文章编号：2095-2163(2021)11-0189-06 中图分类号：TP391 文献标志码：A

精英反向黄金正弦被囊群优化算法

史鸿锋，李永林
（上海工程技术大学 管理学院，上海 201620）

摘 要：针对被囊群算法(Tunicate Swarm Algorithm,TSA)存在的收敛精度低,寻优性能不足等问题,结合精英反向学习策略与黄金正弦算法,提出了精英反向黄金正弦被囊群算法(Elite Opposition-based Golden-Sine Tunicate Swarm Algorithm,EGolden-STSA)。该算法通过提高种群多样性及其质量,提升了算法收敛速度与寻优精度。通过对 10 个基本测试函数进行寻优实验,且与单一策略改进算法进行对比,结果显示出精英反向黄金正弦被囊群优化算法具有更好的寻优能力,验证了优化方法的有效性。将改进的算法进一步用于求解高维问题,实验结果同样显示了其具有良好的寻优性能,算法改进效果明显。

关键词：群智能算法；被囊群算法；精英反向学习；黄金正弦算法

Elite Opposition-based Golden-Sine Tunicate Swarm Algorithm

SHI Hongfeng, LI Yonglin
(College of Management, Shanghai University of Engineering Science, Shanghai 201620, China)

【Abstract】 In order to solve the problems of low convergence accuracy and insufficient optimization performance of the Tunicate Swarm Algorithm, Elite opposition-based Golden-Sine Tunicate Swarm Algorithm is proposed. By improving the diversity and quality of the population, the algorithm improves the convergence speed and optimization accuracy of the algorithm. Through the optimization experiments of 10 basic test functions, and compared with the single strategy improved algorithm, the results show that the Elite opposition-based Golden-Sine Tunicate Swarm Algorithm has better optimization ability, which verifies the effectiveness of the optimization method. The improved algorithm is further applied to solve high-dimensional problems, and the experimental results also show that it has good optimization performance, and the improvement effect of the algorithm is obvious.

【Key words】 Swarm intelligence algorithm; Tunicate swarm algorithm; Elite opposition-based learning; Golden-sine algorithm

0 引 言

现实生活中,许多复杂问题都可以转化为优化问题从而进行求解。为了使决策变量的函数最小化或最大化,优化方法起着重要的作用。研究者们从自然生物群体的进化过程中得到启示,提出了群智能优化算法,其是基于概率随机搜索的进化算法。通过模拟昆虫、鸟类和鱼群等群体觅食、航行和搜索等行为抽象出来的一种算法。经典的群智能算法有粒子群算法[1]和蚁群算法[2]。但是,每个问题都有其自身的复杂性和性质,特定的优化算法并不能解决所有问题。因此,近年来出现许多新的算法。如:旗鱼算法(Sailfish Optimizer, SFO)[3]、蜉蝣算法(Mayfly Algorithm, MA)[4]等等。研究表明,不同的群智能优化算法对于解决不同领域优化问题各有所长。

被囊群算法(Tunicate Swarm Algorithm,TSA)[5]是 2020 年由 Gaurav 等人提出的一种新颖的群智能优化算法。该算法主要是模拟被囊生物喷射推进和群体行为,实现对目标问题的求解。基本的被囊群算法的优势在于操作简单,参数的调整少。但其缺点是收敛性较差,易陷入局部最优。针对这些问题,本文通过精英反向学习策略和黄金分割提升了算法的全局搜索能力,增强了求解精度。

1 基本被囊群算法

被囊生物具有在海洋中寻找食物来源的能力,其用来寻找食物来源的两种行为是喷气推进和群体智能。若对喷气推进和群体行为进行数学建模,目标应满足以下条件:

(1)避免搜索冲突:为了避免冲突,使用向量 $\boldsymbol{A}$ 计算新的搜索个体位置。

基金项目：上海市哲学社会科学规划项目(2017EGL0009);教育部人文社会科学项目(19YJA790028)。

作者简介：史鸿锋 (1995-),男,硕士研究生, 主要研究方向:智能计算、运筹与优化; 李永林 (1983-),男,博士,副教授,硕士生导师, 主要研究方向:运营管理、智能制造、运筹优化。

