

具有自适应特性的改进鸡群算法

郑家明¹, Ryad Chellali¹, 陈 闯¹, 邢尹²

(1 南京工业大学 电气工程与控制科学学院, 江苏 南京 211816;

2 桂林理工大学 测绘地理信息学院, 广西 桂林 541004)

摘 要: 针对基本鸡群算法存在早熟收敛和易陷入局部最优的问题, 提出了一种具有自适应能力的改进鸡群算法 (ICSO). 首先通过佳点集构造初始种群, 使初始解能够均匀的分布在解空间中, 提高种群的多样性; 然后根据算法的实际运行情况, 引入寻食速度因子和聚集度因子, 同时根据这两个参数对惯性权值的影响, 把惯性权值表示为二者的一个函数; 最后把惯性权值加入到母鸡的位置更新公式中, 从而使算法根据实际运行情况动态调整惯性权值, 具有自适应性. 通过对 8 个基准函数的仿真实验, 并与其它几种算法进行对比, 结果证明了改进鸡群算法的优越性.

关键词: 群智能算法; 鸡群算法; 佳点集; 惯性权值; 自适应

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2018)10-0093-06

Improved Chicken Swarm Optimization Algorithm With Adaptive Features

ZHENG Jia-ming¹, Ryad CHELLALI¹, CHEN Chuang¹, XING Yin²

(1 College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing 211816, China;

2 College of Geomatics and Geoinformation, Guilin University of Technology, Guilin Guangxi 541004, China)

Abstract: In order to solve the problem that the basic chicken swarm optimization algorithm is premature convergence and easy to fall into the local optimum, an improved chicken swarm optimization algorithm with adaptive capability is proposed. Firstly, the initial population is constructed by the good point set, so that the initial solution can be uniformly distributed in the solution space and improve the diversity of the population. Then, according to the actual operation of the algorithm, the feeding speed factor and aggregation degree factor are introduced. And the inertia weight is expressed as a function of the two. Finally, the inertia weight is added to the position updating formula of the hen, so that the algorithm dynamically adjusts the inertia weight according to the actual running condition. Adaptability. Through the simulation experiments of eight benchmark functions and comparison with other algorithms, the results prove the superiority of improved chicken swarm optimization algorithm.

Key words: swarm intelligence optimization algorithm; chicken swarm optimization algorithm; good point set; Inertial weights; adaptive

1 引言

鸡群算法 (Chicken Swarm Optimization, CSO) 是一种根据自然界中鸡群的等级次序和生活

习性而抽象得到的智能算法, 与其它群智能算法相比, 鸡群算法具有原理简单、易于实现、全局搜索能力强等优点, 因此可以取得较好的收敛速度和收敛精度. 尽管 CSO 算法得到了广泛的应用, 但是基本

CSO 算法跟其它群智能算法一样,同样存在着早熟收敛和易陷入局部最优的问题.针对 CSO 算法存在的问题,不少研究学者都提出了自己的改进方法,如杨菊靖等^[1]提出了一种混合改进的鸡群算法,算法采用反向学习对种群进行初始化,同时对越界个体进行边界变异操作来保证算法的种群多样性,但是边界变异的方法效果一般,寻优精度也不够理想;李振壁等^[2]提出了一种基于模拟退火的改进鸡群算法,算法在小鸡的位置更新中加入向子群中公鸡学习部分,并采用模拟退火算法进行邻域搜索,使算法具有跳出局部最优进而得到全局最优解的能力,但该改进策略增加了计算机的运行负担;肖亮等^[3]提出了一种基于认知多样性变异的鸡群算法,通过采用逆序协同优化异步实现策略,并使个体最优以一定概率发生变异提高了算法得到全局最优解的能力,但是造成了算法收敛速度过慢.

为了解决基本鸡群算法存在的早熟收敛和易陷入局部最优的问题,本文提出了一种具有自适应性的改进鸡群算法.改进算法中用佳点集构造初始种群,保证初始种群的多样性,避免了随机方式初始种群的不确定性;在母鸡的位置更新公式中加入自适应权值因子,算法能够根据实际迭代过程中的复杂性动态调整权值因子,使算法具有跳出局部极值的能力.在 8 个测试函数上的仿真结果验证了改进鸡群算法 (Improved Chicken Swarm Optimization, ICSO) 的有效性.

