

集成协方差矩阵自适应进化策略与差分进化的优化算法

杨紫晴, 姚加林, 伍国华, 陈学伟, 毛成辉[†]

(中南大学 交通运输工程学院, 湖南 长沙 410075)

摘要: 不同智能优化算法在求解优化问题时通常表现出显著的性能差异. 差分进化(DE)算法具备较好的全局搜索能力, 但存在收敛慢、效率低的不足, 协方差矩阵自适应进化策略(CMA-ES)局部搜索能力强, 具备旋转不变性, 但容易陷入局部最优, 因此, DE和CMA-ES之间具有潜在的协同互补能力. 针对上述问题, 提出了一种集成协方差矩阵自适应进化策略与差分进化的优化算法(CMADE). 在CMADE框架中, DE算法负责全局搜索, CMA-ES算法进行局部搜索. 通过周期性解交换机制实现CMA-ES和DE两个算法间协同交互和反馈控制. 在解交换时, 从DE种群中选择优秀个体, 利用CMA-ES算法在优秀个体周围进行局部搜索. 同时在DE和CMA-ES的混合种群中, 综合考虑解的多样性和最优性, 选取一定比例的解作为DE算法的新种群进行全局搜索, 实现全局搜索与局部搜索的动态平衡. 将CMADE算法与CMA-ES, DE, SaDE, jDE, EPSDE, ACO和SHADE算法在CEC2014标准测试集上进行比较实验. 结果表明, CMADE整体性能显著优于其它比较算法.

关键词: 智能优化算法; 差分进化; 协方差矩阵自适应进化策略; 算法集成; 连续优化

引用格式: 杨紫晴, 姚加林, 伍国华, 等. 集成协方差矩阵自适应进化策略与差分进化的优化算法. 控制理论与应用, 2021, 38(10): 1493 – 1502

DOI: 10.7641/CTA.2021.10002

Ensemble optimization algorithm from covariance matrix adaptive evolution strategy and differential evolution

YANG Zi-qing, YAO Jia-lin, WU Guo-hua, CHEN Xue-wei, MAO Cheng-hui[†]

(School of Transportation, Central South University, Changsha Hunan 410075, China)

Abstract: Different metaheuristic algorithms often show significant performance differences in solving optimization problems. Differential evolution (DE) algorithm has a good global search ability, while it has disadvantages of slow convergence speed and low efficiency. Covariance matrix adaptive evolution strategy (CMA-ES) has strong local search capabilities and rotation invariance, but it is easy to fall into local optimum. Therefore, there is a potential synergy between DE and CMA-ES. To address these problems, this paper proposes an integrated covariance matrix adaptive evolution strategy and differential evolution optimization algorithm (CMADE). Under the framework of the CMADE, DE algorithm is designed for global search, and CMA-ES algorithm is designed for local search. The two algorithms are integrated by a periodic information exchange strategy to realize collaborative interaction and feedback control. During the information exchange, the best individual is selected from the population of DE algorithm, then CMA-ES algorithm is used for local search around the best individual. Considering the diversity and optimality of solutions, we choose a new population from the mixed populations of DE and CMA-ES for global search, to realize the dynamic balance between global search and local search. We conduct extensive experiments on the suit of CEC 2014 benchmark functions and comprehensive comparisons with the seven algorithms including CMA-ES, DE, SaDE, jDE, EPSDE, ACO and SHADE, which demonstrates the superiority of the proposed CMADE.

Key words: intelligent optimization algorithm; differential evolution; covariance matrix adaptation evolution strategy; ensemble algorithm; continuous optimization

Citation: YANG Ziqing, YAO Jialin, WU Guohua, et al. Ensemble optimization algorithm from covariance matrix adaptive evolution strategy and differential evolution. *Control Theory & Applications*, 2021, 38(10): 1493 – 1502

收稿日期: 2021-01-02; 录用日期: 2021-04-23.

[†]通信作者. E-mail: mchh99@csu.edu.cn; Tel.: +86 731-82655234.

本文责任编辑: 王凌.

国家自然科学基金项目(62073341), 中南大学研究生自主探索创新项目(1053320190633)资助.

Supported by the National Natural Science Foundation of China (62073341) and the Independent Exploration and Innovation Project of Central South University (1053320190633).

1 引言

进化算法(evolutionary algorithm, EA)是模仿自然界的生物进化来实现种群的交叉、变异和选择操作的“算法簇”^[1]. EA包括遗传算法(genetic algorithm, GA), 进化规划(evolutionary programming, EP)和进化策略(evolutionary strategies, ES)等^[2-4]. 在相同的框架下, 不同的EA有不同的基因表达方式, 不同的交叉和突变操作以及不同的再生和选择方法. 与传统的基于梯度的优化方法或穷举法等的优化算法相比, EA是一种鲁棒性强、适用性广的全局优化方法, 具有自适应和自学习的特点, 可以有效地解决传统优化算法难以解决的复杂问题, 而不受问题性质的限制.

差分进化算法(differential evolution algorithm, DE)和协方差矩阵适应进化策略(covariance matrix adaptation evolution strategy, CMA-ES)在连续优化问题中表现出良好性能. DE算法由Storn和Price于1995年首次提出^[5], 可以有效地解决复杂的优化问题. DE算法是一种基于种群随机搜索机制的进化算法, 通过差分向量指导进化方向, 并且通过变异、交叉和选择来进行全局搜索. DE算法具有结构简单紧凑、易于实现和使用参数少等优点, 具有很强的全局搜索能力和鲁棒性, 并且不需要依赖于问题的特征信息, 因此它常用来解决复杂环境中的实际优化问题. CMA-ES算法是Nikolaus Hansen等人提出的一种优化算法^[6], 是最著名、应用广泛且性能突出的ES之一, 对复杂的优化问题具有良好的效果. 其主要核心思想有: 1) 去随机化: 新个体是根据多元正态分布产生的; 2) 累积进化路径: 计算一系列连续进化代之间的变异步长之和.

算法的性能通常受到局部和全局搜索能力影响. 全局搜索的目的是找到最优解的大致范围, 而局部搜索则是在该范围内进行精细搜索, 局部和全局搜索的平衡与控制, 对提高智能优化算法的性能具有重要意义, 许多学者在这一方面进行了相关研究. Tanabe和Fukunaga等提出了基于参数自适应成功历史的差分进化算法(success-history based adaptive differential evolution, SHADE), SHADE算法基于成功历史自适应调整参数, 缩放因子 F 根据正态分布来产生, 交叉概率 CR 根据高斯分布产生, 并采用Lehmer均值进行更新, 以此来平衡全局收敛和局部收敛的关系^[7]. Brest等提出了一种自适应控制参数的DE算法(jDE), 在jDE中, 2个主要参数 F 以及 CR 被编码进个体, 参数可以随着个体进入下一代, 最佳的参数和个体相互影响, 并具有更大的概率被保留^[8]. Omran等人将对立学习(opposition-based learning, OBL)、差分进化(DE)和量子力学等概念进行融合, 提出一种基于种群的随机元启发式方法, 用于在连续空间上进行全局优化^[9]. Zhang等提出的(adaptive differential evolution with

