





















所以,系统可靠性可以表示为

$$RE = \sum_{i=0}^{\mu} RE(i) = \frac{r(S_1) \cdot p_{1,2} \cdot r(S_2) \cdot p_{2,3} \cdot r(S_3)}{1 - p_{2,2} \cdot r(S_2)} \quad (29)$$

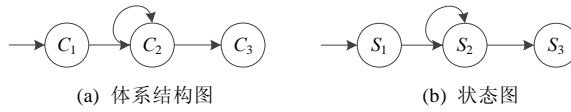


Fig.7 Cycle style  
图 7 循环执行风格

### 3 仿真实验结果与分析

#### 3.1 参数设置

参数设置对随机性算法的性能和效率有显著的影响<sup>[22]</sup>,因此,本节通过设置参数对比分析实验,找出求解软件可靠性分配问题的较优参数.本文提出的 PHEDA-SCDE 算法包含了几个关键参数:交叉算子  $CR_0$  和  $CR_1$ 、初始变异算子  $F_0$ 、阈值  $truncation$ 、决策因子上限  $n_{max}$  和下限  $n_{min}$ .本文采用实验设计法(design of experiments, 简称 DOE)<sup>[23]</sup>,每个参数选择合理取值,将其构成算法的参数集.参数取值如下: $CR_0 \in \{0.1, 0.2, 0.3, 0.4\}$ ,  $CR_1 \in \{0.6, 0.7, 0.8, 0.9\}$ ,  $F_0 \in \{0.2, 0.4, 0.6, 0.8\}$ ,  $truncation \in \{0.2, 0.3, 0.4\}$ ,  $n_{max} \in \{0.0, 1, 0.2\}$ ,  $n_{min} \in \{0.8, 0.9, 1\}$ .这些参数对于本文提出的算法,共有  $4 \times 4 \times 4 \times 3 \times 3 \times 3 = 1728$  种不同配置.为了保证实验结果更可靠,每一种配置都独立运行 5 次.

本节按照文献[12,24]的方法生成受控数据集进行参数设置实验,然后在测试实例集上进行验证.受控数据集详细信息见表 1.

Table 1 The control data set

表 1 受控数据集详细信息

状态图	转移概率	目标函数
	$p_{1,2}=0.6$	$cost+k \times \min\{0, RE-r_0\}$
	$p_{1,3}=0.1$	
	$p_{1,4}=0.3$	
	$p_{2,3}=0.3$	
	$p_{2,5}=0.7$	
	$p_{3,6}=1.0$	
	$p_{4,6}=1.0$	
	$p_{5,5}=0.4$	
	$p_{5,7}=0.6$	
	$p_{6,7}=1.0$	

为探寻参数对算法的影响,本文采用方差分析(analysis of variance,简称 ANOVA)方法来设计实验.主效应图(如图 8 所示)是显示单个参数对结果影响的显示图,根据图例可以分析得到达到哪个值的时候这个参数的影响力最大,从而找出最主要因素.从图 8 中可以明显看出:初始变异算子  $F_0$  对目标函数的影响最大,这主要是因为子算法 SCDE 中有 60% 进行自适应交叉,当其值为 0.6 时,目标函数达到最优化;交叉算子  $CR_0$  与决策因子下限  $n_{min}$  对目标函数影响相对最小.根据图 8 分析可知,选择表 2 中所列出的参数配置较为合理.

Table 2 Parameter tuning of PHEDA-SCDE

表 2 PHEDA-SCDE 算法运行参数设置

参数类型	参数名称	值
$CR_0$	交叉算子	0.2
$CR_1$	交叉算子	0.8
$F_0$	初始变异算子	0.6
$truncation$	阈值	0.3
$n_{max}$	决策因子上限	0.2
$n_{min}$	决策因子下限	1.0