2 改进鸡群算法

2.1 佳点集初始化种群

Haupt 等^[4]研究认为,群智能算法的寻优性能和初始种群的多样性具有一定关系,多样性较好的初始种群可以有效提高算法的收敛速度和精度.基本鸡群算法以随机方式生成初始种群,不能够保证种群的多样性,进而影响算法的性能,因而我们可以采用佳点集方法对种群进行初始化.佳点集技术^[5,6]已经被证明是一种提高随机搜索算法能力的有效方法,佳点集可以使初始个体在搜索空间中分布更均匀,有效保证了初始种群的多样性,提高了算法的稳定性.本文采用佳点集技术对种群进行初始化,使初始个体均匀的分布在搜索空间中,提高鸡群算法的寻优性能.

佳点集的定义和性质详见参考文献[5][6].在

S 维欧氏空间单位立方体 G_s 上近似计算随机 n 个点的函数值组成的加权和的积分时,取 n 个佳点所得的误差最小,即:

$$E_n = \mu - Q_n \quad (1)$$

式中,

$$\mu = \int_{x \in G_s} f(x) dx, Q_n = \sum_{i=1}^n \frac{f(p_n(i))}{n} P_n(k)$$

($1 \leq k \leq n$) 是一个含有 n 个点的点集,也即是说,相同条件下由佳点集得到的初始种群要比随机方式更均匀,种群的多样性更好.另外佳点集的构造与维数无关,这也说明了佳点集的普遍适用性.

在 $\dim$ 维空间中取佳点的方法如下:

$\dim$ 维空间 H 中作含 n 个点的佳点集

$$P_n(k) = \{(\{r^1 * k\}, \dots, \{r^i * k\}, \dots, \{r^t * k\}), k = 1, 2, \dots, n\}$$

$$r^i = \left\{ 2 \cos \frac{2\pi i}{p} \right\}, 1 \leq i \leq t, P \text{ 是满足 } p \geq 2t +$$

3 的最小素数,或者 $r^i = e^i, 1 \leq i \leq t, \{r^i * k\}$ 表示 $r^i * k$ 的小数部分.

2.2 引入自适应惯性权值

在鸡群算法的寻优过程中,开始迭代阶段,算法需要在较大的搜索空间内快速搜索,个体的位置变化很大,即适应度值变化很快;到了迭代后期算法,由于要在小空间内搜索,个体的位置变化相对不大,即适应度值变化变慢.具体来说,例如优化目标是寻找极小值,则有 $f(x_{i,j}^t) \leq f(x_{i,j}^{t-1})$, 定义 $h = f(x_{i,j}^t) / f(x_{i,j}^{t-1})$;同理如果是寻找极大值,则有 $f(x_{i,j}^t) \geq f(x_{i,j}^{t-1})$, 定义 $h = f(x_{i,j}^{t-1}) / f(x_{i,j}^t)$. 所以无论是寻找极大值还是极小值, h 都可以表示为:

$$h = \frac{\min(f(x_{i,j}^t), f(x_{i,j}^{t-1}))}{\max(f(x_{i,j}^t), f(x_{i,j}^{t-1}))} \quad (2)$$

式中, h 为鸡群寻食速度. 根据推导可知 $0 < h \leq 1$, h 跟算法的实际运行情况有关,真实的反映了算法迭代过程中的速度变化.即 h 值越小,鸡群的寻食速度越快,经过了若干次迭代后 $h = 1$,说明算法已经收敛或者找到了最优解.

反映算法实际运行情况的另一个因素是鸡群的聚集度.在算法的运行过程中,种群的方差反映了种群的离散度,如果 μ 代表所有个体当前适应度值的平均值,则 $\mu = \sum_{i=1}^N f(x_{i,j}^t) / N$. 其中 N 为种群总数, $f(x_{i,j}^t)$ 是个体 i 在 j 维空间中迭代次数为 t 时的适

应度值,则鸡群方差 $\sigma^2 = \sum_{i=1}^N (f(x_{i,j}^t) - \mu)^2 / N$.

定义:

$$s = e^{-\sigma^2} \quad (3)$$

式中, s 为鸡群聚集度, $0 < s \leq 1$. 种群方差越小,说明种群越聚集,则种群聚集度越大,同理种群方差越大,说明种群越分散,则种群聚集度越小. 经过了若干次迭代后 $s = 1$,说明种群个体具有同一性.