optional external archive, JADE)是一种改进的DE算法, JADE提出了一种新的变异策略, 即DE/current-to- p Best/1策略, 利用多个最佳解的信息来平衡突变的贪婪性和种群的多样性, 并以自适应方式更新控制参数来提高优化性能^[10]. Wu提出了一种基于多种群的方法来实现多策略的DE集成, 从而产生了一种新的DE变体, 它同时包含3种DE突变策略, 并且通过贡献来动态的分配资源^[11]. 为了提高LSHADE算法的整体性能, Mohamed等人提出了一种基于随机和自适应的半参数自适应方法, 用于在搜索过程中有效地适应差分进化算法的比例因子值, 同时还将LSHADE-SPA与修改后的CMA-ES进行融合, 提高LSHADE-SPACMA算法的搜索能力^[12]. Mallipeddi等提出一种变异策略和参数值集成的改进的DE算法(differential evolution algorithm with ensemble of parameters and mutation strategies, EPSDE), 该算法包含一组变异和交叉策略及其相关参数值, 相应的策略根据子代是否成功来进行保留, 以保证在进化过程中产生更好的策略和个体^[13]. Qin等人提出了一种自适应差分进化算法(strategy adaptation differential evolution, SaDE), 以降低通过试错模式来找到最合适的测试向量和相关参数所产生的计算成本. 在每一代中, SaDE根据在前几代中学习到的选择概率, 将一组测试向量生成策略及其相关参数值分配给当前种群中的不同个体^[14]. Wu等将DE算法中的变异策略、交叉策略、缩放因子和交叉概率作为4个部分, 4个组成部分之间的关系用有向图表示, 图中的一条路径正好对应于DE的一个配置, 在优化过程中, 采用蚁群优化算法根据弧上的信息素为每一个DE寻找一条合理的路径(ACODE)^[15]. 杜永兆等提出一种多种群协方差学习差分进化算法, 采用多种群机制的种群结构, 采用多种群的集成变异策略, 通过协方差学习建立交叉旋转坐标, 使用自适应控制参数来平衡种群的全局搜索与局部搜索^[16]. 肖鹏等将变异因子设置成线性递减函数, 在基向量前加入幅值系数以平衡全局搜索和局部搜索^[17]. 邹杰等人通过协方差矩阵建立特征坐标系, 在坐标系中执行变异和交叉操作, 利用种群信息来保证进化方向, 并且根据历史信息来选择变异策略^[18]. Caraffini 提出了CMA-ES-RIS算法, 该算法采用CMA-ES获得具有高适应度的个体, 并将其反馈给单解局部搜索, 迭代地扰动每个变量, 为了防止早熟收敛, CMA-ES-RIS算法的局部搜索采用重采样机制^[19]. Vinicius等针对约束优化问题, 提出了一种改进的协方差矩阵自适应进化策略, 该方法利用了CMA-ES的重启机制, 并通过自适应惩罚函数处理约束^[20].

相关研究表明, 为了解决具有多个特征的复杂问题, 具有不同优势和特征的不同优化算法、搜索策略以及参数值的集成可以得到更有效的优化算法^[21].

DE算法通过差分向量来指导搜索方向,通过参数种群规模 NP 、缩放因子 F 和交叉概率 CR 进行搜索控制,具备较强的全局搜索能力,但是收敛速度较慢。CMA-ES算法通过概率分布在一定范围内产生个体,适应于小规模种群问题^[22],能快速在局部范围内找到最优解,具备较强的局部搜索的能力,然而容易陷入局部最优。因此,本文提出了一种集成优化算法(简称为CMADe),该算法使用DE进行全局搜索,寻找全局最优解的大致范围,然后在该范围内用CMA-ES进行局部搜索,并通过解交换策略进行算法集成,在解交换过程中,采用概率参数控制解的多样性和质量。

本文的主要贡献包括:

1) 提出了一种基于全局和局部均衡搜索的集成框架,在该框架中,采用DE算法进行全局搜索,采用CMA-ES算法在最具潜力的个体附近进行局部搜索,实现DE和CMA-ES协同进化,搜索全局最优解。

2) 提出了一种解交换策略,采用两个参数 G_{change} 和 ρ 来控制算法的计算资源分配和搜索能力,保证全局搜索和局部搜索的平衡。

2 DE和CMA-ES算法结构

2.1 DE算法结构

DE算法对种群进行交叉变异和选择操作,并且通过缩放因子 F 、交叉概率 CR 和种群规模 NP 的3个参数来控制矢量方向。首先,生成初始种群。然后从总体中随机选择3个个体作为差分向量,并由参数 F 控制:根据一定的规则,将产生的突变个体与预定的目标个体相混合,产生测试个体。最后,利用贪婪原理在父母和后代中分离出下一代。

对于 d 维最小优化问题

$$\min f(x_1, x_2, \dots, x_d), \quad (1)$$

$$x_d^g = x_{1,d}^g, x_{2,d}^g, x_{i,d}^g, \quad (2)$$

其中: g 是代数, $i = 1, \dots, NP$ 是种群大小,初始种群 x_d^g 是在搜索范围内随机生成的。

1) 变异。

在第 g 次迭代中,从总体中随机选择3个个体 x_{p1}, x_{p2}, x_{p3} ,并且 p_1, p_2, p_3 互不相同,这3个个体生成的变异矢量为

$$V_i^g = x_{p1}^g + F * (x_{p2}^g - x_{p3}^g), \quad (3)$$

其中: $\Delta p = x_{p2}^g - x_{p3}^g$ 是一个差分向量, $F \in [0, 2]$ 是缩放因子。以下列出常用的5个变异策略:

$$\text{DE/rand/1} : V_i^g = x_{p1}^g + F * (x_{p2}^g - x_{p3}^g), \quad (4)$$

$$\text{DE/best/1} : V_i^g = x_{p_{\text{best}}}^g + F * (x_{p1}^g - x_{p2}^g), \quad (5)$$

$$\text{DE/current-to-best/1} :$$

$$V_i^g = x_i^g + F * (x_{p_{\text{best}}}^g - x_i^g) + F * (x_{p1}^g - x_{p2}^g), \quad (6)$$

$$\text{DE/best/2} :$$

$$V_i^g = x_{p_{\text{best}}}^g + F * (x_{p1}^g - x_{p2}^g) + F * (x_{p3}^g - x_{p4}^g), \quad (7)$$

DE/rand/2 :

$$V_i^g = x_{p1}^g + F * (x_{p2}^g - x_{p3}^g) + F * (x_{p4}^g - x_{p5}^g). \quad (8)$$

2) 交叉。

将目标个体 x_i^g 与变异个体 V_i^{g+1} 交叉,得出测试个体 U_i^{g+1} 。以参数交叉概率 $CR \in [0, 1]$ 作为阈值,从而使个体 x_i^g 和 V_i^{g+1} 可以被随机选择。

$$U_i^{g+1} = \begin{cases} V_{j,i}^{g+1}, & \text{rand}_{j,i} \leq CR \text{ 或 } j = I_{\text{rand}}, \\ x_{j,i}^g, & \text{其他}, \end{cases} \quad (9)$$

其中从 $[0, 1]$ 中随机生成 I_{rand} 以确保至少存在一维数据不同于目标个体 x_i^g 。

3) 选择。

DE选择操作如下所示:

$$x_i^{g+1} = \begin{cases} U_i^g, & U_i^g < x_i^g, \\ x_i^g, & \text{其他}. \end{cases} \quad (10)$$

2.2 CMA-ES算法结构

CMA-ES算法^[6, 23]是Nikolaus Hansen等人提出的一种新的无约束优化算法。文献[24]在CMA-ES算法的基础上,已成功应用于全局优化、多峰优化、多目标优化、大规模优化和结构工程等领域。

1) 产生新个体。

CMA-ES在每一代中从 $N(m_t, \sigma_t^2, C_t)$ 产生 λ 个个体。

$$x_i = m_t + \sigma_t y_i, y_i \sim N(0, C_t), \quad (11)$$

其中: $y_i \in \mathbb{R}^n$ 是搜索方向。通常, y 可以通过协方差矩阵 C 的特征分解来求解: $C_t = BD^2B^T$,并可以通过标准正态分布获得

$$y_i = BDz_i, z_i \sim N(0, I). \quad (12)$$

2) 计算适应度。

计算每一个新个体的适应度 $f(x)$,并进行适应度排序。

$$f(x_{1:\lambda}) \leq f(x_{2:\lambda}) \leq \dots \leq f(x_{\lambda:\lambda}), \quad (13)$$

下标表示第 i 个个体。

根据适应度排序,截取前 μ 个个体进行参数更新,一般采取 $\mu = \lfloor \lambda/2 \rfloor$,也可采用其他的截取方式。

3) 参数更新。

使用所选的个体分别更新分布参数 m_t, C_t 和 σ_t 。首先更新均值:

$$m_{t+1} = \sum_{i=1}^{\mu} \omega_i x_{i:\lambda} = m_t + \sum_{i=1}^{\mu} \omega_i (x_{i:\lambda} - m_t), \quad (14)$$

$$\sum_{i=1}^{\mu} \omega_i = 1, \omega_1 > \omega_2 > \dots > \omega_{\mu} > 0, \quad (15)$$

其中: m 为均值,下标 t 表示代数。 m_t 是个体 $x_{1:t}, \dots, x_{\lambda:t}$ 的加权平均值。

然后更新协方差矩阵,在CMA-ES中,协方差矩阵的更新包括rank-1和rank- μ .其中rank-1使用历史搜索信息,即进化路径.进化路径和协方差矩阵更新公式表达如下:

$$P_{t+1} = (1-c)p_t + \sqrt{c(2-c)}\sqrt{\mu_\omega} \frac{m_{t+1}-m_t}{\sigma_t}, \quad (16)$$

$$C_{t+1} = (1-c_1-c_\mu)C_t + c_1 p_{t+1} p_{t+1}^T + c_\mu \sum_{i=1}^{\mu} \omega_i \left(\frac{x_{i:\lambda} - m_t}{\sigma_t} \right) \left(\frac{x_{i:\lambda} - m_t}{\sigma_t} \right)^T, \quad (17)$$

其中 c_1 和 c_μ 为学习率.