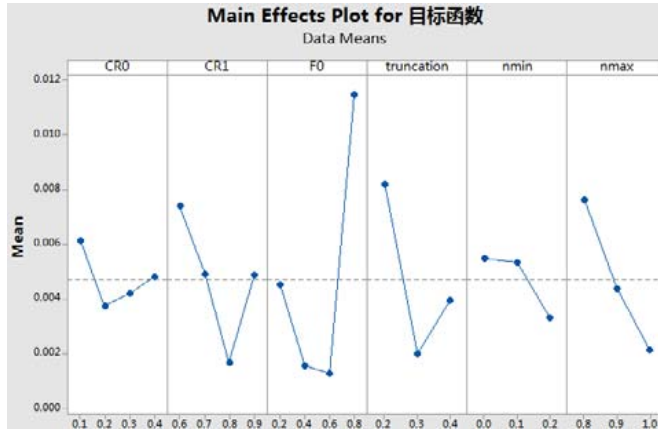


Fig.8 Main effect plot of parameters  
图 8 PHEDA-SCDE 算法主效应图

### 3.2 仿真算例实验与分析

#### 3.2.1 软件可靠性分配算例

为不失算法的通用性,本文采用 3 个仿真算例,分别为单输入单输出系统、单输入多输出系统和多输入多输出系统.以某复杂软件系统可靠性优化仿真算例进行说明<sup>[8]</sup>,如图 9.1 所示.算例 1 是由 12 个构件组成的单输入单输出系统,其中, $C_1$ 代表输入构件, $C_{12}$ 代表输出构件.特别地,构件  $C_7$ 和  $C_8$ 是并发关系, $C_{10}$ 是  $C_{12}$ 的容错构件,构件  $C_4$ 自身循环.系统状态如图 9.1(b)所示,是由图 9.1(a)的体系结构图转变过来.本文按照文献[24]的方法,算例 2(见图 9.2)删除算例 1 中  $C_{11}$ 和  $C_{12}$ 之间的连接、 $C_{11}$ 和  $C_6$ 之间的连接,使得  $C_{11}$ 和  $C_{12}$ 均为输出构件.算例 3(见图 9.3)中系统开始于输入组件  $C_1$ 和  $C_2$ ,结束于输出组件  $C_{11}$ 和  $C_{12}$ .

为了简化模型,本文认为,转移概率  $p_{i,j}$ 已经考虑了连接件的可靠性.算例 1 的转移概率如下所示,算例 2 和算例 3 相应的转移概率与算例 1 相似.

$$\begin{matrix}
 P_{1,2} = 0.6 & P_{1,3} = 0.1 & P_{1,4} = 0.3 & P_{2,3} = 0.3 & P_{2,5} = 0.7 & P_{3,6} = 1 & P_{4,4} = 0.2 \\
 P_{4,p} = 0.8 & P_{5,b} = 1 & P_{6,b} = 1 & P_{p,11} = 1 & P_{b,12} = 1 & P_{11,6} = 0.1 & P_{11,12} = 0.9
 \end{matrix}$$

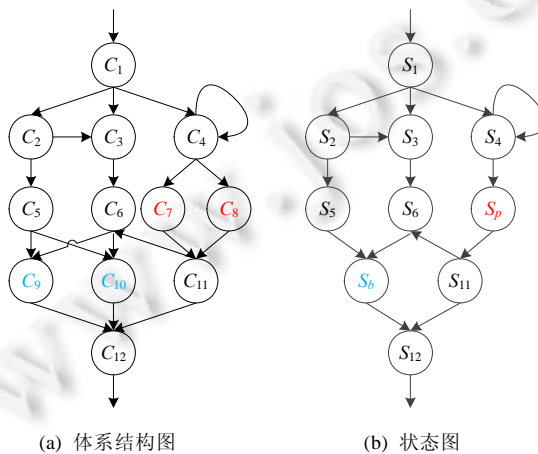


Fig.9.1 A single-input/single-output type (example 1)  
图 9.1 单输入单输出系统体系结构图和状态图(算例 1)

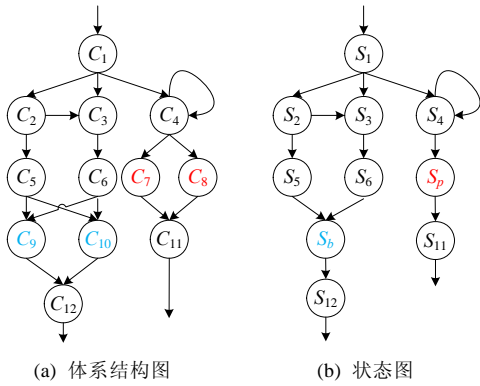


Fig.9.2 A single-input/multiple-output type (example 2)

图 9.2 单输入多输出系统体系结构图和状态图(算例 2)

