

一种动态自适应差分进化算法

李章维 周晓根 张贵军

(浙江工业大学信息工程学院 杭州 310023)

摘要 针对差分进化算法对参数和策略选择敏感而引起的收敛速度、计算代价及可靠性问题,结合抽象凸理论,提出一种动态自适应差分进化算法(DADE)。首先,通过对种群中的个体构建下界支撑面,建立目标函数的下界估计松弛模型;然后,利用下界估计松弛模型计算策略池中各策略生成的新个体的下界估计信息,进而根据下界估计信息及前期的进化经验动态自适应调整策略及其参数,并指导种群更新;最后,根据进化结果更新下界支撑面。6个标准测试函数的数值实验结果表明了所提算法的有效性。

关键词 差分进化,自适应,抽象凸,下界估计,全局优化

中图法分类号 TP18 文献标识码 A

Dynamic Adaptive Differential Evolution Algorithm

LI Zhang-wei ZHOU Xiao-gen ZHANG Gui-jun

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract To solve the problems of convergence speed, computational cost and reliability caused by the choice of parameters and strategies, a dynamic adaptive differential evolution algorithm was proposed in this paper incorporating the abstract convexity theory. Firstly, an underestimate relaxed model of the objective function is built by constructing the supporting hyperplanes for the individuals of the population. Then, the underestimate values of the trial individuals that generated by the strategies in the strategies pool can be obtained from the underestimate relaxed model. So the parameters and strategies can adjust adaptively according to the underestimate and the evolutionary experience before. In addition, the underestimate also can be used to guide the update process. Finally, the underestimate supporting hyperplanes are updated according to the result of evolutionary. Numerical experiment results of the six benchmark problems verify the effectiveness of the proposed algorithm.

Keywords Differential evolution, Self-adaption, Abstract convex, Underestimate, Global optimization

1 引言

进化算法是模拟生物界的进化过程而产生的一种现代化随机搜索方法,已被成功用于求解各种优化问题。典型的进化算法包括差分进化算法(DE)^[1]、遗传算法(GA)^[2]、进化规划(EP)^[3]、进化策略(ES)^[4]及粒子群算法(PSO)^[5]。这些算法不需要导数信息,对函数的性态没有要求,而且适用范围广、鲁棒性强。

DE算法^[1]作为一种随机性全局优化算法,是由Storn和Price在1997年提出的,已被证明是进化算法中简单而最高效的算法。DE算法具有算法通用、不依赖于问题信息、原理简单、易于实现、记忆个体最优解和种群内信息共享以及较强的全局收敛能力等特点,并在电力系统、机械工程、通信、化工及生物信息学等领域得到了广泛应用^[6,7]。然而,相关文献研究表明,DE算法的性能高度依赖于新个体生成策略(如变异、交叉操作)及其对应的控制参数(如种群规模 N_p 、增益常

数 F 和交叉概率 CR)^[8]。由于不同的优化问题有着不同的数学特性,不恰当的新个体生成策略及参数不仅会导致算法计算代价较大(如函数评价次数)、收敛速度较慢,而且极易趋于早熟收敛而陷于局部最优解,此外,在进化过程中的不同阶段采用不同的进化策略及参数对提高算法的性能至关重要^[9]。

针对以上问题,国内外学者相继提出了一些策略和参数自适应机制。Qin等^[8]利用均匀分布对变异和交叉率进行了调整,提出一种自适应差分进化算法(SaDE),即通过一种学习过程来自适应调整变异策略及参数,从而使得进化过程的不同阶段自适应地调整新个体生成策略及参数,并引入拟牛顿局部搜索,以增强算法的全局搜索能力。Zhang等^[10]提出一种带有外部最优存档的自适应差分进化算法(JADE),即对搜索过程中最优解进行存档,并自适应地调整变异率和交叉率。Wang等^[11]提出一种具有复合新个体生成策略和控制参数的差分进化算法(CoDE),即通过不同的生成策略与不同的

本文受国家自然科学基金(61075062),浙江省自然科学基金(LY13F030008),浙江省科技厅公益项目(2014C33088),浙江省重中之重学科开放基金(20120811),杭州市产学研合作项目(20131631E31)资助。

李章维(1967—),男,博士,副教授,主要研究方向为智能信息处理;周晓根(1987—),男,博士,主要研究方向为智能优化;张贵军(1974—),男,博士,教授,主要研究方向为智能信息处理、全局优化理论及算法设计、生物信息学, E-mail: zgj@zjut.edu.cn(通信作者)。

控制参数随机组合来竞争产生新个体。Mallipeddi 等^[9]提出一种具有系综变异策略和参数的差分进化算法 (EPSDE), 即在进化过程中的不同阶段, 对种群中的每个个体设置一组不同的变异策略及与其对应的参数来竞争产生后代。Elsayed 等^[12]提出一种新的自适应差分进化算法 (ISAMODE-CMA), 即将种群分成很多亚种群, 对每个亚种群设置相应的进化策略和交叉参数, 并引入一种学习策略来自适应调整每个亚种群的种群规模, 同时利用协方差适应矩阵来加快算法的收敛速度。以上改进算法均取得了一定的效果。

本文结合抽象凸理论^[13], 提出一种动态自适应差分进化算法 (DADE)。首先设置一组新个体生成策略池, 并对初始种群中的每个新个体建立抽象凸下界支撑面, 从而形成原目标函数的下界估计松弛模型, 进而利用下界估计模型计算各策略生成的新个体的下界估计值, 从而可以根据下界估计信息及前期的进化经验来自适应调整新个体生成策略及其参数, 并指导种群更新, 有效减少目标函数评价次数, 最后, 根据进化结果更新下界支撑面, 使得下界估计模型不断逼近原目标函数。

