

基于混沌扰动策略的果蝇优化算法

李宗岳¹, 陈志军¹, 李名远²

(1 新疆大学 电气工程学院, 新疆 乌鲁木齐 830047; 2 遵义市国家税务局, 贵州 遵义 563000)

摘 要: 果蝇优化算法是模拟果蝇觅食的一种解决优化问题的智能算法, 但是为了避免其在优化过程中容易陷入局部极值的缺陷, 提出一种新的基于混沌扰动的果蝇优化算法. 在迭代过程中, 首先利用基本果蝇算法获得最优个体, 然后对最优个体引入混沌扰动策略, 增强获得全局最优解的可能性. 同时通过动态变化系数 η 调整在迭代寻求过程中混沌扰动变量的值, 提高算法跳出局部最优和寻找全局最优的能力. 对 6 个经典测试函数的仿真结果表明, 新算法的收敛速度、收敛精度和鲁棒性方面比基本果蝇优化算法具有明显优势.

关键词: 果蝇优化算法; 混沌扰动; 最优个体; 动态变化系数

中图分类号: TP301.6; TP18

文献标识码: A

文章编号: 1000-7180(2016)07-0064-05

Fruit Fly Optimization Algorithm Based on Chaotic Interrupt

LI Zong-yue¹, CHEN Zhi-jun¹, LI Ming-yuan²

(1 School of Electrical Engineering, Xinjiang University, Urumqi 830047, China;

2 Zunyi Municipal Office, State Administration of Taxation, Zunyi 563000, China)

Abstract: This paper puts forward a new algorithm based on chaotic interrupt. Firstly the current optimal individual is gained by standard fruit fly optimization algorithm, then chaotic interrupt is added to the current optimal individual to increase the probability of getting the ultimate global optimum. Meanwhile, the value of chaotic variables in the iterative process is adjusted by dynamic variation coefficient η to enhance the ability to escape from the local optimum and seek the global optimum. Simulation results of six classical test functions show that the novel improved algorithm has obvious advantages on convergence precision, convergence speed and robustness.

Key words: fruit fly optimization algorithm; chaotic interrupt; current optimal individual; dynamic variation coefficient

1 引言

果蝇优化算法是由学者潘文超提出的一种寻求全局优化的智能算法^[1], 果蝇优化算法要调整参数少, 程序易于实现, 已在多个领域得到应用^[2-3], 但该算法的参数不易择取, 且易陷入局部最优, 同时由于该算法提出较晚, 目前, 在改善算法性能方面, 无论是理论分析还是应用研究都处于初始阶段, 有必要对果蝇算的改进做相应的研究.

果蝇优化算法每次种群个体更新时, 所有个体都以最优的果蝇个体为中心, 随机分布在其周围, 这一种群个体更新机制简单易于实现, 但容易导致种

群多样性的丢失, 特别是解决高维多极值复杂优化的问题时, 算法容易陷入局部极值, 从而导致收敛精度降低, 收敛速度变慢. 针对 FOA 这一问题, 文献[4]将细菌迁徙操作引入到果蝇优化算法中, 条件满足时, 依据种群个体的适应度赋予相应的迁徙概率, 改善了果蝇算法性能. 文献[5]提出判别算法局部收敛机制, 并在二次寻优时, 引入高斯变异操作对最优果蝇个体进行扰动, 实现了对算法的优化. 文献[6]介绍了基于历史认知的果蝇优化算法, 文献[7]研究了自适应混沌果蝇优化算法. 本文同样将混沌扰动策略融入到 FOA 中, 提出一种新的改进果蝇优化算.

2 基本果蝇优化算法

模拟果蝇种群寻找食物源的特点,果蝇优化算法的主要步骤如下^[8-9]:

(1) 设置群体规模 Np , 最大迭代次数 Maxgen , 果蝇种群初始位置为 $x_{\text{axis}}, y_{\text{axis}}$.

(2) 根据果蝇种群初始位置,按照式(1) 获得果蝇种群中个体的位置.

$$\begin{cases} x_i = x_{\text{axis}} + \text{RValue} \\ y_i = y_{\text{axis}} + \text{RValue} \end{cases} \quad (1)$$

式中, RValue 为搜索距离.