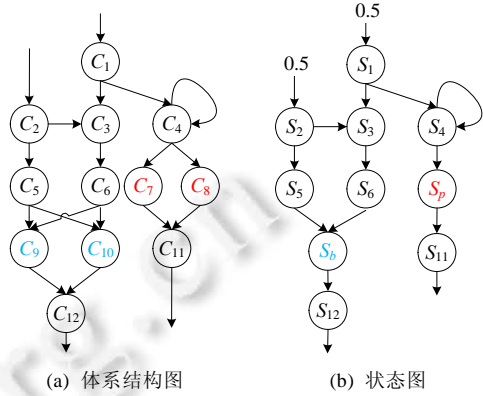


Fig.9.3 A multiple-input/multiple-output type (example 3)

图 9.3 多输入多输出系统体系结构图和状态图(算例 3)

如下对算例 1 进行详细分析,首先依据公式(3)将可靠性优化模型转变为无约束求目标函数最小化:

$$\text{Min } cost + penalty = a_i \cdot \sum_{i=1}^n e^{\frac{(1-c_i) \cdot r_i - r_{i,\min}}{r_{i,\max} - r_i}} + k \cdot \min\{0, RE - r_0\} \quad (30)$$

其中, $a=[0.88 \ 0.9 \ 0.86 \ 0.76 \ 0.89 \ 0.93 \ 0.94 \ 0.75 \ 0.95], c=[0.23 \ 0.28 \ 0.37 \ 0.45 \ 0.05 \ 0.25 \ 0.5 \ 0.2 \ 0.41], r_{i,\min}=0.5, r_{i,\max}=1, r_0=0.9$ .

根据第 3 节中不同体系风格可靠性的分析,算例 1 软件系统可靠性表示为公式(31):

$$RE = 0.42 \cdot r(c_1) \cdot r(c_2) \cdot r(c_3) \cdot r(c_6) \cdot r(c_{12}) + 0.18 \cdot r(c_1) \cdot r(c_2) \cdot r(c_3) \cdot r(c_6) \cdot r(c_b) \cdot r(c_{12}) + 0.1 \cdot r(c_1) \cdot r(c_3) \cdot r(c_6) \cdot r(c_b) \cdot r(c_{12}) + \frac{0.216 \cdot r(c_1) \cdot r(c_4) \cdot r(c_p) \cdot r(c_{11}) \cdot r(c_{12})}{1 - 0.2 \cdot r(c_4)} + \frac{0.24 \cdot r(c_1) \cdot r(c_4) \cdot r(c_p) \cdot r(c_{11}) \cdot r(c_6) \cdot r(c_b) \cdot r(c_{12})}{1 - 0.2 \cdot r(c_4)} \quad (31)$$

其中,各个构件可靠性  $r$  为可靠性分配的自变量,系统决策者根据最终求得的构件可靠性值分配资源,从而达到复杂软件整体可靠性最大化和代价最小化.

### 3.2.2 实验结果和分析

本文主要在收敛速度以及精度方面对 PHEDA-SCDE,DE,SCDE,EDA 以及 DHOACP 算法进行比较,每次目标函数独立运行 20 次,实验经过 200 次迭代,分别得到各种算法最优解的平均值、标准差、最优值、最差值(见表 3)作为目标函数精度的评价结果.同时,根据一次独立运行的 CPU 时间作为算法效率的评价结果,运行时间是一个关键的理论问题<sup>[25]</sup>.从表 3 中可以看出:对于单输入单输出软件系统分配问题(算例 1),PHEDA-SCDE 在求解质量上明显优于其余算法,在求解效率上明显优于 DHOACP,对于单输入多输出系统(算例 2)和多输入多输出系统(算例 3)的软件可靠性分配问题,该算法无论在求解效率还是在求解精度上都具有良好的效果;其次,本文对传统的差分进化算法进行自适应交叉改进,改进结果表明,SCDE 算法在求解质量上明显优于 DE 算法,进一步验证了 SCDE 算法相较于 DE 算法全局搜索能力强,能迅速得到解空间.