Shi 和 Eberhart^[7] 研究发现权值因子 w 较大时算法具有较强的全局搜索能力, w 较小时算法的局部搜索能力更强. 事实上, w 的大小应该随着鸡群的寻食速度和鸡群的聚集度而改变,即 w 可以表示为 h 和 s 的函数:

$$w = f(h, s) \quad (4)$$

当鸡群寻食速度较快时,说明算法在进行大范围的寻优搜索;而当鸡群寻食速度减慢时,即 h 变大,说明算法可能已经锁定最优解范围,此时应减小 w 的值,可以保证算法在小空间内快速找到最优解. 当鸡群较分散时,算法就不易陷入局部最优,而当鸡群的聚集度增大时,即 s 变大,算法就极易陷入局部最优,此时应增大 w 的值,提高算法的全局搜索能力,避免算法陷入局部最优. 由此可知,权值因子 w 应该随着鸡群寻食速度的降低而减小,随着聚集度的增大而增大, w 可以表示为:

$$w = a_0 + a_1 * e^{-a_2 * \frac{h}{s}} \quad (5)$$

式中 a_0, a_1, a_2 均是 $(0, 1]$ 之间的数.

在整个鸡群中,一方面母鸡的数量最多,占大多数,因此母鸡对于鸡群算法的寻优速度和精度起着重要作用;另一方面母鸡也是联系公鸡和小鸡的纽带,起着信息传递的作用. 所以,我们在母鸡的位置更新中引入自适应权值因子,母鸡的位置更新公式改进如下:

$$x_{i,j}^{t+1} = w * x_{i,j}^t + S1 * Rand * (x_{r1,j}^t - x_{i,j}^t) + S2 * Rand * (x_{r2,j}^t - x_{i,j}^t) \quad (6)$$

式中, w 为自适应权值因子.

3 算法流程

Step1: 初始化参数. 算法最大迭代次数 T , 种群数 N , 空间维度 $\dim$, 更新代数 G , 公鸡、母鸡和小鸡的数量, 用佳点集构造初始种群.

Step2: 计算种群个体适应度值, 并初始化个体

最优位置和全局最优位置.

Step3: 若 $\text{mod}(t, G) = 1$, 根据种群个体的适应度值对所有个体进行排序, 然后划分子群并建立等级秩序.

Step4: 根据式(2)、(3)、(4)计算 h, s 和 w , 然后更新公鸡、母鸡和小鸡的位置.

Step5: 计算当前种群个体的适应度值, 更新并保存当前的个体最优位置和全局最优位置.

Step6: 若满足迭代停止条件, 则停止迭代并输出最优解, 否则转到 Step3.

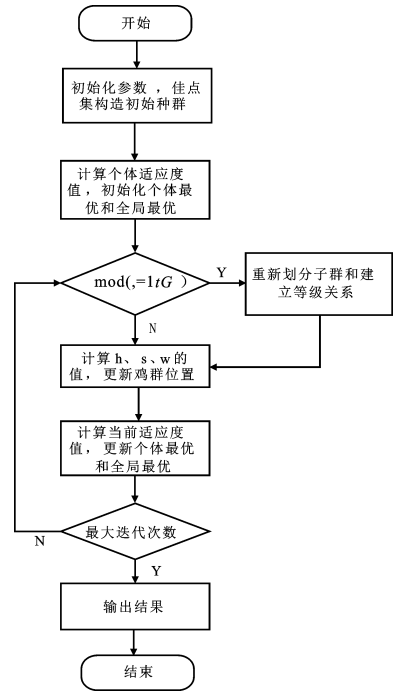


图1 改进鸡群算法流程图

4 数值分析与实验

4.1 测试函数和实验设置

为了验证本文改进算法(ICSO)的有效性,选取了8个标准测试函数^[8]进行仿真实验,并与基本鸡群算法(CSO)、灰狼算法(GWO)、引力搜索算法(GSA)进行对比分析,表1给出了各个测试函数的一些基本信息. 4种算法的相关参数设置见表2. 遵循平等原则,各算法的种群规模均设为30,空间维数均为30,最大迭代次数均为500.

4.2 实验结果与分析

为了避免偶然因素对结果造成影响,各算法均独立运行30次,取30次结果的平均值和标准差来评价算法的性能,测试结果见表3.