(3) 依据式(2) ~ (4) 求取第 i 个果蝇个体与原点的距离 d_i 、相应的味道浓度判定值 S_i 和味道浓度 smell_i .

$$d_i = \sqrt{x_i^2 + y_i^2} \quad (2)$$

$$S_i = 1/d_i \quad (3)$$

$$\text{smell}_i = f(S_i) \quad (4)$$

(4) 利用式(5) 获取种群中味道浓度最优的个体,即

$$[\text{bestsmell}, \text{bestindex}] = \min(\text{smell}) \quad (5)$$

(5) 判断 bestsmell 是否优于前一次迭代的值, 若否, 执行(7).

(6) 按照式(6) ~ (10) 保存最优味道浓度值及相应的位置, 使果蝇群体凭借自己视觉向该位置飞去.

$$\text{smellbest} = \text{bestsmell} \quad (6)$$

$$\begin{cases} x_{\text{axis}} = x(\text{bestindex}) \\ y_{\text{axis}} = y(\text{bestindex}) \end{cases} \quad (7)$$

(7) 判断是否满足结束条件(一般是最大迭代次数或设定的目标值), 若是, 则结束; 反之, 执行(2).

3 基于混沌扰动的果蝇优化算法

FOA 在整个寻优进化过程中仅向种群中的最优果蝇个体学习, 本次迭代, 产生的新的种群个体都聚集到前一次迭代时搜索到的最优果蝇个体周围附近. 若该个体处于局部极值, 极易使算法降低收敛速度和精度, 造成算法的早熟收敛. 在本文中, 为了使果蝇算法快速跳出局部极值, 将混沌扰动策略融入到 FOA 算法, 利用该策略对 FOA 算法的迭代中的最优个体在一定空间内进行遍历搜索, 以获得相对较优的最优果蝇个体.

混沌是存在非线性系统中一种现象, 其运动状态具有随机性, 但可由确定的方程获得.

混沌的一个重要特性就是遍历性, 利用混沌变量进行优化解空间的遍历搜索比一般搜索方式更容易跳出局部极值, 实现全局最优, 提高搜索效率, 传统的 Logistic 方法在映射空间内存在断点, 并且映射点在空间内分布不均, 为了提高算法全局优化搜索能力, 采用遍历性和快速性都比较好的 Tent 映射, 其模型方程如下:

$$z_{k+1} = \begin{cases} 2z_k & 0 \leq z_k \leq 0.5 \\ 2(1 - z_k) & 0.5 < z_k \leq 1 \end{cases} \quad (8)$$

式中, z_k 为混沌变量.

当前最优果蝇个体由标准果蝇优化算法得到, 然后在该解的邻域内进行混沌优化搜索, 如下:

$$x_{i,j}^{(k)} = x_{1,j}^* + \eta x_{i,j} \times z x_{i,j}^{(k)} \quad (9)$$

$$y_{i,j}^{(k)} = y_{1,j}^* + \eta y_{i,j} \times z y_{i,j}^{(k)} \quad (10)$$

式中, (x^*, y^*) 为当前最优果蝇个体的位置, $\eta x(y)_{i,j}$ 为调节系数, $z x(y)_{i,j}^{(k)}$ 为 $[0, 1]$ 区间内的混沌变量.

在搜索初期, 希望果蝇种群个体变动较大以便于跳出局部极值点, $\eta x_{i,j}, \eta y_{i,j}$ 值应取较大; 而在搜索后期, 则要求个体变动相对较小, 以提升算法在局部的开发和深究能力, $\eta x_{i,j}, \eta y_{i,j}$ 值取很小.

定义 $\eta x_{i,j}, \eta y_{i,j}$ 自适应变化:

$$\eta x_{i,j} = \gamma_1 \times \left(\frac{k_{\max} - k + 1}{k_{\max}} \right)^2 \times x_{i,j}^* \quad (11)$$

$$\eta y_{i,j} = \gamma_2 \times \left(\frac{k_{\max} - k + 1}{k_{\max}} \right)^2 \times y_{i,j}^* \quad (12)$$

式中, γ_1 和 γ_2 为邻域半径, $\gamma_1, \gamma_2 \in (0, 1)$; $k_{\max}$ 为算法设置的最大迭代次数; k 为当前的迭代次数.

基于混沌扰动的果蝇优化算法流程:

Step1 设置群体规模 Np , 最大迭代次数 Maxgen , 邻域半径 γ , 混沌扰动时的群体规模 M . 果蝇种群初始位置为 $x_{\text{axis}}, y_{\text{axis}}$.

