(Xiang C C, Huang X Y, Yang Z Y, et al. Niche particle swarm optimization algorithm[J]. Computer Engineering and Applications, 2007, 43(15): 41-43.)
- [11] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization(MOPSO)[C]. Proc of the 2003 IEEE Swarm Intelligence Symposium. Indianapolis: IEEE Press, 2003: 26-33.
- [12] Toscano Pulido G, Coello Coello C A. Using clustering techniques to improve the performance of a particle swarm optimizer[C]. Proc of Genetic and Evolutionary Computation Conference. Berlin: Springer Heidelberg, 2004: 225-237.
- [13] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [14] Margarita Reyes Sierra, Coello Coello C A. Improving pso-based multi-objective optimization using crowding, mutation and ϵ -dominance[J]. Evolutionary Multi-Criterion Optimization, 2005(3140): 505-519.