表 1 基准测试函数

测试函数名称	测试函数	维度	范围	最小值
Sphere	$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100,100]$	0
Schwefel's Problem2. 21	$F_2(x) = \max_{i=1}^n \{ x_i \}$	30	$[-100,100]$	0
Schwefel's Problem2. 22	$F_3(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10,10]$	0
Quartic	$F_4(x) = \sum_{i=1}^n ix_i^4 + random[0,1)$	30	$[-1.28,1.28]$	0
Rastrigin	$F_5(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12,5.12]$	0
Ackley	$F_6(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-32,32]$	0
Griewank	$F_7(x) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600,600]$	0
Penalized	$F_8 = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i,10,100,4)$	30	$[-50,50]$	0

表 2 4 种算法相关参数设置

算法	参数设置
ICSO	$NC = N - NR - NH, NR = 0.15N, NH = 0.7N, NM = 0.5NH, G = 10, FL \in [0.4,1], a_0 = 0.3, a_1 = 0.7, a_2 = 1.0$
CSO	$NR = 0.15N, NH = 0.7N, M = 0.5NH, NC = N - NR - NH, NG = 10, FL \in [0.4,1]$
GWO	$T = 500, N = 30, \dim = 30$
GSA	$\alpha = 20, G_0 = 100$

从表 3 可知,ICSO 算法相对于 CSO 算法来说,除函数 F_3 外,ICSO 算法的寻优精度都要明显优于 CSO,并且算法的稳定性也得到了显著提高;ICSO

算法相对于 GWO 算法,除函数 F_8 外,ICSO 算法的精度和稳定性均要好于 GWO 算法;相对于 GSA 算法,ICSO 算法在 8 个测试函数上均取得了优良的成绩,性能远远优于 GSA 算法. 综上所述,ICSO 算法与其它三种算法在 8 个测试函数上的寻优结果表明:ICSO 算法除对个别函数寻优稍显不足外,对于其它函数,ICSO 算法在寻优精度和算法稳定性上均优于其它三种算法,特别是和 CSO 和 GSA 算法相比,ICSO 算法具有明显优势.

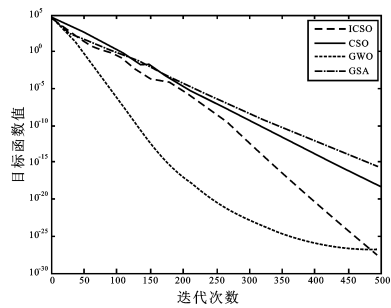
为了更加直观的体现 ICSO 算法的优越性,4 种不同算法对 8 个基准测试函数的目标函数值曲线,如图 2 所示.

表 3 测试函数测试结果

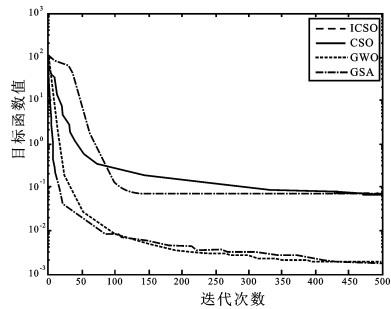
Function	ICSO		CSO		GWO		GSA	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	1.2188E-28	4.8801E-28	3.7343E-19	1.0170E-18	1.2377E-27	1.4437E-27	2.2103E-16	8.9472E-17
F_2	8.6970E-14	3.7901E-13	27.172 8	12.527	6.3990E-07	5.3976E-07	6.848 3	2.413 7
F_3	7.8454E-19	2.6617E-18	7.2611E-19	1.7038E-18	8.9766E-17	6.1960E-17	0.180 5	0.432 2
F_4	1.7000E-03	7.1456E-04	0.068 3	0.08	1.9000E-03	9.9541E-04	0.073	0.038 1
F_5	0	0	2.770 6	7.291 7	1.278 5	2.083 2	27.892	6.828 3
F_6	4.4409E-15	0	6.7593E-11	1.0171E-10	1.0072E-13	1.5577E-14	1.2053E-08	2.8261E-09
F_7	0	0	0.023 6	0.066 9	1.7000E-03	4.2000E-03	26.866 3	5.649 7
F_8	0.406 9	0.085 6	3.5759E+04	1.3608E+05	0.055 2	0.026 3	3.711 1	2.392 5

由收敛曲线可知,对函数 F_1 和 F_6 来说,ICSO 算法的寻优结果均要好于其它三种算法;对函数 F_3 ,ICSO 算法与 CSO 算法持平,但要优于 GWO 和