**Table 3** The solutions of objective functions**表 3** 目标函数实验结果

测试算例	算法	平均值	标准差	最优值	最差值	时间(s)
算例 1	PHEDA-SCDE	<b>515.652 9</b>	0.116 1	515.438 2	515.945 8	0.775 6
	DE	525.787 5	2.480 7	521.684 5	530.882 0	0.248 6
	SCDE	516.325 1	0.340 1	515.706 0	517.068 2	0.221 5
	EDA	1 320.637	1.85e+03	558.818 9	7 685.552	0.163 7
	DHOACP	533.042 5	5.585 4	524.793 9	542.719 2	19.602
算例 2	PHEDA-SCDE	<b>1.3692e+04</b>	4.95e-06	1.3692e+04	1.3692e+04	0.868 4
	DE	1.3694e+04	0.209 3	1.3693e+04	1.3694e+04	0.234 8
	SCDE	1.3693e+04	3.85e-04	1.3693e+04	1.3693e+04	0.224 0
	EDA	1.6623 e+04	3.10 e+03	1.3914 e+04	2.4159 e+04	0.305 2
	DHOACP	1.3694e+04	0.302 1	1.3693e+04	1.3694e+04	18.525
算例 3	PHEDA-SCDE	<b>1.1315e+04</b>	6.20e-06	1.1315e+04	1.1315e+04	0.717 0
	DE	1.1316e+04	0.293 1	1.1315e+04	1.1316e+04	0.187 9
	SCDE	<b>1.1315e+04</b>	1.76e-04	1.1315e+04	1.1315e+04	0.186 3
	EDA	1.3312 e+04	2.39e+03	1.1713 e+04	2.2824 e+04	0.176 0
	DHOACP	1.1316e+04	0.289 3	1.1315e+04	1.1316e+04	22.347

收敛曲线图主要描绘随着进化次数的增加目标函数适应度值的变化趋势,算法收敛曲线图如图 10 所示.图 10(a)为算例 1 的算法收敛曲线图,图 10(b)为算例 2 的算法收敛曲线图,图 10(c)为算例 3 的算法收敛曲线图.为了保证系统可靠性,软件代价不会是一个无穷小的值,一定收敛于某个值.为了直观地描述软件代价的收敛曲线,算法收敛曲线图纵坐标表示适应度值的对数变换,值为  $\log_{10}(f(x))$ .横坐标表示进化代数.从 3 幅图中可以看出,DHOACP 算法虽然在其初期收敛速度最快,但很快陷入局部最优.在算法收敛性上,PHEDA-SCDE 明显优于其他 3 种算法,说明算法在初期快速搜索解空间,有很强的全局搜索能力.因此,PHEDA-SCDE 算法结合分布估计算法和自适应交叉差分进化算法的优点,无论在求解精度还是收敛速度方面,对于复杂软件系统可靠性的分配均具有优秀的效果.算法详细参数设置见表 4.

**Table 4** Parameter tuning of all compared algorithms**表 4** 算法运行参数设置

算法	参数类型	参数名称	值	
PHEDA-SCDE		$CR_0$	交叉算子	0.2
		$CR_1$	交叉算子	0.8
		$F_0$	初始变异算子	0.6
		<i>truncation</i>	阈值	0.3
		$n_{max}$	决策因子上限	0.2
		$n_{min}$	决策因子下限	1.0
	DE		$CR$	交叉算子
		$F_0$	初始变异算子	0.6
SCDE		$CR_0$	交叉算子	0.2
		$CR_1$	交叉算子	0.8
		$F_0$	初始变异算子	0.6
EDA	<i>truncation</i>	阈值	0.3	
DHOACP		$w$	惯性权重	0.8
		$C_1$	加速常数	1.5
		$C_2$	加速常数	1.5
		$V_{max}$	粒子最大速度	1
		$V_{min}$	粒子最小速度	0