2 基本 DE 算法

DE 算法通过种群内个体间的合作与竞争来实现对优化问题的求解, 其本质是一种基于实数编码的具有保优思想的贪婪遗传算法。与其他进化算法一样, DE 算法也包括初始化、变异、交叉和选择₄个操作。算法首先在定义域范围内随机生成初始种群, 然后通过变异和交叉操作产生新个体, 最后通过选择操作决定新个体是否能成功进入下一代。假设目标函数为 $f(x)$, $x = (x_1, x_2, \dots, x_N) \in R^N$, 可行域为 D , 则算法具体步骤如下。

1) 初始化

在目标问题的可行解空间 D 中随机初始化种群 $P = [x_1, x_2, \dots, x_{Np}]$, Np 为种群规模, $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,N}]$ 表示目标问题的解, N 为目标问题的维数。

2) 变异

根据式(1)对第 g 代的每一个目标个体 x_i^g 进行变异操作, 得到与其对应的变异个体 v_i^g , 即

$$v_i^g = x_a^g + F(x_b^g - x_c^g) \quad (1)$$

其中, $a, b, c \in \{1, 2, \dots, Np\}$ 为随机生成的互不相同且与 i 不同的整数; x_a^g 为父代基向量, $(x_b^g - x_c^g)$ 为父代差分向量; F 为增益常数。

3) 交叉

根据式(2)对 x_i^g 和式(1)生成的变异个体 v_i^{g+1} 进行交叉操作, 生成新个体 u_i^{g+1} , 即

$$u_{i,j}^{g+1} = \begin{cases} v_{i,j}^{g+1}, & \text{if } \text{rand}(j) \leq C_R \text{ or } j = \text{rnbr}(i) \\ x_{i,j}^g, & \text{Otherwise} \end{cases} \quad (2)$$

其中, $\text{rand}(j)$ 为 $[0, 1]$ 之间均匀分布的随机数; C_R 为范围在 $[0, 1]$ 之间的交叉概率; $\text{rnbr}(i)$ 为 $\{1, \dots, N\}$ 之间的随机数。

4) 选择

根据式(3)对新个体 u_i^{g+1} 和目标个体 x_i^g 的目标函数进行比较。对于最小化问题, 则选择目标函数值小的个体作为新种群的个体, 即

$$x_i^{g+1} = \begin{cases} u_i^{g+1}, & \text{if } f(u_i^{g+1}) < f(x_i^g) \\ x_i^g, & \text{Otherwise} \end{cases} \quad (3)$$

其中, f 为目标函数。

3 动态自适应差分进化算法

本文提出的 DADE 算法基于抽象凸理论, 通过建立原目标函数的下界估计松弛模型, 从而根据下界估计信息及前期的进化经验来自适应调整新个体生成策略及其参数, 并指导种群更新。

3.1 抽象凸下界估计松弛模型

抽象凸理论^[14-16]作为确定性全局优化的一个重要部分, 泛化了凸分析中仿射弱函数的概念。抽象凸方法则基于一类非凸函数都是它一系列简单弱函数(支撑函数)的上包络这一重要结论, 并引入次梯度这一有力分析工具, 利用一系列简单弱函数将目标问题转化一系列松弛模型, 从而可以通过松弛模型来估计原目标函数。

定义 1 若函数 f 满足如下两个条件:

$$\forall x, y \in R_+^N, x \geq y \Rightarrow f(x) \geq f(y) \quad (4)$$

$$\forall x \in R_+^N, \lambda \in R_{++}: f(\lambda x) = \lambda f(x) \quad (5)$$

则称函数 $f: R_+^N \rightarrow R$ 为正齐次递增函数 (Increasing Positively Homogeneous Functions of Degree One, IPH)^[16]。

定义 2 若存在一个最小的正数 M 使得函数 f 满足条件:

$$|f(x) - f(y)| \leq M \|x - y\|, \forall x, y \in D \quad (6)$$

则称函数 f 为 Lipschitz 连续, M 为其 Lipschitz 常数^[17]。

设如下全局优化问题:

$$\min f(x), x \in D \subset R^N \quad (7)$$

满足定义 2, 其中, $D = \{x_i | a_i \leq x_i \leq b_i, i = 1, 2, \dots, N\}$ 为可行域, $x = [x_1, x_2, \dots, x_N]^T$ 为边界约束可行域 D 上 N 维连续优化变量, $f(\cdot)$ 为定义在可行域 D 上的目标函数。通过如下线性变换将原优化变量 $x = [x_1, x_2, \dots, x_N]^T$ 转化到单位单纯形 $S(S = \{x \in R^{N+1}, x_i \geq 0, \sum_{i=1}^{N+1} x_i = 1\})$ 中:

$$\begin{cases} x_i' = (x_i - a_i) / \sum_{i=1}^N (b_i - a_i) \\ x_{N+1}' = 1 - \sum_{i=1}^N x_i' \end{cases}, i = 1, 2, \dots, N \quad (8)$$