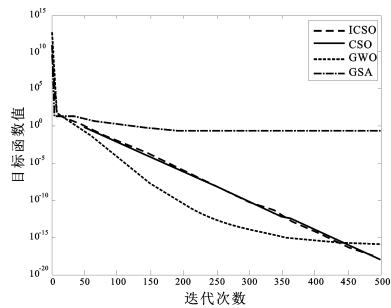
GSA 算法;对函数 F_4 ,ICSO 算法与 GWO 算法持平,但要优于 CSO 和 GSA 算法;对于函数 F_2, F_5 和 F_7 来说,ICSO 算法相对于其它三种算法而言,在精



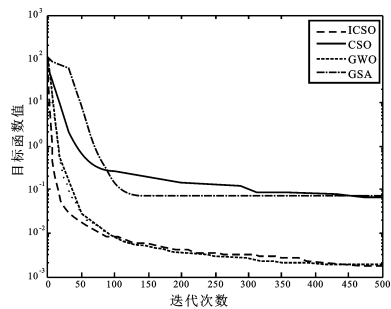
(a) F_1 函数的收敛曲线



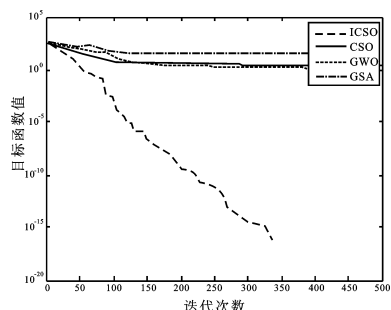
(b) F_2 函数的收敛曲线



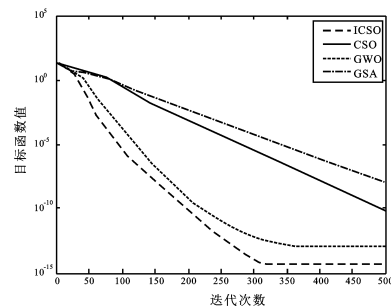
(c) F_3 函数的收敛曲线



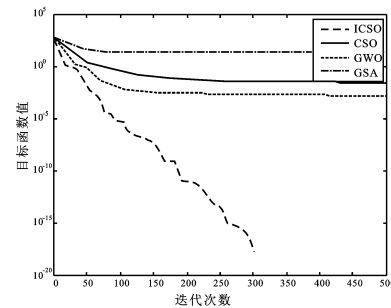
(d) F_4 函数的收敛曲线



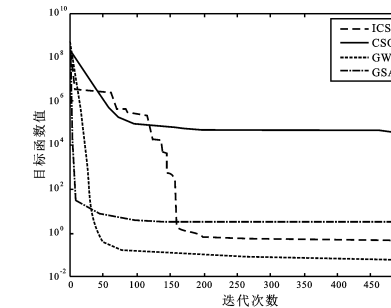
(e) F_5 函数的收敛曲线



(f) F_6 函数的收敛曲线



(g) F_7 函数的收敛曲线



(h) F_8 函数的收敛曲线

图 2 4 种不同算法对 8 个基准测试函数的目标函数值曲线

度和速度上都得到了明显的提高;对函数 F_8 , ICSO 算法略逊色于 GWO 算法,但是还是要优于 CSO 和 GSA 算法. 综合来说, ICSO 算法的寻优性能要优于另外 3 种算法.

5 结束语

针对基本鸡群算法存在早熟收敛和搜索精度不高的问题, 本文提出了一种具有自适应性的改进鸡群算法. 改进算法中通过佳点集构造初始种群, 保证了种群的多样性; 在母鸡的位置更新中引入自适应权值因子, 使算法根据实际运行情况来平衡全局搜索和局部搜索之间的关系, 提高算法跳出局部最优的能力. 通过在 8 个基准测试函数上的仿真实验, 并与其它几种算法进行结果对比, 表明改进鸡群算法不仅能够避免早熟收敛, 而且提高了算法的搜索

精度.

参考文献:

[1] 杨菊靖,张达敏,张慕雪. 一种混合改进的鸡群优化算法[J/OL]. 计算机应用研究,2018,35(11):[2017-11-10]. <http://kns.cnki.net/kcms/detail/51.1196.TP.20171115.1041.006.html>.