CMA-ES算法使用累积步长调整,这是目前最成功的步长调整方法

$$s_{t+1} = (1-c_\sigma)s_t + \sqrt{c_\sigma(2-c_\sigma)} * \sqrt{\mu_\omega} B \sum_{i=1}^{\mu} \omega_i z_{i:\lambda}, \quad (18)$$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|s_{t+1}\|}{E \|N(0, I)\|} - 1 \right)\right). \quad (19)$$

3 基于协方差自适应与差分进化的集成优化算法

混合搜索算法的目的是采取各个算法的优点,弥补缺点.DE算法和CMA-ES算法是2个强大的进化算法,DE算法操作简单灵活,全局搜索能力强,但收敛速度慢.而CMA-ES算法下降速度快,但易陷入局部最优.因此本文将这两个算法集成起来,取长补短,达到全局和局部搜索的平衡.CMADE算法需要找到一个大致的搜索范围,再在该范围进行快速局部搜索,因此算法初期主要集中在DE算法进行全局搜索上,后期进行CMA-ES协同局部搜索.算法整体示意图如图1所示.

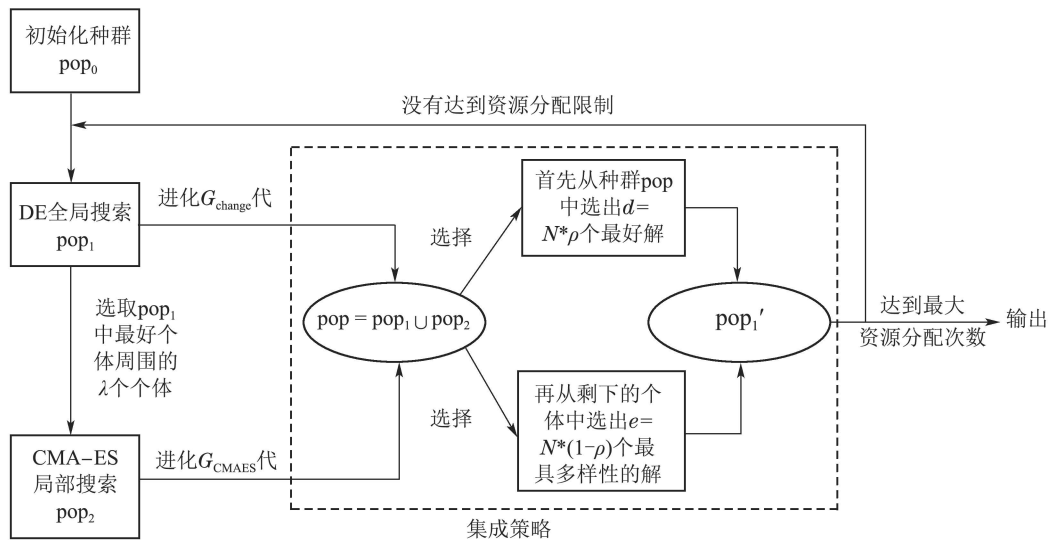


图1 CMADE 算法示意图

Fig. 1 Schematic diagram of CMADE algorithm

1) 随机初始化种群, $X_i = x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n}$, 生成种群 pop_0 . 将该种群赋予DE算法首先进行全局搜索. 此处DE算法的参数设置应该更有利于全局搜索.

2) 种群 pop_0 通过DE算法进行 G_{change} 代全局搜索后, 形成进化后的种群 pop_1 . 此时认为算法已经找到了最优解的大致的范围, 因此, 从 pop_1 中选取合适的种群范围进行局部搜索.

一般认为, 当前种群中, 最好解附近具有更大的搜索潜力, 因此局部搜索种群选择的方式为: 在DE进化后的种群 pop_1 中选取一个最好解, 以这个最好解为中心, 选取周围最近的 NP_{CMAES} 个解形成CMA-ES的初始种群进行局部搜索, 并且保留最终结果 pop_2 .

3) 将搜索后的种群进行解交换, 从中选择新种群重新赋予DE进行全局搜索. 新种群从 pop_1 和 pop_2 中

产生, 即 $pop = pop_1 \cup pop_2$. pop_1 为DE全局搜索后的种群, pop_2 为CMA-ES局部搜索后的种群. 该过程目的在于解的信息交换, 集成DE和CMA-ES算法的优势, 控制协调CMADE算法的全局搜索和局部搜索能力.

DE作为全局搜索算法, 不仅要考虑算法的收敛性, 而且要着重保留种群的多样性. 因此不仅要选取 pop 中最好个体, 还要选取一定数量具有多样性的个体. CMADE首先从种群 pop 中选取 $d = N * \rho$ 个最好个体, 然后从剩下的个体中选取 $e = N * (1 - \rho)$ 个最具有多样性的个体, 形成种群 pop_1' , 并作为新的种群赋予DE进行全局搜索. 其中 NP 为 pop 的种群规模, $\rho \in (0, 1)$ 为选取比率.

个体多样性的选取应考虑解空间的均匀分布. 本文将解空间分为已选解与备选解, 依次计算备选解中

每个个体与已选解中所有个体的距离之和, 选择距离和最远的个体作为最具多样性的解, 加入已选解种群中, 以此反复多次进行选择, 直到满足种群规模为止. 此方法可得到较均匀解的解分布.

根据上述步骤, 可以通过快速的局部搜索保证算法最大限度的利用资源, 通过全局搜索保证算法质量, 提高算法性能. CMADE算法实现过程如下所示:

```

1  设置参数全局迭代次数 $G_{\text{change}}$ , 种群多样性比例 $\rho$ ,  $G=0$ ; 设置 $NP_{\text{DE}}$ ; 设置 $NP_{\text{CMAES}}$ ;
2  随机初始化种群 $\text{pop}_0$ ;
3  // iteration
4  While  $n_{\text{fes}} \leq m_{\text{nfes}}$  do
5  DE算法变异、交叉、选择操作, 进行全局搜索, 形成种群 $\text{pop}_1$ .
6  If  $\text{mod}(G, G_{\text{change}}) == 0$  then
7  选择  $\text{pop}_1$  中最好个体的周围  $NP_{\text{CMAES}}$  个个体, 形成种群 $\text{pop}_2$ .
8  While  $G_{\text{CMAES}} \leq G_{\text{maxCMAES}}$  do
9  CMA-ES局部搜索, 种群 $\text{pop}_2$ 进化.
10 End while
11 End if
12 // information change
13  $\text{pop} = \text{pop}_1 \cup \text{pop}_2$ ;
14 从种群 $\text{pop}$ 中选出 $d = N * \rho$ 个最好解, 从剩下的个体中选出 $e = N * (1 - \rho)$ 个最具多样性的解, 形成一个新的种群 $\text{pop}'_1$ , 重新赋予DE算法进行全局搜索.
15  $G = G + 1$ ;
16 End while

```

其中, 第1行是设置CMADE的主要控制参数, 其中 G_{change} 是一个周期中全局搜索的迭代次数, 也可以控制解交换的次数, ρ 表示在解交换过程中, 选取的种群 pop_1 中最好个体和最具多样性个体的比例, 这2个参数是控制CMADE局部搜索和全局搜索最重要的参数, 直接决定着算法的性能; 第2行是初始化种群, 首先利用DE进行全局搜索.