通讯作者：李永林 Email:Liyonglin7053@ sues.edu.cn

收稿日期：2021-05-24

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \tag{1}$$

$$\vec{G} = c_2 + c_3 - \vec{F} \tag{2}$$

$$\vec{F} = 2 \cdot c_1 \tag{3}$$

其中，G 代表重力；F 代表的是深海水流平流；变量 c_1、c_2、c_3 是[0,1] 范围内的随机数；M 代表个体间的互相作用力。M 向量的计算方法如下：

$$\vec{M} = [P_{\min} + c_1 \cdot P_{\max} - P_{\min}] \tag{4}$$

其中，$P_{\min}$ 和 $P_{\max}$ 代表最初彼此作用速度范围，一般取值为[1,4] 。

(2)向最优邻居移动：个体在满足上述条件之后，开始向相邻最优位置移动。

$$\overrightarrow{PD} = |\overrightarrow{FS} - rand \cdot \overrightarrow{P_p(x)}| \tag{5}$$

其中，PD 代表食物与搜索个体之间的距离；x 代表当前迭代次数；FS 代表食物的位置；$P_p(x)$ 代表个体的位置；$rand$ 代表[0,1] 之间的随机数。

(3)向最优位置收敛：最终每个个体将向最优位置靠近。

$$\overrightarrow{P_p(x)} = \begin{cases} \overrightarrow{FS} + \vec{A} \cdot \overrightarrow{PD}, & \text{if } rand \geqslant 0.5 \\ \overrightarrow{FS} - \vec{A} \cdot \overrightarrow{PD}, & \text{if } rand < 0.5 \end{cases} \tag{6}$$

其中，$P_p(x)$ 代表更新后的位置。

(4)种群行为：该算法保存前两个最优解，并根据最优搜索个体的位置更新其它搜索个体的位置，从而在数学上模拟被囊生物的群体行为。公式(7)用来定义群体行为：

$$P_p(\vec{x} + 1) = \frac{\overrightarrow{P_P(x)} + P_p(\vec{x} + 1)}{2 + c_1} \tag{7}$$

在寻优性能方面，虽然被囊群算法较之前的群智能算法有所提升，但是在迭代过程中依然存在寻优精度不足，收敛较慢等问题。

2 精英反向学习的被囊群算法

2.1 精英反向学习策略

在初始阶段，任何一种群智能算法中，第一步都是生成一个随机种群来表示问题的解。然而，在没有关于搜索空间的先验信息的情况下，群算法依赖于找到的最优位置来更新其它个体的位置。这种更新策略存在一定的局限性。比如，若当前的最优解不是全局最优解，这将导致群算法不收敛于全局解。而原因在于，全局最优解可能与当前最优解的方向相反。因此，可以计算出相反的方向。

针对这个问题，Tizhoosh 等学者于 2006 年提出反向学习策略(Opposition-Based Learning，OBL)[6]，证明了同时考虑随机解及其反向解的全局搜索要优于考虑单一随机解的搜索。

精英反向学习(Elite Opposition-Based Learning，EOBL)[7]是针对反向解不一定更易搜索到全局最优解提出的。因为精英个体所包含的有效信息多于普通个体，所以为了增加种群的多样性，可以利用当前种群中的精英个体来构造反向种群。假设：d 代表 d 维搜索空间，而 U_i 和 L_i 分别表示解 X_i 的上界和下界，K 是区间[0,1] 上的系数，则：

定义 1(精英反向解) 假设当前种群里一般个体对应的极值点为精英个体 $X^e_{i,j} = (X^e_{i,1}, X^e_{i,2}, \cdots, X^e_{i,d})(i = 1,2,\cdots,n; j = 1,2,\cdots,d)$，其反向解为 $\overline{X}^e_{i,j} = (\overline{X}^e_{i,1}, \overline{X}^e_{i,2}, \cdots, \overline{X}^e_{i,d})(i = 1,2,\cdots,n; j = 1,2,\cdots,d)$，