[2] 李振璧,王康,姜媛媛. 基于模拟退火的改进鸡群优化算法[J]. 微电子学与计算机,2017,34(2): 30-33.

[3] 肖亮,刘思彤. 基于认知多样性变异的鸡群算法协同优化异步实现[J]. 计算机科学,2017,44(6A): 99-104.

[4] Haupt R, Haupt S. Practical genetic algorithm[M]. New York: John Wiley and Sons, 2004.

[5] 张铃,张钊. 佳点集遗传算法[J] 计算机学报,2001,24(9): 917-922.

[6] 程军盛,张铃. 基于佳点集遗传算法求解 Job-shop 调度问题[J]. 计算机科学,2002,29(4): 67-68.

[7] Shi Y H, Eberhart R C. A modified particle swarm optimizer[C]// Proceeding s of the IEEE Congresson Evolutionary Computation. Piscataway, USA: IEEE Service Center, 1998, 69-73.

[8] 陈闯, Ryad Chellali, 邢尹. 采用动态权重和概率扰动策略改进的灰狼优化算法[J]. 计算机应用, 2017(12): 3493-3497.

作者简介:

郑家明 男,(1991-),硕士研究生. 研究方向为机器学习、图像处理. E-amil:ngdzjm@163.com.

Ryad Chellali 男,(1964-),教授,博士. 研究方向为机器人、虚拟和增强现实、机器学习等.

陈闯 男,(1992-),硕士研究生. 研究方向为最优化理论、语音信号处理.

邢尹 女,(1992-),硕士研究生. 研究方向为机器学习、GNSS 技术及应用.

(上接第 92 页)

表 1 LDO 的性能对比

文献	工艺/ μm	PSR/dB	噪声/ μV
[5]	0.6	38dB@100kHz	14 μV_{RMS} @1kHz~100kHz
[6]	0.25	50dB@10kHz	14.8 μV_{RMS} @10Hz~100kH
[7]	0.18	60dB@1MHz	—
本文	0.6	72dB@10kHz	15uV _{RMS} @100Hz~100kHz

参考文献:

[1] Gupta V, Rincon M G. A 5mA 0.6um CMOS Miller Compensated LDO regulator with -27dB worst case power supply rejection using 60pF of on-chip capacitance[C]//ISSCC. San Fancisco, CA, USA: IEEE , 2007:520-521.

[2] 王建伟,张启帆,张先仁等. 一种用于 VCO 供电的低噪声 LDO[J]. 微电子学,2015,45(5):602-606.

[3] EL-Nozahim, Amer A, Torres J, et al. High PSR low drop-out regulator with feed-forward ripple cancellation technique[J]. IEEE Journal of Solid-state Circuit, 2010,45(3):565-577.

[4] 阴亚东,阎跃鹏. 一种低噪声高电源抑制比 CMOS 低压差线性稳压器[J]. 固体电子学研究进展,2013,33(6):571-577.

[5] Wonseok Oh, Bakkaloglu B , Aravind B, et al. A Low 1/f Noise CMOS Low-Dropout Regulator with Current-Mode Feedback Buffer Amplifier[J]. IEEE Cus-

tom Integrated Circuits Conference, 2015, 54(10): 213-216.

[6] Magod R, Suda N, Ivanov V, et al. A low-noise output capacitorless low-dropout regulator with a switched-RC bandgap reference[J]. IEEE Transactions on Power Electronics, 2017,32(4):2856-2864.

[7] Lim Y , Lee J, Lee Y, et al. An External Capacitor-Less Ultralow-Dropout Regulator Using a Loop-Gain Stabilizing Technique for High Power-Supply Rejection Over a Wide Range of Load Current[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems ,2017,25(11):3006 - 3018.

[8] 王忆. 高性能低压差线性稳压器研究与设计[D]. 杭州:浙江大学,2010.

[9] 张龙,来强涛,刘生有等. 一种全负载稳定的高 PSRR LDO 的设计[J]. 微电子学与计算机,2017,34(2): 58-62.

作者简介:

黄松 男,(1993-),硕士研究生. 研究方向为模拟集成电路设计. E-mail:1085796485@qq.com.

姜洪雨 男,(1977-),硕士,高级工程师. 研究方向为专用集成电路设计.

刘智 男,(1972-),博士,研究员. 研究方向为通信集成电路及数模混合集成电路设计.

耿增建 男,(1958-),研究员. 研究方向为专用集成电路设计.