

基于模拟退火机制的精英协同进化算法

贺玫璐, 罗 杰

(南京邮电大学 自动化学院, 江苏 南京 210046)

摘 要: 为了获取更好的全局寻优性能, 同时保持较快的收敛速度, 文中结合精英策略、协同进化思想和模拟退火机制, 提出了一种基于模拟退火机制的精英协同进化算法(SACEA)。算法维持三个种群: 精英种群、普通种群和随机种群。精英个体组团, 并和其他组员个体协作或对其引导来达到进化目的。SACEA 算法在精英组团过程中引入随机种群以增加种群多样性, 同时随机个体和精英个体的合作采用快速模拟退火机制来实现, 使算法获得了更好的全局寻优性。通过对 15 组标准测试函数的仿真, 并和已有的算法进行对比, 很容易得出: SACEA 算法具有更强的全局寻优能力, 同时收敛速度也有所提高。

关键词: 精英策略; 协同进化; 模拟退火; 收敛速度; 全局寻优能力

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2015)01-0091-05

doi: 10.3969/j.issn.1673-629X.2015.01.021

Elite Co-evolutionary Genetic Algorithm Based on Simulated Annealing Mechanism

HE Mei-lu, LUO Jie

(College of Automation, Nanjing University of Posts & Telecommunications,
Nanjing 210046, China)

Abstract: In order to get better global optimization performance and maintain the fast convergence speed, combined with the elite strategy and the concept of co-evolutionary and simulated annealing mechanism, put forward a new algorithm, that is the elite co-evolutionary genetic algorithm based on simulated annealing method (SACEA). The algorithm maintains three populations including elite population, common population and stochastic population. And then elite individuals form teams and exchange information with other team members with the cooperating operation or leading operation. SACEA introduces the stochastic population to evolution to improve diversity of population, at the same time, the stochastic individual and the elite individual using fast simulated annealing method to realize the purpose of cooperation. Through all above, the algorithm gets the better global optimization performance. Simulation on 15 standard test simulation and compared with existing algorithms, it is clearly shown that SACEA has better ability of searching globally optimal solution and makes an improvement in convergence speed.

Key words: elitist strategy; co-evolution; simulated annealing; convergence speed; ability of global optimization

0 引 言

协同进化解决了早熟收敛问题, 有很强的搜索和自学习能力, 已经得到广泛应用。Potter 等提出了合作型协同进化遗传算法(CCGA)^[1]。Danoy 等提出了竞争型协同进化算法(hLCGA)^[2]。王磊等提出了一种协同人工免疫计算模型^[3]。

群体中的精英个体和全局最优解之间的亲和度要大于群体中其他个体。因此, 合理利用精英个体对种

群进化的推动作用, 设计协同进化算法是目前的一大研究热点。孟伟等提出了蜜蜂进化型遗传算法^[4]。慕彩虹等提出了 M 精英协同进化算法^[5]。刘全等提出了双精英协同进化算法^[6]。周昌乐等采用精英策略提出了一种基于遗传算法的宋词生成计算模型^[7]。应伟勤等结合精英主义提出了一种集合热力学选择^[8]。

基于 Metropolis 原则设计的模拟退火算法一定能找到全局最优解, 但它存在收敛速度慢的问题。文献

收稿日期: 2014-01-17

修回日期: 2014-04-23

网络出版时间: 2014-11-17

基金项目: 国家自然科学基金资助项目(61071167)

作者简介: 贺玫璐(1988-), 女, 硕士研究生, 研究方向为智能算法、模式识别与人工智能等; 罗 杰, 博士, 教授, 研究方向为分布式智能控制、群体智能等。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20141117.2202.007.html>

[9]提出了一种改进的模拟退火算法,极大地提高了其收敛速度。王庆荣等^[10]在公交排班优化上采用模拟退火改进的遗传算法,取得了很好的效果。高鹰等提出了基于模拟退火的粒子群优化算法^[11],算法效果良好。

为了更好地提升全局寻优性能,文中在上述算法的基础上,在精英协同进化的基础上加入模拟退火机制,提出了一种基于模拟退火机制的精英协同进化算法(SACEA)。

1 协同进化算法

协同进化算法由于其高并行性和种群之间紧密联系的性能,在高维数值优化和实践方面都得到了很好的应用。

目前,众多学者通过精英结合协同进化,提出了许多算法。比如,文献[5]提出的M精英协同进化算法,它是通过精英种群和普通种群合作来达到进化目的的,各个精英各自组团,通过主导精英和精英团员的协作操作,以及对普通团员的引导操作来达到精英带动进化的目的。文献[6]提出的双精英协同进化算法,也是划分精英种群和普通种群,然后从精英种群中选取两个代表精英,采取不同的进化策略,一个用来加快算法的收敛速度,一个用来保持种群多样性。文中的SACEA算法也是基于精英协同进化算法提出的。