其中, $x' = [x_1', x_2', \dots, x_{N+1}'] \in S$ 为线性转换变量。

对式(8)进行反变换, 可得到:

$$x_i = x_i' \sum_{i=1}^N (b_i - a_i) + a_i, i = 1, 2, \dots, N \quad (9)$$

将式(9)代入式(8), 可将原目标函数转化为:

$$\min \bar{f}(x'), x' \in S \subset R_+^{N+1} \quad (10)$$

其中, $\bar{f}(\cdot)$ 为定义在 $N+1$ 维单位单纯形区域 S 上的目标函数。

设 $I = \{1, 2, \dots, N+1\}$, 则根据抽象凸理论可知, 目标函数 $\bar{f}$ 关于如下函数(支撑函数)是抽象凸的^[17]:

$$h(x') = \min_{i \in I} l_i x_i' \quad (11)$$

其中, $l_i = \frac{\bar{f}(x')}{x_i'}$ 。由文献^[15]可知, 函数 $\bar{f}$ 加上常数 $C (C > -\min \bar{f}(x') + 2M)$ 可转化为 IPH 函数, 进而可以根据其 K 个

已知点的支撑函数 $h^k (k=1, 2, \dots, K)$ 建立下界估计松弛模型:

$$H^K(x') = \max_{k=1, \dots, K} h^k(x') = \max_{k=1, \dots, K} \min_{i \in I} l_i^k x_i' \quad (12)$$

其中向量

$$l^k = (\frac{\bar{f}(x'^k)}{x_1'^k}, \frac{\bar{f}(x'^k)}{x_2'^k}, \dots, \frac{\bar{f}(x'^k)}{x_{N+1}'^k}) \quad (13)$$

称为支撑向量。

3.2 策略自适应机制

由于不同优化问题的数学特性不同,在进化的各阶段需要的策略也不同,因此,本文的策略池中包括以下 4 种变异策略:

1) DE/rand/1

$$v_i^g = x_{r1}^g + F \cdot (x_{r2}^g - x_{r3}^g) \quad (14)$$

2) DE/best/1

$$v_i^g = x_{best}^g + F \cdot (x_{r1}^g - x_{r2}^g) \quad (15)$$

3) DE/current-to-best/1

$$v_i^g = x_i^g + F \cdot (x_{best}^g - x_i^g) + F \cdot (x_{r1}^g - x_{r2}^g) \quad (16)$$

4) DE/pbest/2

$$v_i^g = x_{pbest}^g + F \cdot (x_{r1}^g - x_{r2}^g) + F \cdot (x_{r3}^g - x_{r4}^g) \quad (17)$$

其中, $r1, r2, r3, r4 \in \{1, 2, \dots, Np\}$ 为随机生成的互不相同且与 i 不同的正数, x_{best}^g 表示第 g 代种群中的最优个体, ψ 表示 $[1, 0.5Np]$ 之间的随机正数, x_{pbest}^g 表示第 g 代中 ψ 个个体中的最优个体。

经过变异、交叉操作生成新个体后,根据下界估计松弛模型计算各策略生成的新个体的下界估计值。通过比较各下界估计值动态选择变异策略,而无需计算各新个体的目标函数值。以图 1 所示的一维问题为例, $T1, T2, T3$ 和 $T4$ 分别为策略 1)、2)、3) 和 4) 生成的新个体,根据式(12)计算出各新个体的下界估计值 $y_{T1}^u, y_{T2}^u, y_{T3}^u$ 和 y_{T4}^u ,如图 1 所示,因为 $T1$ 的下界估计值最小,则选择策略 1) 生成的新个体 $T1$ 为当前目标个体对应的新个体,同时还可以利用新个体的下界估计值指导种群更新,以减少目标函数评价次数。如图 1 所示,假设 A 为目标个体, $T1$ 为新个体,由于 $T1$ 的下界估计值 y_{T1}^u 大于 A 的目标函数值,因此无需计算新个体 $T1$ 的目标值就可以判定舍弃新个体 $T1$ 而保持 A 不变。

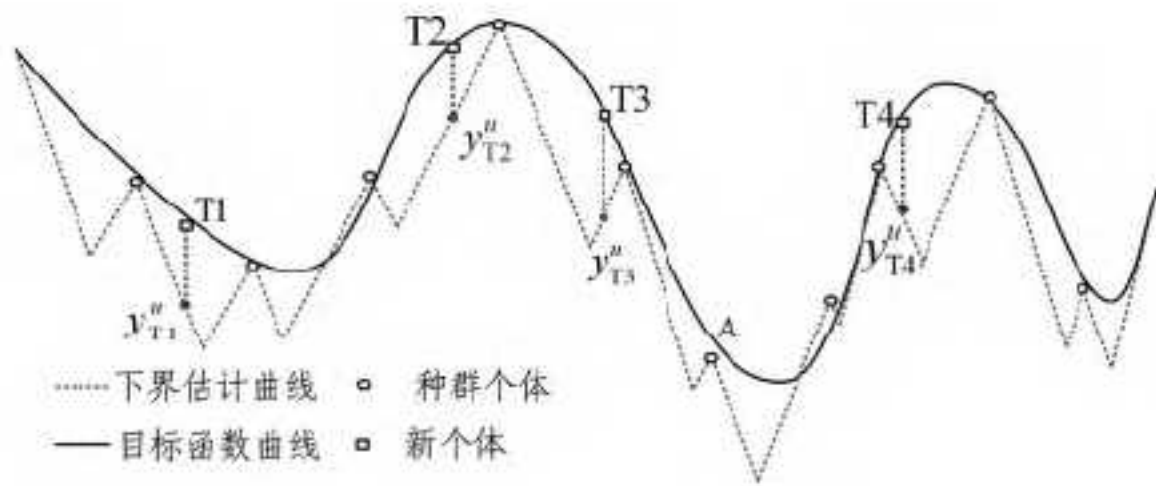


图 1 新个体下界估计示意图

3.3 参数自适应机制

在传统 DE 算法中,交叉概率 CR 及增益常数 F 的设置与问题的数学特性有着重要的关系,参数的设置直接影响着算法的性能。交叉概率 CR 通常与所求问题的性质有关(如多模态问题和单模态问题),其直接决定了当前种群中变异个体的数目;增益常数 F 则对算法的收敛速度有很大的影响。受文献[8]启发,DADE 算法采用以下参数自适应机制。