可定义为：

$$\overline{X^e_{i,j}} = K * (\alpha_j, \beta_j) - X^e_{i,j} \tag{8}$$

其中，$X^e_{i,j} \in [\alpha_j, \beta_j]$，$\alpha_j = \min(X_{i,j})$，$\beta_j = \min(X_{i,j})$，$\alpha_j, \beta_j$ 为动态边界。若 $\overline{X}^e_{i,j}$ 越过边界成为非可行解，可根据如下公式来重置：

$$\overline{X}^e_{i,j} = rand(\alpha_j, \beta_j) \tag{9}$$

精英反向学习策略中，在特定迭代 t 中生成的相反解的数量计算如下：

$$N_{op} = round\left(N - t \times \frac{N - 1}{T}\right) \tag{10}$$

其中，t 表示当前迭代；N 为种群大小；T 为算法的最大迭代次数。

由表达式可以看出，相反解的数量在整个迭代过程中减少，这有助于增强算法的开发能力。相反解的减少性质也有助于避免搜索代理在搜索过程中的高度多样性或扰动。

2.2 黄金正弦算法

黄金正弦算法[8]是利用数学中正弦函数设计，并通过引入黄金分割数来提升寻优性能的一种新型智能算法。算法大致可以分为 3 个步骤：

2.2.1 初始化

黄金正弦算法的初始步骤，是通过随机生成每个维度的均匀分布来更新初始空间。

$$V_i = rand * (ub - lb) + lb \tag{11}$$

其中，V_i 为第 i 个个体的初始值，ub 与 lb 为上下限值。

2.2.2　黄金分割系数

黄金正弦算法在位置更新过程中，引入黄金分割系数来缩小空间，从而引导个体不断趋近最优值。

$$\begin{aligned} x_1 &= a*(1-\tau)+b*\tau \\ x_2 &= a*\tau+b*(1-\tau) \end{aligned} \tag{12}$$

其中，x_1、x_2 是黄金分割系数，目的是平衡“搜索”与“开发”的过程；a 和 b 的初始默认值分别是 $-\pi$ 和 π；τ 为黄金分割率，$\tau=\frac{1-\sqrt{5}}{2}$。

2.2.3　位置更新

$$V_i^{t+1}=V_i^t\left|\sin(r_1)\right|-r_2\sin(r_1)\left|x_1D_i^t-x_2V_i^t\right| \tag{13}$$

其中，V_i^t 是第 i 个个体在 t 次的迭代位置；D_i^t 是第 i 个个体在 t 次的最优位置；r_1 是 $[0,2\pi]$ 内的随机数；r_2 是 $[0,\pi]$ 内的随机数。

2.3　精英反向黄金正弦被囊群优化算法 π

为解决被囊群算法存在的不足，受文献[9-10]的启发，本文提出了精英反向黄金正弦被囊群优化算法。此外，采用群体选择机制，按照适应值，将当前被囊群体与其反向群体进行排序，其中最优的 s 个个体作为下一代被囊生物个体，以此来提高种群的质量。种群的多样性，为算法良好的全局寻优性能打下坚实基础。并且对于每一代种群，该策略提供了不同于局部极值点的反向解来引导算法跳出局部最优，增强其寻优性能。同时，动态边界可获得一个逐步减小的搜索空间，更有利于提升算法的全局收敛速度。

本文在基本的被囊群算法基础上，引入黄金正弦算法，对被囊群生物的喷气推进与群体行为进行改进。基本的被囊群算法在确定食物位置时，引起群体靠近最优个体的过程中，使得群体大量聚集。虽然这样的方式有利于提升算法后期的收敛速度，但却在一定程度上使得种群的多样性大幅下降，容易导致陷入局部最优。针对该问题，本文引进黄金正弦算法对种群的喷射推进与群体行为进行了改进。EGolden-STSA 算法中，根据正弦函数所有值单位圆的扫描，类似于优化问题中搜索空间的搜索。结合基本 TSA 算法，使得被囊群每个个体可以在迭代过程中充分交流信息，了解位置差信息，再利用黄金分割搜索技术，逐步缩小搜索空间。通过参数 r_1 和 r_2 控制更细距离与方向，从而达到更好的寻优结果。EGolden-STSA 算法执行步骤如下：