Step 2 依据基本果蝇优化算法(2) ~ (7) 计算并比较果蝇个体的味道浓度, 保留最优味道浓度及对应的最优个体位置.

Step3 对当前最优个体按式(9) ~ (12) 执行混沌搜索, 执行基本果蝇优化算法步骤(3) ~ (6), 更新 bestsmell 及对应的最优个体位置.

Step4 判断结束条件是否满足, 若满足, 则结束; 否则, 执行 Step 2.

4 实验比较与分析

4.1 实验设计

本文选用几个常用的基准测试函数来验证新算

法的性能,如表 1 所示,并与基本果蝇算法进行对比. 相关参数设置为: $Np=30,M=30,x_{axis},y_{axis}$ 在搜索区间随机取值,RValue 在 $[-1,1]$ 内随机取值. 软件平台为 Windows7 系统、机器主频 2.6 GHz,内存 2 GB,MATLAB2013a.

表 1 测试函数

函数	函数形式	维数	搜索区间	理论极值	目标精度
Sphere	$f_1(x)=\sum_{i=1}^n x_i^2$	30	$[-100,100]$	0	10^{-5}
Griewank	$f_2(x)=\frac{1}{4\,000}\sum_{i=1}^n (x_i)^2-\prod_{i=1}^n \cos(x_i/\sqrt{i})+1$	30	$[-600,600]$	0	10^{-6}
Rosenbrock	$f_3(x)=\frac{1}{4\,000}\sum_{i=1}^{n-1}(100(x_{i+1}-x_i)^2+(x_i-1)^2)$	30	$[-100,100]$	10	30
Rastrigin	$f_4(x)=\sum_{i=1}^n (x_i^2-10\cos(2\pi x_i)+10)$	30	$[-100,100]$	0	10^{-4}
Ackley	$f_5(x)=-20\exp(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^n x_i^2})-\exp(\frac{1}{30}\sum_{i=1}^n \cos 2\pi x_i)+20+e$	30	$[-100,100]$	0	10^{-1}
Schaffer	$f_6(x)=\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{(1+0.001(x_1^2+x_2^2))^2}-0.5$	2	$[-100,100]$	-1	$10^{-5}-1$

4.2 算法性能分析

4.2.1 最大迭代次数下的收敛精度及速度

利用 FOA 和 ICFOA 分别对表 1 的测试函数进行 20 次独立运行实验,Maxgen=2 000,结果如表 2 所示.图 1 是 6 个测试函数适应度进化曲线,对于函数 f_6 ,将 f_6+1 作为适应度函数,并且函数的适应度均取以 10 为底的对数. Sphere 是峰值获得比较困难的单峰函数. Rosenbrock 是经典的复杂优化函数,由于该函数在取值区间内走势平坦,因此搜

评估性能的方式如下:(1) 固定最大迭代数,评价算法的收敛精度及速度,并和其它算法进行对比;(2) 固定目标精度,评价算法达到目标精度所需的平均迭代数和达优率,并和其它算法进行对比.

索全局最优解有一定的难度. Griewank 为典型的具有大量局部极值的多模态函数,能够检验优化算法的全局搜索能力. Rastrigin 和 Ackley 是存在很多局部极小的多峰值函数. Schaffer 为强烈振荡的多峰函数. 由表 2 可以看出,对于所有测试函数,ICFOA 的最优值、最差值、优化均值和标准差均优于 FOA. 由图 2 可以看出,ICFOA 的收敛速度比 FOA 明显有优势. 总体上,ICFOA 的收敛精度、收敛速度和鲁棒性均显著提高.

表 2 FOA 和 ICFOA 的性能比较

函数	最优值		最差值		优化均值		标准差	
	FOA	ICFOA	FOA	ICFOA	FOA	ICFOA	FOA	ICFOA
f_1	1.539 5e-04	1.897 7e-22	1.628 5e-04	3.878 3e-22	1.577 7e-04	2.953 9e-22	2.071 5e-06	5.943 8e-23
f_2	8.059 3e-06	4.551 9e-15	8.309 1e-06	1.243 4e-14	8.152 9e-06	8.126 8e-15	6.640 5e-08	2.143 3e-15
f_3	27.641 9	26.962 0	30.702 8	27.800 2	28.540 3	27.735 4	0.688 4	0.182 7
f_4	0.020 6	1.477 9e-12	0.021 7	3.467 4e-12	0.021 1	2.464 2e-12	3.043 9e-04	5.291 6e-13
f_5	0.009 4	2.739 6e-11	0.009 5	3.769 9e-11	0.009 5	3.295 3e-11	4.284 7e-05	2.253 2e-12
f_6	-1.000 0	-1.000 0	-0.990 3	-1.000 0	-0.999 5	-1.000 0	0.002 2	2.922 1e-07