由于正态分布不仅具有规律性,而且带有随机性,因此,在 DADE 算法中,对于每个个体的增益常数 F ,均通过平均值为 0.5、标准偏差为 0.3 的正态分布函数随机生成,即 $F_i = N(0.5, 0.3)$ 。

相关文献研究表明[8], CR 的设置在很大程度上决定了算法求解的成功与否。事实上,对于给定的问题,合理的 CR 值应属于较小的范围。为了根据进化过程中的反馈信息自适应地调整 CR 的值,设置一个学习代数 (Learning Generation, L_G) 用于记录当前代的前 L_G 代中成功进入下一代的新个体对应的交叉概率 CR 。当第 k 个策略生成的新个体成功进入下一代时,则将其对应的 CR 记录到数组 C_{RM}^k 中,即当第 k 个策略生成的新个体的下界估计值小于其目标个体的目标函数值,且新个体的目标函数值也小于目标个体的函数值时,则记录其对应的 CR 值。从而通过取前 L_G 代 C_{RM}^k 中各元素的平均值来动态调整第 k 个策略在当前代的 C_{RM}^k 值。 C_{RM}^k 可通过如下公式计算:

$$C_{RM}^k = \frac{\sum_{g=G-L_G}^{G-1} \sum_{i=1}^{S_g^k} C_{RM,i,g}^k}{\sum_{g=G-L_G}^{G-1} S_g^k}, k=1, 2, \dots, T; G > L_G \quad (18)$$

其中, S_g^k 表示第 g 代中第 k 个策略生成的新个体成功进入下一代的数目。进而可以通过动态调整 C_{RM}^k 值来自适应调整 CR 值。对于第 k 个策略,其交叉概率 C_R^k 服从均值为 C_{RM}^k 、标准偏差为 $Std=0.1$ 的正态分布,即每个个体的 $C_{R,i,g}^k$ 为:

$$C_{R,i,g}^k = N(C_{RM}^k, Std), i=1, 2, \dots, Np \quad (19)$$

其中, C_{RM}^k 初始值为 0.5,且 Std 应设置得较小以保证 $CR \in [0, 1]$,因此将其设置为 0.1。

3.4 算法流程

以最小化问题为例,整个算法的基本流程如下。

1. 初始化:设置各参数值,并在定义域范围内随机生成初始种群;
2. 建立下界估计松弛模型,根据式(13)计算初始种群中各个体的支撑向量,从而形成原目标函数的下界估计松弛模型;
3. 生成新个体,根据参数自适应机制,利用各变异策略生成新个体,并根据式(12)计算出各新个体的下界估计值,通过策略自适应机制动态选择当前目标个体对应的新个体;
4. 利用下界估计信息指导种群更新,比较新个体的下界估计值与目标个体的函数值,如果新个体的下界估计值小于目标个体的函数值,则无需计算新个体的目标函数值,且保持目标个体不变,否则计算出新个体的目标函数值,通过与目标个体的函数值相比较,以确定新个体是否能够替换目标个体;
5. 更新下界估计模型,根据式(13)计算成功进入下一代的新个体的支撑向量,并更新下界估计模型;
6. 判断是否满足终止条件,如果满足则终止,否则继续步骤 3。

4 数值研究

4.1 测试函数及参数设置

为了验证所提算法的有效性,本文选取 6 个典型标准测

试函数进行优化验证。表 1 列出了各测试函数的表达式及参数,其中 f_1 和 f_2 为单模态函数, f_3-f_6 为多模态函数。

选用以下 5 种算法与所提算法进行比较。

- DE1:DE/rand/1/bin, $F=0.9,CR=0.1$;
- DE2:DE/rand/1/bin, $F=0.5,CR=0.5$;
- CoDE:参见文献[11];

- SaDE:参见文献[8];
- EPSDE:参见文献[9]。

DADE 算法设置,学习代数 $L_G=20$,增益常数和交叉概率初始值均设置为 0.5,均值 C_{km}^* 初始值为 0.5。此外,鉴于比较的合理性和公平性,6 种算法的种群规模均设置为 $N_p=30$,且各测试函数均独立运行 30 次。

表 1 标准测试函数

函数名	表达式	实验维数	搜索范围	全局最优值(解)
Sphere	$f_1(x) = \sum_{i=1}^N x_i^2$	10,30	$(-100,100)^N$	$0(0,\dots,0)$
Rosenbrock	$f_2(x) = \sum_{i=1}^{N-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	5,10	$(-30,30)^N$	$0(1,\dots,1)$
Griewank	$f_3(x) = 1 + \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos(\frac{x_i}{\sqrt{i}})$	10,30	$(-600,600)^N$	$0(0,\dots,0)$
Schaffer 2	$f_4(x) = \sum_{i=1}^{N-1} (x_i^2 + x_{i+1}^2)^{0.25} (\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1)$	10,30	$(-100,100)^N$	$0(0,\dots,0)$
Ackley	$f_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}) - \exp(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i)) + 20 + e$	10,30	$(-30,30)^N$	$0(0,\dots,0)$
Rastrigin	$f_6(x) = 10N + \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i))$	5,10	$(-5,5)^N$	$0(0,\dots,0)$