第3-16行是算法的迭代部分. 其中, 第5行表示DE的全局搜索, DE算法有3个主要参数, 即种群规模 NP , 缩放因子 F 和交叉概率 CR , 由于DE算法在CMADE中主要起到全局搜索的作用, 因此缩放因子 F 和交叉概率 CR 的设置要使得性能较偏向于全局搜索; 而对于DE的种群规模 NP_{DE} , 由于DE的作用是为算法找到最优解的大致范围, 因此 NP_{DE} 不需要过大, 设置在合理范围内可以即节省资源, 又起到全局搜索的作用.

第6-11行表示CMA-ES进行局部搜索. 第6行表示在每个周期中, 当DE进化 G_{change} 代后, 此时认为已

经找到了较优解的大致范围, 则利用CMA-ES进行局部搜索; 第7行表示在 pop_1 最好个体附近进行局部搜索, CMA-ES的初始种群为DE最好个体附近的个体集合.

第8-10行为CMA-ES局部搜索. CMA-ES算法本身具有收敛快, 种群规模小的特点, 因此可以在节约资源的同时, 快速找到局部优解.

第12-14行表示解交换过程. 该过程实现了2个进化算法的集成, 以及全局搜索和局部搜索的反馈控制, 保持了DE算法的全局搜索能力, 同时保证了算法的收敛速度; 第13行表示下一个周期的种群从全局搜索和局部搜索后的集成种群中选取; 第14行为解的选取方式和步骤, 具体选取方式在本节第3)点中进行了说明.

在许多的集成算法与改进算法中, 大多数通过参数自适应的方法来动态的平衡全局搜索与局部搜索能力, 其基本原则是对差的解进行变异、扰动或更新操作, 并将好的解进行保留. 而CMADE采用的方法是分别设计有效的全局搜索和局部搜索策略, 然后在优化过程中将局部搜索和全局搜索策略进行合理融合. DE算法使用差分向量进化, 而CMA-ES算法采用无梯度下降的搜索方式, 这2种算法的搜索机制差异大且各具特点. 采用集成框架可以实现2个算法优势互补.

4 实验与分析

4.1 实验设置

为了测试CMADE算法的性能, 使用CEC2014测试集中的30个函数进行测试, 其中, F1-F3为单峰函数, F4-F16为多峰函数, F17-F22为混合函数, F23-F30为组合函数, 选择对比维度为30维. 每个算法在每个函数上运行30次. 所有算法目标函数评价的最大次数均设置为 $D \times 10000$, D 表示维度. 对比算法的参数设置与他们发布的版本一致.

实现CMADE时, 对应的参数设置为: $NP_{\text{DE}} = 50$, $NP_{\text{CMAES}} = 10$, $G_{\text{change}} = 1000$, $\rho = 0.3$. CMADE与其他7种最先进的算法进行了比较, 其中包括它的2个子算法: DE和CMA-ES算法, 和5种DE变体算法: SaDE^[14], jDE^[8], EPSDE^[13], ACO^[15]和SHADE^[7]. 选择这7种作为比较算法的原因解释如下: 首先, DE和CMA-ES是CMADE混合算法的子算法, 因此需要将它们进行比较, 来判断两者相结合之后的性能是否更具有优势. 第二, jDE和SHADE是有代表性的DE变体, 它非常高效, 在文献中通常会被引用作为基线算法. 第三, SaDE, EPSDE是包含了多重突变策略的算法. 最后, ACO是最近提出的DE混合算法, 它反映了DE的最新进展, 因此, 将CMADE与它们进行比较是有意义的. 以上算法的所有突变策略和参数设置都与原始参考文献中给出的一致, 不做修改.

4.2 实验结果

表1给出了通过在具有30变量的每个基准函数上对7个算法中的每一个运行30次所获得的计算结果,表中提供了测试函数的平均误差和标准误差.

此外,在CMADE与CMA-ES, DE, SaDE, jDE, EPSDE, ACO和SHADE之间将平均误差进行了0.05显著性水平的Wilcoxon秩和检验.与7个算法相比较后,符号“-”,“+”和“ \approx ”表示结果好、差、相等于对比算法,并将统计结果在表1中最后3行给出.从表1给出的数据中,可以得出一些观察和结论.

首先,对于单峰函数F1-F3, CMA-ES, DE, SaDE, jDE, EPSDE, ACO, SHADE和CMADE均在F2中表现出最佳性能,找到了全局最优解.只有DE和SaDE在函数F2上表现较差.在F1上,尽管7个对比算法的性能上优于CMADE,但是CMADE能够通过花费更多的函数评估次数来找到函数F1的全局最优解.

其次,对于简单的多峰函数F4-F16, CMADE整体上优于CMA-ES, DE, SaDE, jDE, EPSDE, ACO, SHADE.在F12中, CMADE会比DE, SaDE, jDE, EPSDE, ACO和SHADE算法更具有优势;在F15中, CMADE比DE, jDE, EPSDE, ACO, SHADE算法更具有优势. CMADE在F9和F16上的性能表现完全优于7个比较算法,不存在任何一个基准函数完全表现为劣势.

CMADE在7个(F6, F8-F11, F14, F16)、8个(F7, F9-F13, F15-16)、6个(F4, F6, F7, F11, F12, F16)、5个(F6, F9, F12, F13, F15)、8个(F6, F7, F9, F11, F12, F14-F16)、7个(F6, F7, F9, F11, F12, F15, F16)和7个(F8-F12, F15-F16)上的性能优于CMA-ES, DE, SaDE, jDE, EPSDE, ACO, SHADE.因此, CMADE在简单的多峰函数上性能的表现比其他7个算法都具有很强的竞争力.

对于混合函数F17-F22, CMADE在F18和F19上的性能完全优于其他7个比较算法,但在F22上表现比除DE算法外的其余6个算法差. CMADE仅仅在F22的性能上比DE, JADE, EPSDE较为劣势,而在F17-F21上, CMADE性能皆表现较好.

在组合函数F23-F30上, CMADE的性能在F23和F25中与7个比较算法基本持平,在F25上更占优势;而在F24上的性能上明显优于CMA-ES, DE, SaDE, jDE, EPSDE, ACO和SHADE算法.从总体上来看, CMADE在与7个比较算法的性能相比,好、差、相等的比例分别为41.1%, 12.5%, 46.4%,表明CMADE在

组合函数的性能表现与其他算法持平明显,但是依然存在大多数结果更具有优势,值得注意的是, SaDE在任何测试函数内的混合函数上均不能胜过CMADE.

在所有30个具有30个变量的基准函数中, CMA-DE的综合性能均优于其他5个对等DE变体,即SaDE, jDE, EPSDE, ACO, SHADE.实际上,在最后3行中报告的Wilcoxon的秩和检验结果表明,在16, 12, 16, 13和13个函数上, CMADE分别优于SaDE, jDE, EPSDE, ACO和SHADE,而分别在7, 8, 8, 7, 8个函数上比它们要差一些.从表1中可以看出,在30个变量的基准函数F9, F12, F16, F18, F19, F24和F30上, CMA-DE优于其他5个对比算法,在F2, F3, F5, F6, F7, F23, F25上均与它们持平.而与它的子算法比较, CMADE比CMA-ES和DE算法更优的基准函数个数分别为15, 14,更差的基准函数个数分别为8, 5,总体上性能优于原始算法.

综上所述,在本实验中, CMADE的整体性能是最好.然而,不同的比较算法在不同的基准函数上可能具有其特定的优势,例如:在函数F1, F17, F27上, SHADE的性能优于其他同等算法,而CMADE在函数F16, F18, F19, F24和F25上均优于其他所有比较算法. CMADE结合了DE与CMA-ES算法的优势,虽然在某些问题的优化方面未能完全取其优势,但是从整体上来看, CMADE性能要优于其他对比算法.因此, CMADE的性能更具有竞争力.