表 3 为 ICFOA 与参考文献算法的优化均值比较(注:文献[4]中 $Np=30$,Maxgen=800, f_4 对应的搜索范围为 ± 5.12 ,文献[7]中 $Np=15$,Maxgen=200,文献中未提及 $f_1\sim f_6$ 相应的搜索范围、 Np 和 Maxgen 同文中设置). 由表 3 可以看出,对于 Sphere(f_1)、Rosenbrock (f_3)和 Ackley(f_5),ICFOA 的优化均值均优于其它算法,对于 Griewank

(f_2), ICFOA 的优化均值优于 AFOABM 和 FOAAM,比 FOABHC 和 ACFOA 差. 对于 Rastrigin(f_4), ICFOA 的优化均值优于 AFOABM,比 FOAAM 和 FOABHC 和 ACFOA 差. 对于 Schaffer (f_6), ICFOA 的收敛精度与参考文献的算法相同. 表中“—”表示文献中没有涉及.

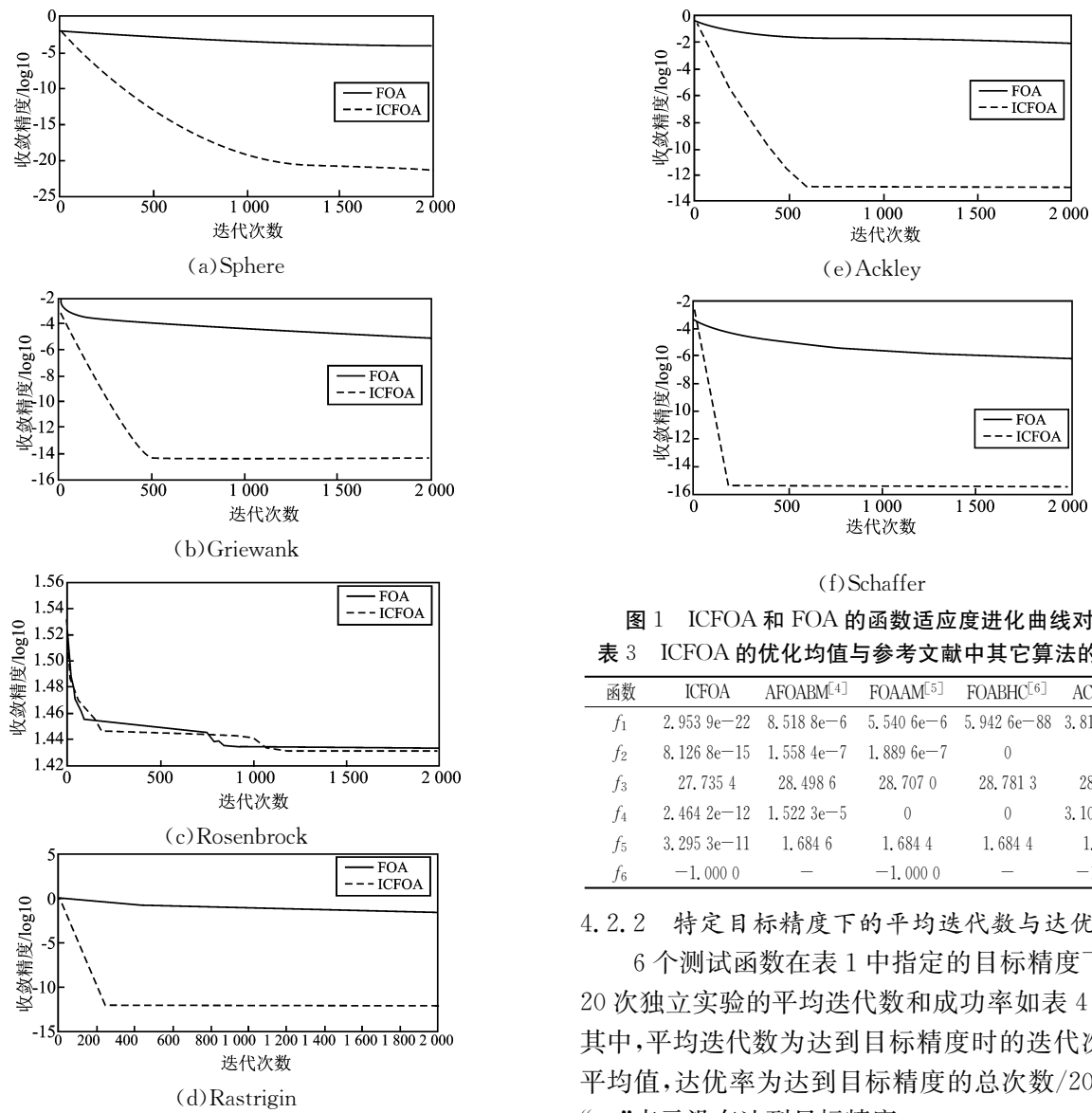


图 1 ICFOA 和 FOA 的函数适应度进化曲线对比
表 3 ICFOA 的优化均值与参考文献中其它算法的比较

函数	ICFOA	AFOABM ^[4]	FOAAM ^[5]	FOABHC ^[6]	ACFOA ^[7]
f_1	2.953 9e-22	8.518 8e-6	5.540 6e-6	5.942 6e-88	3.812 6e-21
f_2	8.126 8e-15	1.558 4e-7	1.889 6e-7	0	0
f_3	27.735 4	28.498 6	28.707 0	28.781 3	28.732 7
f_4	2.464 2e-12	1.522 3e-5	0	0	3.108 6e-15
f_5	3.295 3e-11	1.684 6	1.684 4	1.684 4	1.684 4
f_6	-1.000 0	—	-1.000 0	—	-1.000 0

4.2.2 特定目标精度下的平均迭代数与达优率

6 个测试函数在表 1 中指定的目标精度下经过 20 次独立实验的平均迭代数和成功率如表 4 所示. 其中,平均迭代数为达到目标精度时的迭代次数的平均值,达优率为达到目标精度的总次数/20. 表中“—”表示没有达到目标精度.