4.2 实验结果与分析

为了验证所提算法在计算代价和可靠性方面的优势,采用 30 次独立运行的平均目标函数评价次数 (Function Evaluations, FES) 和成功率 (Success Rate, SR) 两个指标来进行衡量,其中,成功率为成功求解次数与总运行次数之比。算法终

止条件为 $|f(x_{best}) - Opti| \leq 10^{-5}$,“Opti”为函数的全局最优值。结果见表 2,其中“null”表示在设定的最大目标函数评价次数内无法找到满足条件的最优值,“W”表示所提算法优于所比算法的个数,“T”表示所提算法等于所比算法的个数,“L”表示所提算法差于所比算法的个数。

表 2 所比算法的平均函数评价次数和成功率

函数	维数	DE1		DE2		CoDE		SaDE		EPSDE		DADE	
		FES	SR	FES	SR	FES	SR	FES	SR	FES	SR	FES	SR
f_1	30	39081	1.00	18913	1.00	39180	1.00	16315	1.00	11689	1.00	12873	1.00
	10	15581	1.00	5851	1.00	11911	1.00	4869	1.00	5202	1.00	2262	1.00
f_2	10	83569	1.00	null	0.00	38370	1.00	null	0.00	20715	0.93	21224	1.00
	5	29390	1.00	12529	0.33	13980	1.00	31042	0.83	9892	1.00	9596	1.00
f_3	30	44927	1.00	20873	1.00	44653	1.00	16824	0.57	12448	0.90	11856	1.00
	10	40271	1.00	22252	0.90	52131	1.00	16703	0.97	21833	0.93	9323	1.00
f_4	30	150465	1.00	77315	1.00	170040	1.00	50927	0.23	67677	0.93	52921	1.00
	10	61736	1.00	21794	1.00	61303	1.00	19431	0.97	24192	1.00	21342	1.00
f_5	30	54070	1.00	26160	1.00	56427	1.00	23962	0.40	16002	0.93	16451	1.00
	10	23493	1.00	8611	1.00	22963	1.00	7215	1.00	7358	1.00	5855	1.00
f_6	10	17539	1.00	22378	1.00	27136	1.00	11514	1.00	10692	1.00	6770	1.00
	5	11538	1.00	5821	1.00	10396	1.00	5730	1.00	4874	1.00	3591	1.00
平均值		47638	1.00	22045	0.85	45708	1.00	18593	0.75	17715	0.97	14505	1.00
W/T/L		12/0/0	0/12/0	12/0/0	3/9/0	12/0/0	0/12/0	10/0/2	7/5/0	9/0/3	5/7/0	-/-/-	-/-/-

从表 2 的最后一行可以看出,12(6×2)个测试问题中,在计算代价方面,DADE 算法优于 DE1、DE2 和 CoDE 算法所有问题,且分别优于 SaDE 和 EPSDE 算法 10 个和 9 个问题, SaDE 和 EPSDE 分别只在 2 个和 3 个问题上优于 DADE 算法。在可靠性方面,DADE、DE1 和 CoDE 算法对于所有问题的成功率均为 1,且与 DE2、SaDE 及 EPSDE 算法相比, DADE 分别优于 DE2、SaDE 及 EPSDE 算法 3 个、7 个和 5 个问题,对于其余问题,成功率均相同。其次,从 12 个问题的平均结果来看,DADE 算法的平均目标函数评价次数为 14505, DE1、DE2、CoDE、SaDE 和 EPSDE 算法分别为 47638、22045、45708、18593 和 17715,也就是说,DADE 算法的效率相对于 DE1、DE2、CoDE、SaDE 和 EPSDE 算法分别提升了 69.6%、34.2%、68.3%、22.0%和 18.1%;而且 DADE 算法可靠性也很高,与 DE1 和 CoDE 算法一样,平均成功率均为 1.00。其次是 EPSDE 算法,成功率为 0.97;DE2 算法为 0.85;SaDE 算

法,最低为 0.75。DE1 和 CoDE 算法虽然与 DADE 算法一样成功率均为 1,但是 DE1 和 CoDE 算法计算代价较大,平均目标函数评价次数为 DADE 算法的 3 倍多,相对于 DE 算法来说, SaDE 算法虽然在计算代价方面得到了改进,但是其可靠性较低。因此,只有 DADE 算法能够在减小计算代价的同时提高算法的可靠性。

以目标函数评价次数为终止条件,从平均值(Mean)和标准偏差(Standard deviation, Std)两方面进行比较分析以说明 DADE 算法在解的质量方面的优势。同时,为了验证 DADE 算法的优化结果相对于其它算法是否有显著性优势,采用 Wilcoxon Signed Rank Test^[18]对优化结果进行非参数假设检验(Sig),显著性水平为 0.05,结果见表 3,其中,“+”表示所提算法显著优于所比算法,“≈”表示所提算法与所比算法没有显著性差异。