4.3 灵敏度分析

CMADE中,有2个与集成策略相关的参数,他们分别是控制DE和CMA-ES迭代次数及交换次数的参数 G_{change} ,以及控制交换后种群多样性的参数 ρ .另外,由于DE的种群规模 NP_{DE} 和CMA-ES的种群规模 NP_{CMAES} 和迭代次数 G_{CMAES} 直接关系到计算资源的配置,对CMADE算法的影响较大,所以在进行参数设置的时候,也将这2个参数也纳入考虑范围.选择不同的参数对CMADE性能会造成影响,本文选择了 G_{change} , ρ 和 NP_{DE} 3个主要参数,分析了他们对CMADE性能的影响.

在参数灵敏度分析中, G_{change} 的候选值包括500, 1500, 2000, NP_{DE} 的选值包括30, 70, 90, ρ 的选值包括0.2, 0.4, 0.5.其他参数均设置为默认值(即 $NP_{\text{DE}} = 50$, $NP_{\text{CMAES}} = 10$, $G_{\text{change}} = 1000$, $\rho = 0.3$, $G_{\text{CMAES}} = 20$),并且仍然进行0.05显著性水平的Wilcoxon秩和检验,对比分析其分别运行30次的平均误差.符号“-”,“+”和“ \approx ”表示结果好、差、相等于对比参数结果.参数 G_{change} , ρ 和 NP_{DE} 的灵敏度分析结果如表2所示.

表 1 CMADE 与其他算法的比较($D=30$)
Table 1 Comparison of computational results between CMADE and other algorithms ($D=30$)

序号		CMA-ES	DE	SaDE	jDE	EPSDE	ACODE	SHADE	CMADE
F1	平均值	0.00E+00	8.55E+04	9.99E+04	5.38E+04	1.13E+04	9.51E+03	0.00E+00	1.34E+05
	标准差	(0.00E+00) +	(2.23E+04) +	(0.00E+00) +	(2.79E+04) +	(1.97E+04) +	(2.72E+04) +	(0.00E+00) +	9.28E+04
F2	平均值	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	0.00E+00
F3	平均值	0.00E+00	1.47E-09	4.02E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	(0.00E+00) ≈	(8.06E-09) ≈	(9.57E+01) ≈	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	(0.00E+00) ≈	0.00E+00
F4	平均值	0.00E+00	3.43E+00	4.75E+01	8.95E-01	2.97E+00	2.11E+00	2.22E-01	3.49E+00
	标准差	(0.00E+00) +	(1.23E+01) ≈	(3.70E+01)-	(7.89E-01) +	(1.98E+00) +	(1.16E+01) +	(4.44E-01) +	1.24E+01
F5	平均值	2.00E+01	2.10E+01	2.05E+01	2.04E+01	2.04E+01	2.05E+01	2.06E+01	2.05E+01
	标准差	(1.00E-05) ≈	(4.07E-02) ≈	(4.10E-02) ≈	(4.24E-02) ≈	(6.99E-02) ≈	(5.05E-02) ≈	(1.05E-01) ≈	1.05E-01
F6	平均值	0.00E+00	1.82E-01	5.25E+00	9.13E-01	1.85E+01	1.05E+01	1.16E-05	2.23E-01
	标准差	(3.25E+00)-	(6.10E-01) +	(1.97E+00)-	(2.01E+00)-	(1.27E+00)-	(5.64E+00)-	(3.59E-05) +	5.61E-01
F7	平均值	0.00E+00	7.39E-04	7.30E-03	0.00E+00	8.22E-04	3.29E-04	0.00E+00	0.00E+00
	标准差	(0.00E+00) ≈	(2.83E-03)-	(1.19E-02)-	(0.00E+00) ≈	(2.53E-03)-	(1.80E-03)-	(0.00E+00) ≈	0.00E+00
F8	平均值	4.07E+02	2.05E+01	3.32E-02	0.00E+00	0.00E+00	0.00E+00	3.44E+01	2.26E+01
	标准差	(9.74E+01) -	(1.07E+01) +	(1.82E-01) +	(0.00E+00) +	(0.00E+00) +	(0.00E+00) +	(3.04E+00)-	6.29E+00
F9	平均值	6.30E+02	1.68E+02	3.69E+01	3.94E+01	4.43E+01	7.62E+01	7.90E+01	3.64E+01
	标准差	(1.43E+02) -	(4.11E+01)-	(1.14E+01) ≈	(6.11E+00)-	(6.75E+00)-	(6.61E+00)-	(7.20E+00)-	1.46E+01
F10	平均值	5.09E+03	1.88E+03	2.45E-01	2.08E-03	1.64E+00	1.20E+02	1.11E+03	4.42E+02
	标准差	(7.11E+02) -	(2.17E+03)-	(4.10E-01) +	(6.35E-03) +	(7.91E+00) +	(2.66E+01) +	(1.17E+02)-	2.23E+02
F11	平均值	4.32E+03	7.00E+03	3.19E+03	2.44E+03	3.46E+03	3.81E+03	4.46E+03	3.02E+03
	标准差	(8.76E+02) -	(2.60E+02)-	(5.70E+02)-	(3.17E+02) +	(4.09E+02)-	(2.56E+02)-	(2.82E+02)-	7.82E+02
F12	平均值	2.38E-08	2.30E+00	8.02E-01	4.72E-01	5.09E-01	7.87E-01	8.50E-01	3.91E-01
	标准差	(1.34E-08) +	(2.94E-01)-	(1.27E-01)-	(5.78E-02)-	(4.15E-02)-	(9.00E-02)-	(1.05E-01)-	2.41E-01
F13	平均值	3.58E-02	3.67E-01	2.41E-01	3.01E-01	2.52E-01	2.76E-01	1.85E-01	2.68E-01
	标准差	(1.16E-02) +	(6.41E-02)-	(3.33E-02) +	(3.22E-02)-	(4.68E-02) +	(4.40E-02) ≈	(2.39E-02) +	6.77E-02
F14	平均值	4.06E-01	2.74E-01	2.29E-01	2.64E-01	2.85E-01	2.73E-01	2.59E-01	2.63E-01
	标准差	(4.32E-02) -	(3.64E-02) ≈	(3.51E-02) +	(3.10E-02) ≈	(7.90E-02)-	(3.01E-02) ≈	(3.00E-02) ≈	4.45E-02
F15	平均值	1.33E+00	1.53E+01	3.88E+00	5.63E+00	5.51E+00	8.07E+00	8.87E+00	5.07E+00
	标准差	(7.36E-01) +	(2.63E+00)-	(1.19E+00) +	(6.20E-01)-	(6.06E-01)-	(5.89E-01)-	(9.03E-01)-	2.60E+00
F16	平均值	1.21E+01	1.28E+01	1.10E+01	9.79E+00	1.11E+01	1.09E+01	1.16E+01	9.53E+00
	标准差	(8.47E-01)-	(1.92E-01)-	(2.67E-01)-	(3.38E-01) ≈	(4.63E-01)-	(3.39E-01)-	(4.39E-01)-	1.08E+00
F17	平均值	5.18E+02	1.31E+03	1.41E+04	9.48E+02	3.74E+04	5.54E+02	4.24E+02	5.37E+02
	标准差	(1.73E+02) ≈	(3.35E+02)-	(1.18E+04)-	(8.94E+02)-	(3.34E+04)-	(2.32E+02) ≈	(1.31E+02) +	4.23E+02
F18	平均值	2.24E+01	4.27E+01	4.86E+02	2.01E+01	1.01E+03	2.52E+01	2.33E+01	1.38E+01
	标准差	(1.78E+01) -	(1.64E+01)-	(5.14E+02)-	(1.05E+01)-	(2.88E+03)-	(1.35E+01)-	(3.51E+00)-	8.60E+00
F19	平均值	6.94E+00	3.73E+00	4.50E+00	4.42E+00	1.33E+01	3.82E+00	4.55E+00	2.68E+00
	标准差	(2.31E+00) -	(9.54E-01)-	(1.33E+00)-	(8.00E-01)-	(1.21E+00)-	(4.90E-01)-	(3.94E-01)-	5.43E-01
F20	平均值	3.63E+00	1.56E+01	9.49E+01	1.24E+01	7.67E+01	1.71E+01	1.67E+01	1.13E+01
	标准差	(1.52E+00) +	(9.74E+00)-	(1.00E+02)-	(3.79E+00)-	(2.29E+02)-	(5.14E+00)-	(2.01E+00)-	5.55E+00
F21	平均值	2.93E+02	4.71E+02	3.30E+03	2.17E+02	8.28E+03	2.27E+02	2.97E+02	3.69E+02
	标准差	(1.24E+02) +	(3.14E+02)-	(4.22E+03)-	(1.33E+02) +	(1.23E+04)-	(1.25E+02) +	(1.02E+02) +	2.98E+02
F22	平均值	8.94E+02	2.00E+02	1.39E+02	1.27E+02	2.07E+02	1.03E+02	1.38E+02	3.48E+02
	标准差	(3.03E+02) -	(1.40E+02) +	(6.21E+01) +	(6.45E+01) +	(8.45E+01) +	(5.67E+01) +	(5.30E+01) +	2.24E+02
F23	平均值	3.14E+02	3.15E+02	3.15E+02	3.15E+02	3.14E+02	3.15E+02	3.15E+02	3.15E+02
	标准差	(0.00E+00) ≈	(2.19E-13) ≈	(2.11E-13) ≈	(5.78E-14) ≈	(1.05E-12) ≈	(5.78E-14) ≈	(5.78E-14) ≈	2.23E-13
F24	平均值	5.60E+02	2.14E+02	2.26E+02	2.24E+02	2.28E+02	2.25E+02	2.25E+02	2.00E+02
	标准差	(4.95E+02) -	(1.10E+01)-	(1.14E+00)-	(1.11E+00)-	(4.73E+00)-	(1.46E+00)-	(2.07E+00)-	3.06E-02
F25	平均值	2.07E+02	2.03E+02	2.09E+02	2.03E+02	2.00E+02	2.00E+02	2.03E+02	2.00E+02
	标准差	(1.99E+01) ≈	(2.27E-01) ≈	(1.53E+00) ≈	(5.61E-01) ≈	(1.25E-01) ≈	(3.88E-02) ≈	(5.04E-02) ≈	2.42E-13
F26	平均值	1.50E+02	1.00E+02	1.07E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
	标准差	(1.43E+02) -	(6.42E-02) ≈	(2.53E+01)-	(4.52E-02) ≈	(4.45E-02) ≈	(4.17E-02) ≈	(2.56E-02) ≈	6.90E-02
F27	平均值	7.36E+02	3.94E+02	4.15E+02	3.39E+02	8.22E+02	3.36E+02	3.00E+02	4.01E+02
	标准差	(3.40E+02) -	(2.56E+01) ≈	(5.07E+01) ≈	(4.86E+01) +	(1.47E+02)-	(4.54E+01) +	(2.88E-11) +	1.67E-01
F28	平均值	7.04E+03	8.17E+02	8.93E+02	7.95E+02	3.97E+02	8.03E+02	8.46E+02	8.21E+02
	标准差	(2.95E+03) -	(2.63E+01) ≈	(3.28E+01)-	(1.68E+01) ≈	(1.43E+01) +	(3.31E+01) ≈	(1.34E+01) ≈	2.51E+01
F29	平均值	2.04E+02	4.82E+02	1.15E+03	8.07E+02	2.14E+02	7.18E+02	7.16E+02	5.65E+02
	标准差	(3.54E-01) +	(2.58E+02) +	(1.78E+02)-	(1.00E+02)-	(1.23E+00) +	(7.73E+01)-	(2.39E+00)-	2.62E+02
F30	平均值	4.89E+02	4.05E+02	1.50E+03	1.25E+03	5.84E+02	1.10E+03	2.23E+03	4.21E+02
	标准差	(1.80E+02) -	(2.24E+01) ≈	(5.87E+02)-	(5.01E+02)-	(1.68E+02)-	(4.56E+02)-	(7.76E+02)-	4.35E+01
-		15	14	16	12	16	13	13	
+		8	5	7	8	8	7	8	
≈		7	11	7	10	6	10	9	