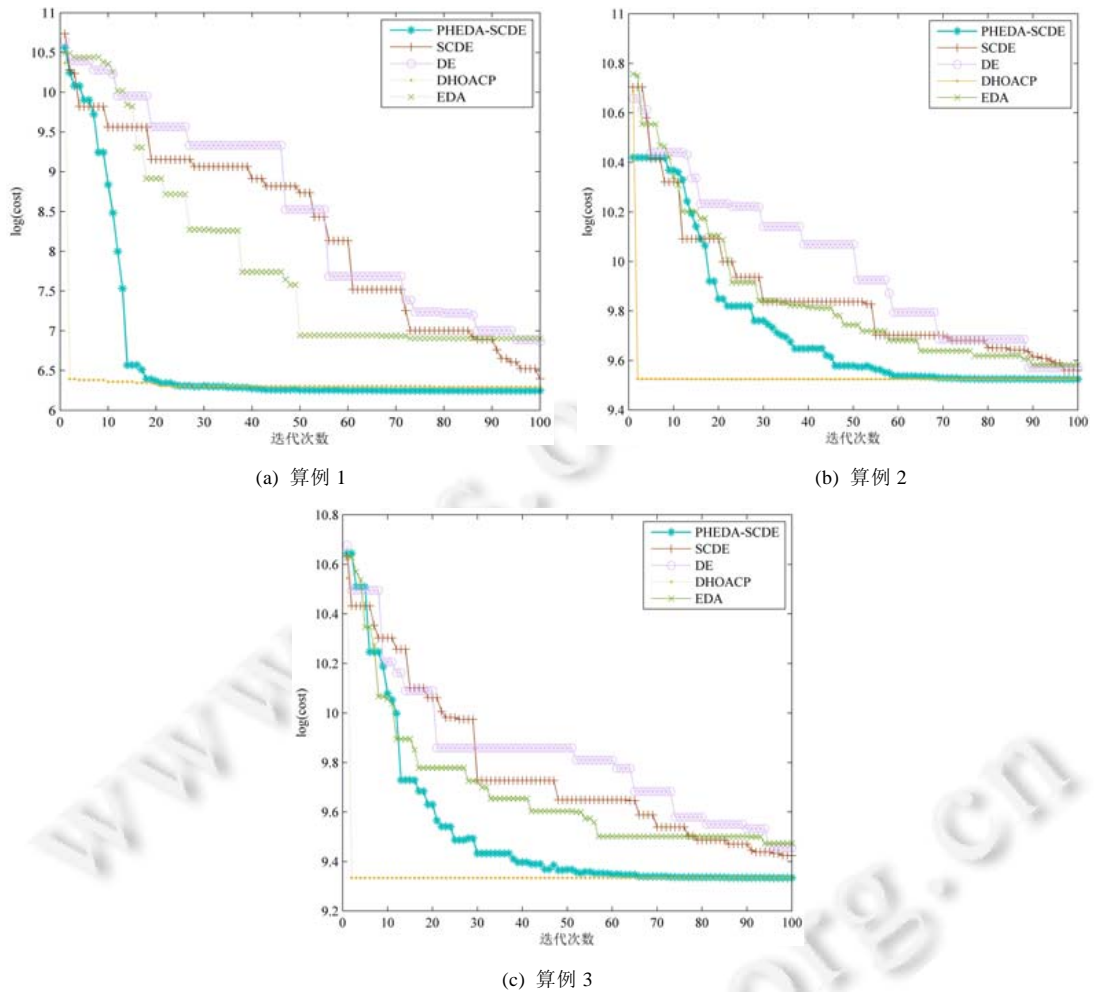


Fig.10 The convergence curve of all compared algorithms

图 10 算法收敛曲线图

## 4 总 结

可靠性问题一直是软件可靠性领域的重中之重,合理分配构件可靠性有利于增强系统可靠性、提高资源利用率、减少开发维护成本.对于大型复杂软件系统,本文提出一种新型的元启发式算法——基于罚函数的混合分布估计和自适应交叉差分进化的优化算法(PHEDA-SCDE),该方法结合分布估计算法和差分进化算法的优点.在算法初期,PHEDA-SCDE 具有全局搜索能力强、收敛速度快;在算法中后期,该算法具有较强的局部搜索能力,避免对复杂优化问题陷入局部最优.实验结果证明,算法以更少的成本、较短的计算时间优于其他现有算法.

未来的研究方向主要是云计算环境下的软件可靠性评估和分配问题,提高算法的精度和收敛速度,改进目标函数的准确性,研究构件可靠性对系统可靠性的影响.

## References:

- [1] Kuo W, Prasad VR. An annotated overview of system reliability optimization. *IEEE Trans. on Reliability*, 2000,49(2):176-187.
- [2] Ravi V, Reddy PJ, Zimmermann HJ. Fuzzy global optimization of complex system reliability. *IEEE Trans. on Fuzzy Systems*, 2000, 8(3):241-248.