表 3 所比算法的优化结果性能表

函数	维数	评价 次数	DE1		DE2		CoDE		SaDE		EPSDE		DADE	
			平均值 (标准差)	Sig	平均值 (标准差)	Sig	平均值 (标准差)	Sig	平均值 (标准差)	Sig	平均值 (标准差)	Sig	平均值 (标准差)	Sig
f_1	30	20000	4.92E-01 (1.63E-01)	+	3.00E-06 (1.78E-06)	+	7.34E-01 (3.69E-01)	+	7.68E-08 (1.15E-07)	+	4.64E-12 (1.09E-11)	+	9.28E-12 (1.10E-11)	
	10	10000	1.99E-02 (1.33E-02)	+	6.49E-12 (5.60E-12)	+	1.50E-02 (8.28E-03)	+	1.41E-14 (5.05E-14)	+	6.84E-14 (1.45E-13)	+	2.99E-17 (6.94E-17)	
f_2	10	30000	3.06E+00 (1.27E+00)	+	4.76E+00 (1.49E+00)	+	1.48E-01 (1.47E-01)	+	4.56E+00 (1.82E+00)	+	1.33E-01 (7.28E-01)	≈	3.11E-11 (8.66E-11)	
	5	20000	1.20E-02 (1.46E-02)	+	6.34E-01 (8.03E-01)	+	3.70E-07 (1.23E-06)	+	2.34E-01 (2.97E-01)	+	5.62E-16 (3.08E-15)	+	3.01E-21 (1.53E-20)	
f_3	30	30000	1.48E-02 (4.31E-03)	+	6.74E-10 (1.81E-09)	+	4.55E-02 (5.74E-02)	+	1.93E-02 (2.70E-02)	+	8.35E-03 (1.69E-02)	≈	2.90E-15 (5.88E-15)	
	10	30000	4.17E-03 (4.26E-03)	+	4.93E-04 (2.70E-03)	≈	5.86E-02 (2.29E-02)	+	6.57E-04 (2.50E-03)	≈	4.86E-07 (1.28E-06)	+	2.44E-12 (9.42E-12)	
f_4	30	100000	4.31E-02 (1.88E-02)	+	8.89E-09 (4.50E-09)	+	1.18E-01 (2.65E-02)	+	2.99E-01 (9.92E-01)	+	1.68E-01 (9.16E-01)	+	9.16E-16 (4.63E-16)	
	10	50000	1.23E-03 (7.58E-04)	+	2.03E-17 (1.75E-17)	≈	9.07E-04 (4.39E-04)	+	1.79E-04 (9.79E-04)	+	1.61E-15 (3.45E-15)	+	1.32E-17 (8.89E-18)	
f_5	30	50000	3.17E-05 (5.73E-06)	+	9.59E-12 (4.00E-12)	+	5.74E-05 (2.06E-05)	+	9.89E-01 (7.57E-01)	+	1.69E-01 (3.90E-01)	≈	2.24E-14 (8.95E-15)	
	10	20000	9.61E-05 (4.12E-05)	+	2.05E-14 (1.33E-14)	+	8.61E-05 (3.33E-05)	+	3.32E-15 (9.01E-16)	+	3.55E-15 (9.22E-15)	+	3.40E-15 (1.35E-15)	
f_6	10	15000	4.17E-04 (3.50E-04)	+	4.79E+00 (2.07E+00)	+	3.37E+00 (1.02E+00)	+	4.57E-10 (1.86E-09)	+	7.22E-10 (3.92E-09)	+	2.64E-13 (5.90E-13)	
	5	8000	1.39E-02 (1.34E-02)	+	2.27E-10 (7.64E-10)	+	1.21E-02 (1.34E-02)	+	9.62E-11 (2.92E-10)	+	2.03E-13 (6.82E-13)	≈	6.84E-14 (7.57E-14)	
+ / ≈			12/0		10/2		12/0		11/1		8/4		-/-	

对比表 3 中的数据可以看出,对于大多数测试问题,DADE 算法优化结果的两项性能均优于其它比较算法。对于问题 $f_1-10, f_2 \sim f_4, f_5-30$ 和 f_6 , DADE 算法的优化性能优于其它算法;对于问题 f_1-30 , EPSDE 算法优于其它算法;SADE 算法在问题 f_5-10 上优于其它算法;DE1、DE2 和 CoDE 不能在任何问题上优于其它算法。在显著性方面,从表 3 的最后一行可以看出,DADE 算法明显优于 DE1 和 CoDE 算法所有问题,且分别明显优于 DE2、SaDE 和 EPSDE 算法 10 个、11 个和 8 个问题,对于其它问题虽然没有显著性差异,但结果均优于其它算法。

评价次数为横坐标,目标函数值的对数为纵坐标,选取 4 个 30 维测试问题绘制 6 种算法的平均收敛曲线图并进行比较分析。从图 2(a)可以看出,对于函数 f_1 ,EPSDE 算法收敛速度最快,DADE 算法次之。图 2(b)和图 2(c)表明,6 种算法在对函数 f_3 和 f_4 进行优化求解时,DADE 算法收敛速度最快,DE2、SaDE 和 EPSDE 算法均陷入了局部最优而无法成功求解;DE1 和 CoDE 算法虽然没有陷入局部最优,但是收敛速度较慢。图 2(d)表明,对于函数 f_5 ,在算法初期时,EPSDE 算法收敛速度最快,SaDE 算法次之,但是两者很快就出现了早熟收敛,而 DADE 算法能够稳健地向着全局最优收敛,DE2 算法次之,CoDE 算法收敛速度最慢。

总体来说,本文所提出的 DADE 算法无论对于多模态函数还是单模态函数,所表现出的整体性能都显著优于其它算法,不仅计算代价小、可靠性高,而且收敛速度快。

结束语 基于抽象凸理论,提出一种动态自适应差分进化算法(DADE)。在 DADE 算法中,设置一组新个体生成策略,并通过构建个体的下界支撑面建立目标函数的下界估计松弛模型,从而利用下界估计松弛模型获取目标函数的下界估计信息,进而根据下界估计信息及前期的进化经验来自适应调整变异策略及参数,同时指导种群更新。6 个典型测试函数的数值实验结果表明,所提算法可以以廉价的计算代价求得质量较高(标准偏差小)的解,不仅可靠性高,而且收敛速度快,是一种有效的全局优化算法。