表2 参数灵敏度分析
Table 2 Parameter sensitivity analysis

序号	G_{change}	G_{change}	G_{change}	ρ	ρ	ρ	NP_{DE}	NP_{DE}	NP_{DE}	标准
	500	1500	2000	0.2	0.4	0.5	30	70	90	CMADE
F1	1.85E+05-	1.35E+05≈	1.15E+05+	1.99E+05+	1.45E+05+	1.47E+05+	1.55E+05+	1.87E+05+	2.55E+05+	1.34E+05
F2	0.00E+00≈	0.00E+00≈	0.00E+00≈	0.00E+00≈	0.00E+00≈	0.00E+00≈	2.10E-06+	0.00E+00≈	1.36E-05+	0.00E+00
F3	0.00E+00≈	0.00E+00≈	0.00E+00≈	0.00E+00+	0.00E+00≈	0.00E+00≈	5.70E+00+	0.00E+00≈	0.00E+00≈	0.00E+00
F4	1.02E+01-	7.46E+00-	3.42E+00≈	3.26E+00-	1.27E+00-	1.01E+01+	5.10E+00+	5.61E+00+	3.76E+00+	3.49E+00
F5	2.05E+01≈	2.05E+01≈	2.05E+01≈	2.05E+01≈	2.05E+01≈	2.05E+01≈	2.06E+01≈	2.05E+01≈	2.08E+01≈	2.05E+01
F6	3.90E-01≈	2.21E-01≈	1.98E-01≈	4.20E-01+	2.56E-01+	2.52E-01+	1.63E+00+	1.86E-02-	2.61E-01+	2.23E-01
F7	9.04E-04-	0.00E+00≈	4.93E-04-	0.00E+00≈	2.47E-04+	5.75E-04+	3.29E-03+	0.00E+00≈	0.00E+00≈	0.00E+00
F8	2.63E+01-	2.04E+01+	1.88E+01+	2.25E+01≈	2.29E+01≈	2.15E+01≈	2.88E+01+	3.08E+01+	1.01E+02+	2.26E+01
F9	4.13E+01-	3.07E+01+	2.27E+01+	3.85E+01+	4.40E+01+	3.75E+01≈	4.20E+01+	8.27E+01+	1.81E+02+	3.64E+01
F10	3.92E+02+	3.23E+02+	3.79E+02+	3.93E+02-	4.45E+02≈	4.10E+02-	4.82E+02+	6.53E+02+	2.65E+03+	4.42E+02
F11	3.18E+03-	3.19E+03-	2.81E+03+	3.05E+03≈	3.04E+03≈	2.93E+03≈	3.42E+03+	2.90E+03≈	5.23E+03+	3.02E+03
F12	4.15E-01-	4.14E-01-	3.56E-01+	3.51E-01-	3.53E-01-	3.54E-01-	4.40E-01+	5.74E-01+	1.19E+00+	3.91E-01
F13	2.95E-01-	2.84E-01-	3.13E-01-	2.56E-01≈	2.07E-01-	2.24E-01-	3.23E-01+	3.81E-01+	4.07E-01+	2.68E-01
F14	2.72E-01≈	2.55E-01≈	2.57E-01≈	2.55E-01≈	2.49E-01-	2.63E-01≈	3.06E-01+	2.71E-01≈	2.78E-01+	2.63E-01
F15	5.49E+00-	5.58E+00-	8.10E+00-	5.61E+00+	3.80E+00-	4.87E+00≈	5.35E+00+	1.50E+01+	1.62E+01+	5.07E+00
F16	9.52E+00≈	9.42E+00≈	9.59E+00≈	9.31E+00≈	9.20E+00≈	9.08E+00≈	1.02E+01+	1.08E+01+	1.19E+01+	9.53E+00
F17	6.28E+02-	5.86E+02-	7.32E+02-	6.52E+02+	6.37E+02+	5.05E+02-	9.82E+03+	1.34E+03+	1.69E+03+	5.37E+02
F18	1.45E+01≈	8.88E+00+	1.47E+01≈	1.29E+01-	1.33E+01≈	1.22E+01-	2.62E+01+	4.50E+01+	5.99E+01+	1.38E+01
F19	2.61E+00≈	2.51E+00+	3.00E+00-	2.51E+00-	2.48E+00-	2.36E+00-	3.04E+00+	3.98E+00+	4.85E+00+	2.68E+00
F20	1.32E+01-	1.20E+01-	1.12E+01≈	1.15E+01≈	1.16E+01≈	8.69E+00-	7.01E+01+	1.37E+01+	4.27E+01+	1.13E+01
F21	3.42E+02+	4.20E+02-	4.52E+02-	3.45E+02-	4.04E+02+	3.67E+02≈	7.83E+02+	3.17E+02-	9.30E+02+	3.69E+02
F22	3.72E+02-	3.39E+02≈	3.56E+02-	3.33E+02≈	2.91E+02-	4.20E+02+	3.63E+02≈	3.57E+02≈	2.84E+02-	3.48E+02
F23	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02≈	3.15E+02
F24	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02
F25	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02≈	2.00E+02
F26	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02≈	1.00E+02
F27	3.97E+02≈	4.01E+02≈	3.97E+02≈	4.01E+02≈	3.97E+02≈	4.01E+02≈	3.92E+02≈	4.01E+02≈	4.01E+02≈	4.01E+02
F28	8.34E+02≈	8.20E+02≈	8.21E+02≈	8.17E+02≈	8.27E+02≈	8.21E+02≈	8.59E+02≈	8.15E+02≈	8.18E+02≈	8.21E+02
F29	6.29E+02-	5.35E+02+	5.26E+02+	6.43E+02+	5.93E+02+	6.00E+02+	1.05E+03+	3.49E+02-	3.48E+02-	5.65E+02
F30	4.42E+02-	4.42E+02-	4.22E+02≈	4.50E+02+	4.17E+02≈	4.21E+02≈	6.80E+02+	5.95E+02+	9.84E+02+	4.21E+02
-	14	9	8	8	7	7	22	15	19	
+	2	6	7	6	7	7	0	2	2	
≈	14	15	15	16	16	16	8	13	9	