- [3] Nakagawa Y, Nakashima K. A heuristic method for determining optimal reliability allocation. *IEEE Trans. on Reliability*, 2009, R-26(3):156–161.
- [4] Ramirez-Marque JE, Rocco CM. Evolutionary optimization technique for multi-state two-terminal reliability allocation in multi-objective problems. *IIE Transactions*, 2010,42(8):539–552.
- [5] Hsieh TJ, Yeh WC. Penalty guided bees search for redundancy allocation problems with a mix of components in series-parallel systems. *Computers & Operations Research*, 2012,39(11):2688–2704.
- [6] Khalili-Damghani K, Abtahi AR, Tavana M. A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems. *Reliability Engineering & System Safety*, 2013,111(2):58–75.
- [7] Tran DH, Cheng MY, Cao MT. Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality trade off problem. *Knowledge-Based Systems*, 2015,74(1):176–186.
- [8] Zhang J, Lei H. A pre-distribution algorithm of component reliability in Internetware system. *Computing*, 2015,97(7):755–768.
- [9] Zhang E, Chen Q. Multi-Objective reliability redundancy allocation in an interval environment using particle swarm optimization. *Reliability Engineering & System Safety*, 2016,145:83–92.
- [10] Goševa-Popstojanova K, Trivedi KS. Architecture-based approach to reliability assessment of software systems. *Performance Evaluation*, 2001,45(2-3):179–204.
- [11] Wang WL, Pan D, Chen MH. Architecture-Based software reliability modeling. *Journal of Systems & Software*, 2006,79(1):132–146.
- [12] Cheung RC. A user-oriented software reliability model. *IEEE Trans. on Software Engineering*, 1980,6(2):118–125.
- [13] Helander ME, Zhao M, Ohlsson N. Planning models for software reliability and cost. *IEEE Trans. on Software Engineering*, 1998, 24(6):420–434.
- [14] Li F, Wu T, Hu M, Dong J. An accurate penalty-based approach for reliability-based design optimization. *Research in Engineering Design*, 2010,21(2):87–98.
- [15] Mettas A. Reliability allocation and optimization for complex systems. In: *Proc. of the Annual Reliability and Maintainability Symp.* 2000,30(6):216–221.
- [16] Wang SY, Wang L, Xu Y. An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Int'l Journal of Production Research*, 2013,51(12):3778–3793.
- [17] Das S, Suganthan PN. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. on Evolutionary Computation*, 2011, 15(1):4–31.
- [18] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. on Evolutionary Computation*, 2009,13(2):398–417.
- [19] Wang S, Li Y, Yang H, Liu H. Self-Adaptive differential evolution algorithm with improved mutation strategy. *Applied Intelligence*, 2017,6:1–15.
- [20] Alguliev RM, Aliguliyev RM, Isazade NR. DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization. *Knowledge-Based Systems*, 2012,36(36):21–38.
- [21] Lipowski A, Lipowska D. Roulette-Wheel selection via stochastic acceptance. *Physica A Statistical Mechanics & Its Applications*, 2011,391(6):2193–2196.
- [22] Zhang B, Pan QK, Gao L, Zhang XL, Sang HY, Li JQ. An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming. *Applied Soft Computing*, 2016,52(C):14–27.
- [23] Kempthorne O, Montgomery DC. Design and analysis of experiments. *Journal of the American Statistical Association*, 2008, 101(474):853–854.
- [24] Lo JH, Huang CY, Chen IY, Kuo SY, Lyu MR. Reliability assessment and sensitivity analysis of software reliability growth modeling based on software module structure. *Journal of Systems & Software*, 2005,76(1):3–13.
- [25] Yu Y, Qian C. Running time analysis: Convergence-based analysis reduces to switch analysis. In: *Proc. of the IEEE Congress on Evolutionary Computation*. New York :IEEE, 2015. 2603–2610.

## 附录 A

通过一个简单的算例来介绍 PHEDA-SCDE 算法的进化操作,对其有一个直观的认识.假设用分布估计算法求解  $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$  的最大值,其中  $x_j \in [0, 1] (j=1, 2, 3)$ .

第 1 步,依据公式(5)初始化种群  $x, x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}, i=1, 2, \dots, NP$ .种群大小  $NP$  为 5,种群维度  $D$  为 3,通过适应度函数  $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$  计算各个个体适应度值,此时,最优个体为个体 1.初始种群及适应度值见表 5.