步骤 1　初始化种群 Pp；

步骤 2　设置初始参数与最大迭代次数；

步骤 3　根据目标函数计算每个被囊生物个体的适应度值；

步骤 4　反向种群 $OP=\{\}$；

步骤 5　根据 $\alpha_j=\min(X_{i,j})$、$\beta_j=\min(X_{i,j})$ 计算个体的当前搜索边界；

步骤 6　由公式(8)对种群个体生成精英反向解并构成反向种群 OP；

步骤 7　从当前种群和反向种群中选择适应度值较好的 s 个个体作为下一代种群，并将适应度值最小的个体记为 FS；

步骤 8　更新参数 $\boldsymbol{A}$；

步骤 9　当 $|\boldsymbol{A}|\geqslant 1$ 时，根据公式(7)更新位置；

步骤 10　当 $|\boldsymbol{A}|<1$ 时，则根据式(14)更新位置：

$$\overrightarrow{Pp(x+1)}=\begin{cases}\overrightarrow{FS}-\vec{A}*\overrightarrow{PD} & r<0.5\\ \overrightarrow{Pp(x)}*\left|\sin(r_1)\right|+r_2\sin(r_1)* & \\ \left|x_1*\overrightarrow{FS}-x_2*\overrightarrow{Pp(x)}\right| & r\geqslant 0.5\end{cases} \tag{14}$$

步骤 11　检查可行性，若满足条件，则更新个体位置，否则不变；

步骤 12　重复步骤 3~11，直到达到最大迭代次数时停止。

对于算法的复杂度分析，总体初始化为 $o(n*d)$。

其中，n 为总体规模；d 为测试问题的维度；精英反向学习 $o(n*d)$；适应度计算 $o(\mathrm{Max}_{intrration}*n*d)$；被囊生物喷射推进与群体行为为 $o(N)$。所以整体时间复杂度为 $o(\mathrm{Max}_{intrration}*n*d)$，未增加计算负担。

3　仿真测试

3.1　参数设置

算法实验环境为 Windows10、64bit 系统、8G 内存，采用 MATLAB2018b 进行仿真实验。本文选取了基本的被囊群算法(TSA)、精英反向被囊群算法(ETSA)、黄金正弦被囊算法(Golden-STSA)和精英反向黄金正弦被囊群算法(EGolden-STSA)进行对比。种群规模为 30，迭代次数 500 次，共有参数相同。

3.2　测试函数

本文选取了 10 个基本测试函数，其特征性质不同。其中包括单模态与多模态函数，具有一定的代表性，可更全面的检测算法的性能，评估改进策略的

有效性。具体信息见表 1。

表 1 测试函数及其具体信息

Tab. 1 Test functions and information

	函数名称	维数	范围	最优值
f_1	Sphere Model	30	[-100,100]	0
f_2	Schwefel' sproblem 2.22	30	[-10,10]	0
f_3	Schwefel' sproblem 1.2	30	[-100,100]	0
f_4	Schwefel' sproblem 2.21	30	[-100,100]	0
f_5	Generalized Rosenbrock' function	30	[-30,30]	0
f_6	Step function	30	[-100,100]	0
f_7	Quartic function	30	[-1.28,1.28]	0
f_8	Powell function	30	[-4,5]	0
f_9	Sum Squares function	30	[-10,10]	0
f_{10}	Generalized Penalized function	30	[-50,50]	0

3.3 实验结果与分析

为了验证对 ETSA 算法改进的有效性与合理性，在每个测试函数上独立运行 30 次。表 2 列出了经过 30 次独立运行后，TSA、Golden-STSA、ETSA 和 EGolden-STSA 的实验结果(最优值、平均值、标准差)。