从表2得出的结果可看出,在与不同参数 G_{change} 的版本下,标准CMADE与其他版本在F2, F3, F5, F6, F14, F16, F23-F28这12个基准函数上性能几乎一致。在F13, F15, F17这3个基准函数中均比其他版本的性能更具有优势,而在F10基准函数中表现劣势。值得注意的是,在组合函数F23-F30中,标准CMADE与其他版本的性能表现非常接近且略胜一筹,好、差和相似的比例为4:2:18。分析可得,选择 G_{change} 为1000是较合适的。 G_{change} 对CMADE算法的影响可解释为:当 G_{change} 设置过小时,DE算法还未能寻找到全局解的大致范围,因此局部搜索效果不佳,造成资源浪费,而当 G_{change} 设置过大时,DE和CMA-ES算法不能达到良好的配合效果。

随着参数 ρ 数值的变化,标准CMADE与其他版本的CMADE相比, F2, F3, F8, F11, F16, F23-F28基本保持一致,在其他基准函数表现均有差异,表明针对不同的问题,更改 ρ 的参数值对CMADE的性能具有一定的影响,同时通过实验结果也可以看出,在组合函数F23-F30的表现上除F29和F30差异较大以外,其余表现几乎一致。

在不同参数值 NP_{DE} 下, CMADE性能表现差异非

常明显,标准CMADE与其他版本相比,性能更优的基准函数个数分别为20, 15, 19性能更差的个数分别为0, 2, 2, F5, F23-F28这7个基准函数在不同的 NP_{DE} 参数设置中性能表现相似, F1-F4, F7-F20, F30这19个基准函数上明显比其他版本有具有优势,表明参数值 NP_{DE} 对CMADE算法的性能影响很大,而且随着 NP_{DE} 数值越大,其性能下降尤为明显。同样, NP_{DE} 取值过小($NP_{\text{DE}}=30$),对其性能影响依然很大,因此 NP_{DE} 应保持在较小规模下的合理区间。对参数 NP_{DE} 解释如下:DE的种群规对于DE算法的效果有着较大的影响,本文DE算法主要的作用是找到解的大致范围,即进行全局搜索,因此可以设置较小的种群规模以节省资源。

从整体上来看,在CEC2014测试函数中,有一半左右的函数对于参数 G_{change} 和 ρ 不敏感,在另外一半函数上, G_{change} 和 ρ 会对CMADE算法的性能产生影响。而对于参数 NP_{DE} ,在大部分函数上均会对算法的性能产生影响,因此可以对这3个参数进行进一步研究。

4.4 贡献率评估

在CMADE中,DE算法用作全局搜索,而CMA-ES算法用作局部搜索,因此对着2个子算法的贡献率

行评估具有重要意义. 本文选取了4个具有代表性的基准函数F6, F11, F15, F27. 图2展示了算法进化过程中, DE和CMA-ES的解更新比率的变化曲线, 以及DE和CMA-ES算法各自的收敛曲线. 解更新比率是指迭代至当前代数中, 各子算法所作用的解下降区间

占目前解的总下降区间的比率. 算法的参数设置为 $G_{change}=1000$, $G_{CMAES}=20$. 当DE全局搜索每1000代后, 利用CMA-ES进行局部搜索, 搜索代数数为20代. 因此, 在该参数设置下, 算法迭代1020次为一个周期, DE迭代共6个周期, CMA-ES迭代5个周期.

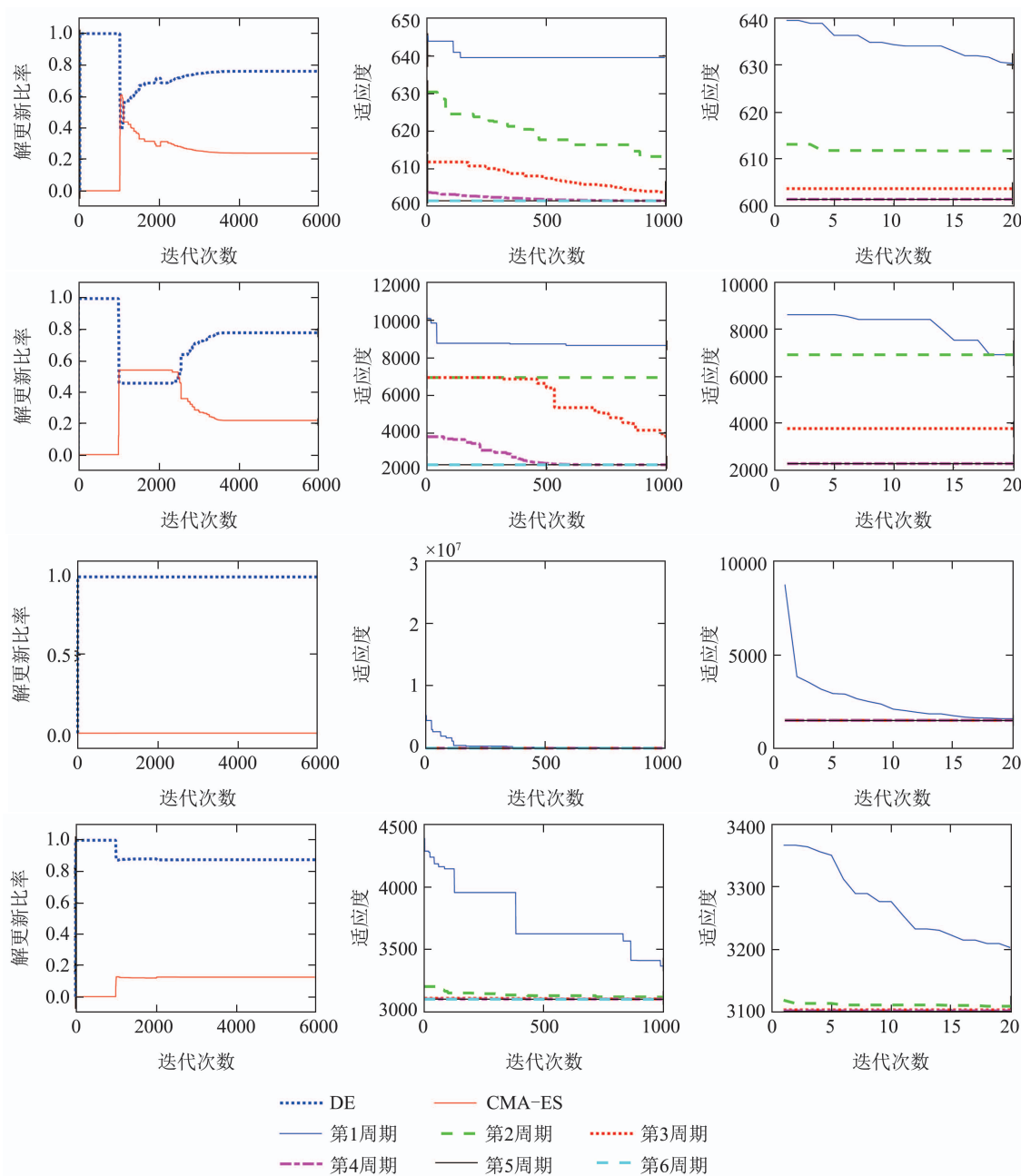


图 2 4个代表函数中子算法的贡献率及收敛曲线图

Fig. 2 Contribution rate and sub-algorithm convergence curve of 4 representative functions