Table 5 The initial population and its fitness

表 5 初始种群及适应度值

个体	$x_1$	$x_2$	$x_3$	$f$
1	0.2	0.8	0.9	1.9
2	0.4	0.1	0.2	0.7
3	0.6	1	0	1.6
4	0.3	0.1	0.5	0.9
5	0.7	0.1	0.7	1.5

第 2 步,通过 EDA 算法对初始种群进行优化.对种群按照适应度值从高到低排序,假设阈值  $truncation=0.6$ ,选择较优的  $truncation \times NP=3$  个个体,分别为个体 1、个体 3 和个体 5.

a) 根据均值表达式(32)和方差表达式(33),得到较优种群的均值和方差:

$$u_j = \frac{\sum_{i=1}^{NP} x_{ij}}{NP} \quad (32)$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{NP} (x_{ij} - u_j)^2}{NP}} \quad (33)$$

b) 根据公式(34)构建概率模型,并从中采样,得到  $NP$  个新样本,构成新种群:

$$P(x) = \prod_{j=1}^D \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2}\left(\frac{x_j - u_j}{\sigma_j}\right)^2} \quad (34)$$

算法 EDA 生成的实验种群及适应度值见表 6.

Table 6 The EDA population and its fitness

表 6 EDA 实验种群及适应度值

个体	$u_1$	$u_2$	$u_3$	$f$
1	0.2	0.8	0.9	1.9
3	0.6	1	0	1.6
5	0.7	0.1	0.7	1.5
6	0.5	0.7	0.4	1.6
7	0.6	0.1	0.8	1.5

c) 假设此时为算法初期,决策因子  $n=4$ .每个个体被选中的概率与适应度值大小成正比.适应度值越大,被选中的概率越高.此时,采用轮盘赌方法选择 4 个适应度值较高的实验个体,组成 EDA 候选实验种群,其种群及适应度值见表 7.

Table 7 The EDA candidate population and its fitness

表 7 EDA 候选实验种群及适应度值

个体	$u_1$	$u_2$	$u_3$	$f$
1	0.2	0.8	0.9	1.9
3	0.6	1	0	1.6
5	0.7	0.1	0.7	1.5
7	0.6	0.1	0.8	1.5

第 3 步,通过 SCDE 算法对初始种群进行优化.

a) 假设初始变异算子  $F_0=0.1$ ,当前迭代次数为 1,依据公式(6)自适应变异生成变异个体  $v$ ,例如  $v_2=x_2+0.2 \times$



$(x_5-x_4)$ .变异种群及适应度值见表 8.

**Table 8** The SCDE mutation population and its fitness

**表 8** SCDE 变异种群及适应度值

个体	$v_1$	$v_2$	$v_3$	$f$
1	0	1	1	2
2	1	0.1	0.6	1.7
3	1	1	0.4	2.4
4	0.1	1	1	2.1
5	0.5	1	1	2.5

b) 假设交叉算子  $CR_0=0.4, CR_1=0.7$ ,依据公式(9)和图 1 进行自适应交叉,生成实验个体  $u$ .SCDE 实验种群及适应度值见表 9.

**Table 9** The SCDE population and its fitness

**表 9** SCDE 实验种群及适应度值

个体	$u_1$	$u_2$	$u_3$	$f$
1	0.2	0.8	0.9	1.9
2	1	0.1	0.6	1.7
3	0.2	0.8	0.9	1.9
4	0.5	0.7	0.4	1.6
5	0.5	1	1	2.5

c) 根据第 2 步中假设的决策因子  $n=4$ ,此时,轮盘赌选择 1 个适应度值较高的实验个体 5,组成 SCDE 候选实验种群.

第 4 步,合并 EDA 的候选实验个体 1,3,5,7 和 SCDE 的候选实验个体 5 为新种群,其种群及适应度值见表 10.

**Table 10** The new population and its fitness

**表 10** 新种群及适应度值

个体	$u_1$	$u_2$	$u_3$	$f$
1	0.2	0.8	0.9	1.9
2	0.6	1	0	1.6
3	0.7	0.1	0.7	1.5
4	0.6	0.1	0.8	1.5
5	0.5	1	1	2.5

第 5 步,依据公式(8)对当前种群和新种群进行选择,选择更优向量产生下一代.

第 6 步,依据公式(10)更新决策因子  $n$ .

第 7 步,若终止条件满足,则算法终止;否则,转向第 2 步继续执行.



徐悦(1994—),女,江苏苏州人,博士生,主要研究领域为软件可靠性,智能优化算法.



皮德常(1971—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据挖掘,系统优化.