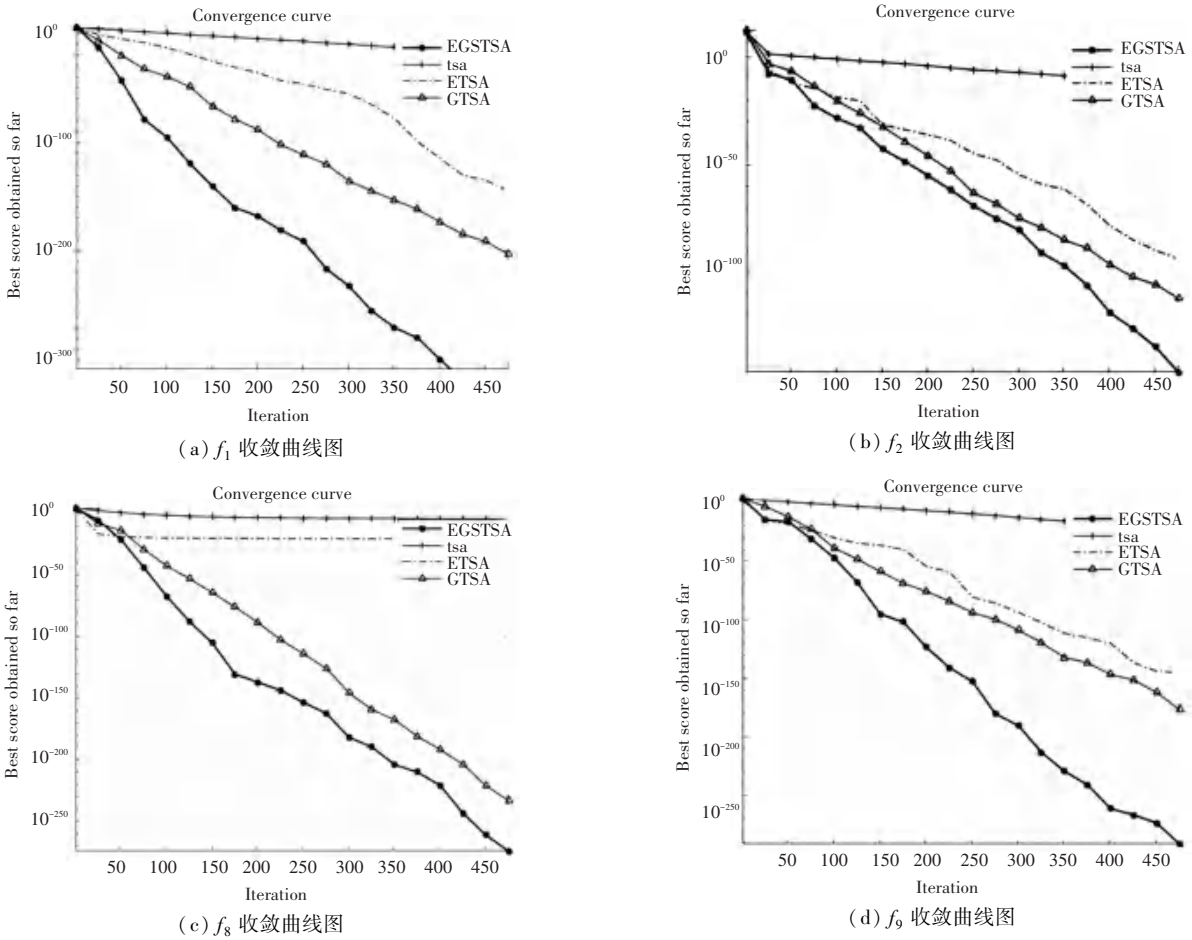
由表 2 可知，在所选测试函数中，EGolden-STSA 算法的寻优能力明显优于基本算法与单策略算法。函数 f_1、f_8、f_9 可以直接寻得最优值 0，且寻优效果显著。对于其它测试函数，EGolden-STSA 算法在效果上也表现出明显优势。如，对于函数 f_5 这个较难找到最优值的单模态函数，对比改进前的算法，其寻优精度提升了 10 个数量级。双策略改进后的算法其寻优效果进一步提升，适用范围更加广泛。对于函数 f_{10}，是一个多模态函数，主要用来衡量算法的探索能力，EGolden-STSA 算法与 ETSA 算法寻优效果均表现良好，在寻找最优值方面，EGolden-STSA 算法略占优势，且 EGolden-STSA 算法标准差优于其它 3 种算法，说明算法改进，增强了原算法的稳定性，鲁棒性较好。以上分析表明 EGolden-STSA 的整体寻优性能优于 TSA、ETSA 和 Golden-STSA。

图 1 选取了部分收敛迭代曲线图展示。由总体函数收敛曲线可知，改进算法在收敛精度上有明显的提升，克服了 TSA 算法寻优精度不高的缺点。而函数 f_1、f_2、f_8、f_9，则直观展示了改进后的 EGolden-STSA 在收敛速度上的提升，更快的收敛到最优值，远远优于相比较的其它算法。综合来看，改进的 EGolden-STSA算法在各方面上都有了较好的提升。