表 4 ICFOA 的平均迭代数和达优率与参考文献中其它算法的比较

函数	目标精度	FOA 平均迭代数/达优率	ICFOA 平均迭代数/达优率	AFOABM ^[4] 平均迭代数/达优率	FOAAM ^[5] 平均迭代数/达优率	FOABHC ^[6] 平均迭代数/达优率	ACFOA ^[7] 平均迭代数/达优率
f_1	10^{-5}	—/0	54.4/1	532/1	1416/1	28.0/1	1/1
f_2	10^{-6}	—/0	71.2/1	98/1	381/1	13.85/1	1/1
f_3	30	231.7/1	8.0/1	1/1	—	—	1/1
f_4	10^{-4}	—/0	77.0/1	313/1	82/1	30.0/1	2.1/1
f_5	10^{-1}	131.2/1	86.1/1	—/0	—/0	—/0	—/0
f_6	$10^{-5}-1$	463.9/1	20.5/1	—	—	—	138.6/1

由表 4 可知,对于 Sphere(f_1)、Griewank(f_2)和 Rastrigin(f_4),FOA 没有达到目标精度,ICFOA 和参考文献中的 AFOABM、FOAAM、FOABHC 及 ACFOA 都达到目标精度,且达优率均是 100%,但 ICFOA 平均迭代数比 FOA、FOAAM 和 AFOABM 要少,比 FOABHC 和 ACFOA 多,对于 Rosenbrock (f_3),ICFOA 的平均迭

代数略高于 AFOABM 和 ACFOA,只占 FOA 平均迭代数的 0.035%. 对于 Ackley(f_5),ICFOA 的平均迭代数为 86.1,达优率为 100%,无论平均迭代数还是达优率均优于其它算法. 对于 Schaffer(f_6),ICFOA 的平均迭代数占 FOA 平均迭代数的 0.044%,占 ACFOA 平均迭代数的 1/6 左右.

从整体上讲,ICFOA 在收敛精度、收敛速度及稳定性上均优于 FOA,与 AFOABM、FOAAM 和 ACFOA 比较,其收敛可靠性和精度相当好.