参考文献

- [1] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of global optimization, 1997, 11(4): 341-359

(下转第 74 页)

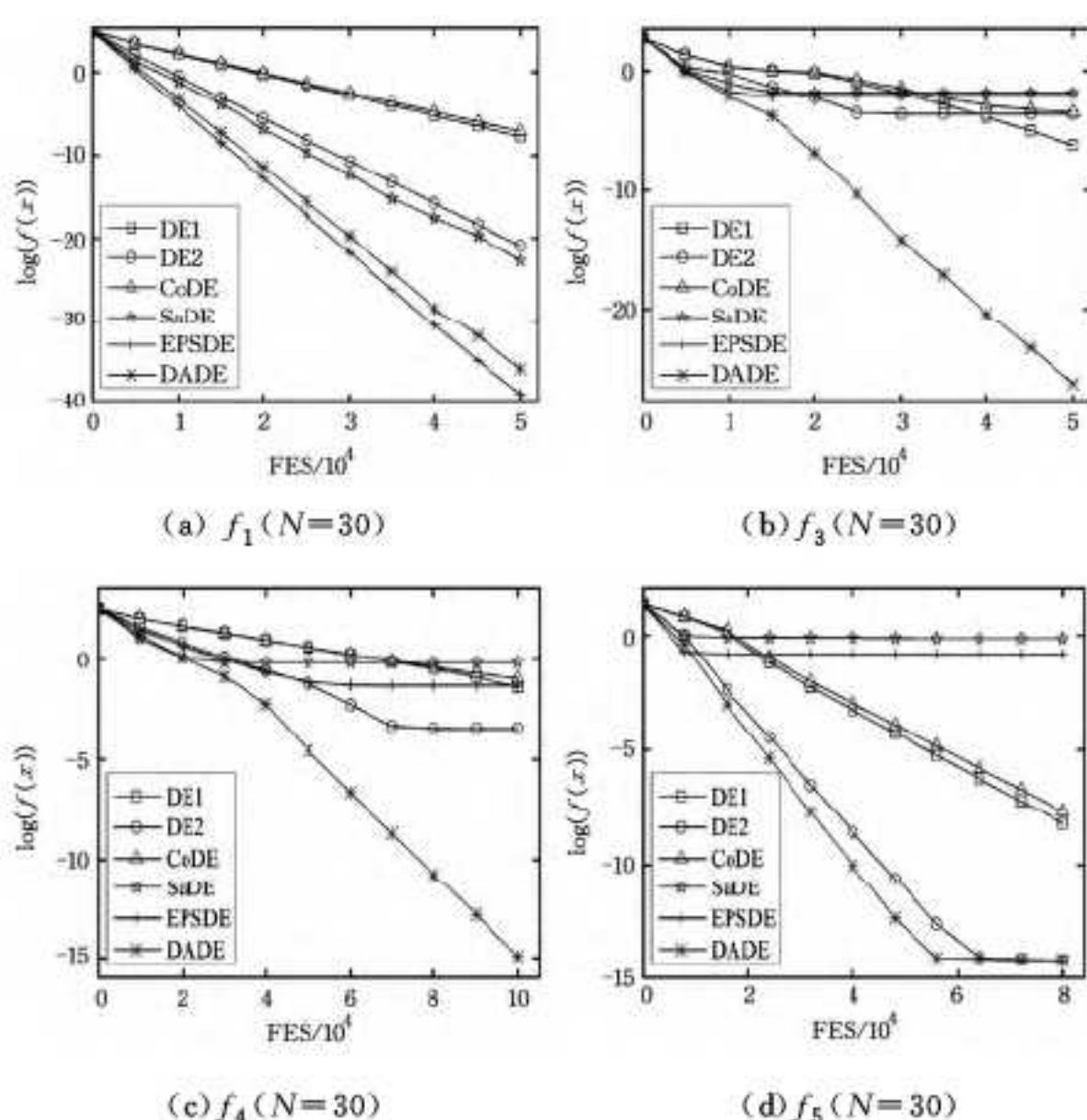


图 2 平均收敛曲线

为了验证所提算法在收敛速度方面的优势,以目标函数

$X_3 > X_1 > X_2 > X_4$, 即在各相关应急管理部门制定的现有网络舆情突发事件₄个备选应急决策方案中, 应急救援方案 X_3 的效果显然是最好的; 事实上我们也注意到, 应急决策方案 X_3 由于其应急风险和成本都很小, 可救援人数也较多, 其他应急效果指标也相对较好, 所以该方案通过加权集结各应急指标得出的综合效果区间评估值确实最优。因此, 最高决策者可要求相关应急操作部门最终选择执行方案 X_3 来应急处置当前网络舆情突发事件, 从而使得整个应急实施效果达到最优。

结束语 网络舆情突发事件应急决策方案的选择直接关系到不良网络舆情的引导调节和突发事件灾害损失的风险控制, 在危机处置过程中选择最优的方案能将应急成本降到最小且应急效果达到最优。传统应急方案优选一般依靠决策者和专家根据以往应急经验评估得出, 具有较强的主观性。本文提出了基于应急决策评估指标的区间信息熵来确定各指标的客观权重, 进而通过加权集结得到各备选应急方案的综合效果区间评估值, 并根据区间值可能优先度排序方法, 最终可得网络舆情突发事件各应急决策方案的优先排序, 从而选出效果最优的应急策略。