表 2 测试函数实验结果

Tab. 2 Experimental results of test function

函数	算法	最优值	平均值	标准差
f_1	TSA	3.42E-24	1.82E-21	4.29E-21
	Golden-STSA	9.16E-256	3.39E-182	0
	ETSA	8.87E-223	1.27E-107	6.97E-107
	EGolden-STSA	0	3.32E-272	0
f_2	TSA	1.15E-14	1.36E-13	2.30E-13
	Golden-STSA	9.79E-126	5.85E-97	3.14E-96
	ETSA	1.09E-108	3.28E-34	1.80E-33
	EGolden-STSA	2.61E-192	1.51E-119	8.31E-119
f_3	TSA	4.94E-09	1.46E-3	5.20E-3
	Golden-STSA	2.16E-155	2.41E-101	1.32E-100
	ETSA	1.97E-44	7.35E-06	3.93E-05
	EGolden-STSA	5.55E-287	2.21E-166	0
f_4	TSA	1.59E-2	3.86E-1	4.95E-1
	Golden-STSA	5.03E-124	3.46E-88	1.89E-87
	ETSA	6.41E-104	2.55E-67	1.38E-66
	EGolden-STSA	3.12E-186	7.02E-122	3.84E-121
f_5	TSA	2.62E+1	2.84E+1	9.29E-1
	Golden-STSA	2.58E+1	2.71E+1	9.96E-1
	ETSA	1.21E-07	2.46E-4	4.38E-4
	EGolden-STSA	3.43E-09	1.29E-4	1.88E-4
f_6	TSA	5.99E-09	1.21	1.88
	Golden-STSA	5.99E-09	5.96	1.20
	ETSA	2.18E-09	1.41E-4	2.98E-4
	EGolden-STSA	1.06E-11	1.41E-4	2.98E-4
f_7	TSA	4.25E-3	9.38E-3	3.56E-3
	Golden-STSA	1.32E-05	3.26E-4	3.74E-4
	ETSA	3.14E-05	5.07E-4	5.15E-4
	EGolden-STSA	6.11E-06	1.81E-4	2.20E-4
f_8	TSA	4.49E-06	5.24E-4	7.35E-4
	Golden-STSA	6.47E-251	2.64E-65	1.44E-64
	ETSA	1.19E-74	2.19E-05	9.09E-05
	EGolden-STSA	0	4.55E-192	0
f_9	TSA	3.41E-25	2.32E-22	3.87E-22
	Golden-STSA	8.21E-248	2.35E-167	0
	ETSA	2.88E-203	4.87E-114	2.67E-113
	EGolden-STSA	0	1.66E-265	0
f_{10}	TSA	3.9E-1	8.84	4.70
	Golden-STSA	8.24E-3	1.02E-1	7.68E-2
	ETSA	1.77E-11	4.82E-07	9.65E-07
	EGolden-STSA	5.34E-12	2.01E-07	2.86E-07



(a) f_1 收敛曲线图　(b) f_2 收敛曲线图

(c) f_8 收敛曲线图　(d) f_9 收敛曲线图

图 1　部分函数收敛曲线图

Fig. 1　Convergence curve of some functions

3.4　求解高维函数的实验分析

根据改进算法在低维实验结果对比分析可知，EGolden-STSA 在低维上表现良好，但是鉴于目前工程问题较为复杂，许多改进算法在面对高维测试函数极易失效，所以为了测试本文 EGolden-STSA 算法的有效性，进行了高维函数优化实验。高维参数设置为 500 维与1 000维。具体结果见表 3。

表 3　求解高维度函数的实验结果

Tab. 3　Experimental results of solving high-dimensional functions

	EGolden-STSA					
	维度 d = 500			维度 d = 1 000		
	最优值	平均精度	标准差	最优值	平均精度	标准差
f_1	0	7.08E-265	0	0	3.33E-249	0
f_2	5.40E-191	5.43E-106	2.97E-105	3.88E-176	8.73E-100	4.78E-99
f_3	1.84E-221	6.92E-76	3.02E-75	1.58E-184	1.70E-42	9.34E-42
f_4	2.21E-191	6.95E-120	3.29E-119	2.48E-178	7.86E-129	4.30E-128
f_5	1.99E-06	16.47	90.18	1.94E-07	98.90	301.74
f_6	1.27E-08	3.21E-05	7.12E-05	7.10E-07	7.53E-05	1.21E-4
f_7	3.53E-06	2.08E-4	2.79E-4	5.54E-06	1.56E-4	1.69E-4
f_8	0	2.38E-221	0	0	1.31E-264	0
f_9	0	9.03E-241	0	0	7.37E-208	0
f_{10}	1.20E-11	6.61E-08	1.70E-07	3.05E-11	1.57E-07	4.31E-07

（下转封三）