从图2可以看出, 在CMAD算法中, 首先将种群赋予DE进行全局搜索, 因此在算法初期, CMA-ES算法不参与迭代, 解更新比率为0. 在CMA-ES参与搜索后, 解更新比率开始有所改变. 从总体上看, DE的贡献率普遍大于CMA-ES, 这是因为DE的资源分配高于CMA-ES, 且CMA-ES用作局部搜索, 因此其搜索范围有限. 然而CMA-ES算法下降速度快, 能够快速

找到了局部最优, 大大的减少了DE的搜索资源, 可见CMA-ES的局部搜索对CMAD算法起着重要的作用. 从图2可以看出, CMA-ES算法能够快速下降且达到局部最优, 因此给CMA-ES分配资源分配占比较少是合理的.

5 结论

局部搜索和全局搜索的反馈控制是优化算法十分

重要部分,在多模式的复杂问题当中,需要平衡空间中全局和局部的搜索,一般而言常采用的有2种方法.一是通过各种参数设置来控制勘探和开发之间的平衡;另一种则是开发具有不同机制的混合技术,来进行局部搜索和全局搜索的控制和反馈,本文采用了后一种方法. DE作为全局优化算法,拥有强大的全局搜索能力,并且有着结构简单的优点,然而同样存在收敛速度慢,效率低的问题. CMA-ES算法作为最强大的进化策略之一,以在广泛的单峰函数范围内快速收敛到最优值而闻名. 然而它十分容易陷入局部最优,具有收敛速度快、无梯度下降等特点. 因此本文提出了CMADE算法,采用DE算法进行全局搜索,寻找全局最优解的大致范围,而CMA-ES算法进行局部搜索,快速搜索到该范围的局部最优,大大的节省了资源,提升效率. CMADE算法采用了解交换的策略,保证DE搜索的全局性,以此不断进化. 本文最后将CMADE与具有代表性的7个算法,包括CMA-ES, DE, SaDE, jDE, EPSDE, ACO, SHADE, 将这些算法在CEC2014测试集中的30个测试函数上进行30维的对比. 实验结果证明, CMADE算法性能表现较好,具有一定的竞争力.

参考文献:

- [1] VIKHAR P A. Evolutionary algorithms: a critical review and its future prospects. *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGT-SPICC)*. Jalgaon: IEEE, 2016: 261 – 265.
- [2] ALAM T, QAMAR S, DIXIT A, et al. Genetic algorithm: reviews, implementations, and applications. *International Journal of Engineering Pedagogy*, 2020, 10(6): 57 – 77.
- [3] LEE K Y, YANG F F. Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming. *IEEE Transactions on Power Systems*, 1998, 13(1): 101 – 108.
- [4] LI Z, LIN X, ZHANG Q, et al. Evolution strategies for continuous optimization: a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 2020, 56.
- [5] STORN R, PRICE K. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.
- [6] HANSEN N. The CMA evolution strategy: a comparing review. *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms*. Berlin, Heidelberg: Springer, 2006, 192: 72 – 102.
- [7] TANABE R, FUKUNAGA A. Success-history based parameter adaptation for differential evolution. *IEEE Congress on Evolutionary Computation*. Cancun: IEEE, 2013: 71 – 78.
- [8] BREST J, GREINER S, BOSKOVIC B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 646 – 657.
- [9] OMRAN M G H, SALMAN A. Constrained optimization using CODEQ. *Chaos Solitons & Fractals*, 2009, 42(2): 662 – 668.
- [10] ZHANG J Q, SANDERSON A C. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945 – 958.
- [11] WU G H, MALLIPEDDI R, SUGANTHAN P N, et al. Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, 2016, 329: 329 – 345.
- [12] MOHAMED A W, HADI A A, FATTOUH A M, et al. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. *IEEE Congress on Evolutionary Computation (CEC)*, San Sebastian: IEEE, 2017: 145 – 152.
- [13] MALLIPEDDI R, SUGANTHAN P N, PAN Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 2011, 11(2): 1679 – 1696.
- [14] QIN A K, HUANG V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 398 – 417.
- [15] WU G H, PENG W X, HU X C, et al. Configuring differential evolution adaptively via path search in a directed acyclic graph for data clustering. *Swarm and Evolutionary Computation*, 2020, 55.
- [16] DU Yongzhao, FAN Yuling, LIU Peizhong, et al. Multi-populations covariance learning differential evolution algorithm. *Journal of Electronics & Information Technology*, 2019, 41(6): 1488 – 1495. (杜永兆, 范宇凌, 柳培忠, 等. 多种群协方差学习差分进化算法. 电子与信息学报, 2019, 41(6): 1488 – 1495.)
- [17] XIAO Peng, ZOU Dexuan, ZHANG Qiang. Efficient dynamic self-adaptive differential evolution algorithm. *Computer Science*, 2019, 46(6A): 124 – 132. (肖鹏, 邹德旋, 张强. 一种高效动态自适应差分进化算法. 计算机科学, 2019, 46(6A): 124 – 132.)
- [18] ZOU Jie, LI Jun. Multi-strategy covariance matrix learning differential evolution algorithm. *Computer Engineering and Applications*, 2021, 57(7): 78 – 87. (邹杰, 李俊. 多策略协方差矩阵学习差分进化算法. 计算机工程与应用, 2021, 57(7): 78 – 87.)
- [19] CARAFFINI F, LACCA G, NERI F, et al. A CMA-ES super-fit scheme for the re-sampled inheritance search. *IEEE Congress on Evolutionary Computation*. Cancun: IEEE, 2013: 1123 – 1130.
- [20] MELO V V, LACCA G. A modified covariance matrix adaptation evolution strategy with adaptive penalty function and restart for constrained optimization. *Expert Systems with Applications*, 2014, 41(16): 7077 – 7094.
- [21] WU G H, MALLIPEDDI R, SUGANTHAN P N. Ensemble strategies for population-based optimization algorithms – a survey. *Swarm and Evolutionary Computation*, 2018, 44: 695 – 711.
- [22] HANSEN N, OSTERMEIER A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 2001, 9(2): 159 – 195.
- [23] AUGER A, HANSEN N. A restart CMA evolution strategy with increasing population size. *IEEE Congress on Evolutionary Computation*. Edinburgh, Scotland: IEEE, 2005: 1769 – 1776.
- [24] ALRASHDI Z, SAYYAFZADEH M. ($\mu + \lambda$) Evolution strategy algorithm in well placement, trajectory, control and joint optimisation. *Journal of Petroleum Science and Engineering*, 2019, 177: 1042 – 1058.

作者简介:

杨紫晴 硕士研究生, 目前研究方向为智能优化算法, E-mail: maimai@csu.edu.cn;

姚加林 副教授, 硕士生导师, 目前研究方向为智能优化算法, E-mail: yaojialn@csu.edu.cn;

伍国华 教授, 博士生导师, 目前研究方向为智能优化与决策、计算智能, E-mail: guohuawu@csu.edu.cn;

陈学伟 硕士研究生, 目前研究方向为智能优化算法, E-mail: 184212057@csu.edu.cn;

毛成辉 副教授, 硕士生导师, 目前研究方向为智能优化算法, E-mail: mchh99@csu.edu.cn.