5 结束语

本文将混沌扰动策略引入基本果蝇优化算法,提出一种新颖的改进果蝇优化算法.在寻优迭代过程中,使用混沌策略对最优果蝇个体进行扰动,进行二次寻优,以摆脱局部极值的困扰,寻找到更优的果蝇个体.基准测试函数的结果证明,与基本果蝇优化算法相比,新算法具有比较好的收敛精度、收敛速度和鲁棒性.

参考文献:

[1] 潘文超. 果蝇最佳化演算法—最新演化式计算技术[M]. 台湾: 沧海书局,2011.

[2] 刘志雄,王雅芬,张煜. 多种群果蝇优化算法求解自动化仓库拣选作业调度问题[J]. 武汉理工大学学报, 2014,33(3):71-77.

[3] 王克甫,薛鹏,黄全振,等. 求解旅行商问题的改进果蝇算法[J]. 计算机工程与设计, 2014, 34(8):

2789-2792.

[4] 刘成忠,韩俊英. 基于细菌迁徙的自适应果蝇优化算法[J]. 计算机工程与科学,2014,36(4):690-696.

[5] 韩俊英,刘成忠. 自适应变异的果蝇优化算法[J]. 计算机应用研究,2013,30(9):2641-2644.

[6] 韩俊英,刘成忠. 基于历史认知的果蝇优化算法[J]. 计算机科学与探索,2014,8(3):368-375.

[7] 韩俊英,刘成忠. 自适应混沌果蝇优化算法[J]. 计算机应用,2013,33(5):1313-1316.

[8] Pan W T. Using modified fruit fly optimization algorithm to perform the function test and case studies[J]. Connection Science,2013,25(2/3):151-160.

[9] Pan W T. A new fruit fly optimization algorithm; taking the financial distress model as an example[J]. Knowledge-Based Systems,2012,26(2):69-74.

作者简介:

李宗岳 男,(1988-),硕士研究生.研究方向为嵌入式控制系统开发与应用、智能算法优化.

陈志军(通讯作者) 男,(1967-),教授,研究生导师.研究方向为智能信息处理和智能控制与系统开发.

E-mail:chenzj1110@163.com.

(上接第 63 页)

其进行修正并更新自己的时间码,从而解决了动态可重构总线各节点间时间同步的问题^[6],使总线上所有节点维持在一个时间系统内.通过仿真测试表明同步误差最高在 100 ns 左右,为总线系统提高时间确定性提供了技术保障,同时也可满足总线时隙令牌仲裁方案对同步精度的需求.以 16 通道动态可重构总线在 Virtex-5 系列 FPGA 上实现为验证,在每通道速率为 100 Mb/s 的实际系统中,该方法具有低开销,高可靠的特性,满足动态可重构总线的需求.

参考文献:

[1] Wang Jiulong. Development state and thought of the satellite synthesized electronic system [J]. Spacecraft Engineering, 2007, 16(5): 68-73.

[2] 崔维嘉,樊少杰. 新一代的总线结构—PapidIO[J]. 通信技术,2001,34(4):49-50.

[3] Du Rui, Zhang Weigong, Deng Zhe, et al. Parallel CRC algorithm design and implementation of new bus

[J]. Computer Engineering and Dedign, 2013, 34(1): 131-135.

[4] Deng Zhe, Zhang Weigong, Zhu Xiaoyan, et al. Design and implementation of data transmission management method for dynamic reconfigurable bus [J]. Computer Engineering, 2013, 39(1): 265-269.

[5] 杨毅. MLVDS 标准介绍及德州仪器 SN65MLVD200 系列[J]. 今日电子,2003,(29):53-55.

[6] 王铮. IEEE1588 时钟同步协议的研究与实现[J]. 计算机网络,2012,37(23):53-55.

作者简介:

李超 男,(1990-),硕士研究生.研究方向为高可靠嵌入式系统设计.

邱柯妮(通讯作者) 女,(1978-),博士,讲师.研究方向为高可靠嵌入式系统设计. E-mail:qiukn@chu.edu.com.

张伟功 男,(1967-),博士,研究员.研究方向为高可靠嵌入式系统设计.

张少楠 男,(1990-),硕士研究生.研究方向为高可靠嵌入式系统设计.