参 考 文 献

- [1] 陈兴, 王勇, 吴凌云, 等. 多阶段多目标多部门应急决策模型[J]. 系统工程理论与实践, 2010, 30(11): 1977-1985
- [2] 曾润喜. 网络舆情突发事件预警指标体系构建[J]. 情报理论与实践, 2010, 33(1): 77-80
- [3] 张一文, 齐佳音, 方滨兴, 等. 非常规突发事件网络舆情热度评价指标体系构建[J]. 情报杂志, 2010, 29(11): 71-75
- [4] 戴媛, 郝晓伟, 郭岩, 等. 基于多级模糊综合评判的网络舆情安全评估模型研究[J]. 信息安全, 2010(5): 60-62
- [5] 刘樑, 沈焱, 曹学艳, 等. 基于关键信息的非常规突发事件预警模型研究[J]. 管理评论, 2012, 24(10): 166-175
- [6] 张秋菊, 朱帮助. 基于熵权灰色关联分析的群体性突发事件预警模型[J]. 数学的实践与认识, 2011, 41(24): 25-32
- [7] 郭泳亨, 卢兴华, 刘云. 应急决策效果的模糊综合评判[J]. 科学技术与工程, 2006, 6(5): 588-592
- [8] 赵希男, 靖可, 陈周坤. 基于个体特征识别的应急决策有效性评价方法[J]. 系统工程, 2009, 27(7): 75-79
- [9] 左春荣, 田涛, 马英. 基于 Markov 链的非常规突发事件应急决策模型[J]. 统计与决策, 2012, 19: 57-60
- [10] 孔衍, 陈舜恒, 张锦, 等. 三标度 AHP 法在核应急决策中的应用[J]. 数学的实践与认识, 2013, 43(9): 109-114
- [11] 李明磊, 王红卫, 祁超, 等. 非常规突发事件应急决策方法研究[J]. 中国安全科学学报, 2012, 22(3): 158-162
- [12] 郭文帅, 寇刚, 彭怡, 等. 面向突发事件的模糊多目标应急决策方法[J]. 系统工程理论与实践, 2012, 32(6): 1298-1304
- [13] 靖可, 赵希男, 王艳梅. 基于区间偏好信息的不确定性应急局部群决策模型[J]. 运筹与管理, 2010, 19(2): 97-103
- [14] 徐泽水, 达庆利. 区间型多属性决策的一种新方法[J]. 东南大学学报, 2003, 33(4): 498-501
- [15] 徐泽水, 达庆利. 基于模糊语言评估的多属性决策方法[J]. 东南大学学报, 2002, 32(4): 656-658
- [16] Sanayei A, Farid Mousavi S, Yazdankhah A. Group decision making process for supplier selection with VIKOR under fuzzy environment[J]. Expert Systems with Applications, 2010, 37: 24-30
- [17] Burillo P, Bustince H. Entropy on intuitionistic fuzzy sets and on interval-valued fuzzy sets[J]. Fuzzy Sets and Systems, 1996, 78: 305-316
- [18] 王云华, 柯慧燕. 复杂性测度在非常规突发事件决策评估的应用[J]. 武汉理工大学学报, 2011, 33(6): 999-1002
- [19] Zhang J, Sanderson A C. JADE: adaptive differential evolution with optional external archive[J]. IEEE Trans on Evolutionary Computation, 2009, 13(5): 945-958
- [20] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters[J]. IEEE Trans on Evolutionary Computation, 2011, 15(1): 55-66
- [21] Elsayed S M, Sarker R A, Essam D L. An improved self-adaptive differential evolution algorithm for optimization problems[J]. IEEE Trans on Industrial Informatics, 2013, 9(1): 89-99
- [22] Rubinov A M. Abstract convexity and global optimization[M]. New York: Springer, 2000
- [23] 张贵军, 何洋军, 郭海峰, 等. 基于广义凸下界估计的多模态差分进化算法[J]. 软件学报, 2013, 24(6): 1177-1195
- [24] Beliakov G. Geometry and combinatorics of the cutting angle method[J]. Optimization, 2003, 52(4/5): 379-394
- [25] Beliakov G. Cutting angle method-a tool for constrained global optimization[J]. Optimization Methods and Software, 2004, 19(2): 137-151
- [26] Beliakov G. A review of applications of the cutting angle methods[M]. New York: Springer, 2005: 209-248
- [27] Corder G W, Foreman D I. Nonparametric statistics for non-statisticians: a step-by-step approach[M]. Hoboken: John Wiley & Sons, 2009

(上接第 56 页)

- [2] Srinivas M, Patnaik L M. Genetic algorithms: A survey [J]. Computer, 1994, 27(6): 17-26
- [3] Fogel L J, Owens A J, Walsh M J. Artificial Intelligence through Simulated Evolution[M]. New York: Wiley, 1966
- [4] Rechenberg I. Evolutions Strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution[M]. Stuttgart: Fromman-Holzboog, 1973
- [5] Kennedy J. Particle swarm optimization[M] // Encyclopedia of Machine Learning. New York: Springer, 2010: 760-766
- [6] Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art[J]. IEEE Trans on Evolutionary Computation, 2011, 15(1): 4-31
- [7] 宋晓宇, 朱加园, 孙焕良. 一种求解带时间窗车辆路径问题的混合差分进化算法[J]. 计算机科学, 2014, 41(12): 220-225
- [8] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 398-417
- [9] Mallipeddi R, Suganthan P N, Pan Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies[J]. Applied Soft Computing, 2011, 11(2): 1679-1696