2 模拟退火

模拟退火算法1953年被提出,已被广泛应用到组合优化领域^[12-13]。它的提出是基于马尔可夫链的遍历性理论,能有限度地接受差解,因此有较强的全局寻优能力。

在保证全局寻优能力的基础上,为了提高收敛速度,文献[9]提出了一种改进的模拟退火算法,采用最优解记忆并及时更新;提出了新的温度更新函数,当某温度下某状态被多次接受,则加大温度降低幅度,否则减小温度降低幅度;同时采用双阈值控制抽样和退火过程,连续max1次最优解不变,则结束退火;某温度下连续max2次状态不变,则结束抽样。

在对以上内容深入研究的基础上,文中提出了基于模拟退火机制的精英协同进化算法,具体介绍如第三节。

3 基于模拟退火机制的精英协同进化算法(SACEA)

3.1 概念介绍

在介绍算法之前,先来介绍下随机种群比例参数:

随机种群的规模比例参数 r 的定义如下:

$$r = r_0 + \alpha * d \quad (1)$$

$$\text{其中, } \alpha = \begin{cases} 1, & \text{evaluation} \in [0, 3/\text{Maxev}) \\ 2, & \text{evaluation} \in [3/\text{Maxev}, 2 * \text{Maxev}/3), \\ 3, & \text{evaluation} \in [2 * \text{Maxev}/3, \text{Maxev}] \end{cases}$$

evaluation为当前评估次数,Maxev为最大评估次数; d 为参数调整步长。

3.2 算法的具体步骤

3.2.1 算法主体步骤

(1)初始化种群 $\text{pop}(t)$,计算个体适应度,将种群内个体按适应度降序排列。

(2)依据适应度将 $\text{pop}(t)$ 划分为精英种群和普通种群(M 个精英个体、 $N - M$ 个普通个体),并复制生成子代精英种群和普通种群;同时复制当前适应度最大的精英个体。

(3)精英组团:

①每个精英各自组建大小为 $G = 0.8 * (N - M) / M$ (G 向上取整)的团队。

②其中,从精英种群和普通种群中等概率地选取 $(1 - r) * G$ 个个体,然后随机生成 $r * G$ 个个体。

(4)进化操作:

①如果团队中的个体来自精英种群,进行协作进化:生成一个均匀分布的随机数 $u = U(0, 1)$;如果 $u < P_{\text{co}}$,执行CCOI算子,否则执行FCO算子。其中, $U(0, 1)$ 为在区间 $(0, 1)$ 内产生的一个均匀分布的随机数; P_{co} 为长方体交叉概率,是预先设定好的参数。

②如果来自普通种群,进行引导进化:执行CCOI算子。三个算子说明参考文献[5]。

③如果个体是随机生成的,则利用模拟退火机制进行进化,为了提高退火速度,参考文献[9],文中采用一种新的退火算法。过程后续介绍。

(5)更新子代精英和普通种群:

①对于协作和引导操作得到的新个体,如果适应度大,则由其代替父代个体;计算当前子代种群最差和第二差的个体,对于模拟退火机制进化得到的新个体,与当前子代种群中最差个体进行比较,如果其适应度大,则代替子代普通种群的最差个体。

②检验当前子代种群是否含有父代最优个体,没有则用父代最优个体替换子代种群第二差个体。

(6) M 次循环完毕,合并子代精英种群和普通种群为 $\text{pop}(t)$,令 $t = t + 1$ 。

(7)判断是否满足终止条件,满足则输出 $\text{pop}(t)$ 中的最优解;否则返回(2)。

3.2.2 具体退火步骤

上述步骤(4)中提到的模拟退火步骤如下:

Step1:令当前个体为初始解 S ,最优解 $\text{best} = S$,

$S(0) = S$,计数参数 $p = 0, i = 0$ 。

Step2: 令 $t = t_i$,进行抽样(参数为 t 、best、 $s(i)$) ; 抽样完成得到最优解 best₀ 和当前状态 s' 。同时令 $s(i + 1) = s'$ 。如果 $f(\text{best}) \leq f(\text{best}_0)$,则 $p = p + 1$;否则 $P = 0$,best = best₀。

Step3: 退火。 $t_{i+1} = e^{-\beta}t_i, i = i + 1$ 。 $\beta = \text{acceptn}/(\text{acceptn} + \text{max2})$ 。

Step4: 如果 $p \geq \text{max1}$ (阈值 1) ,则转向下一步; 否则返回 Step2。

Step5: 以 best 为最终的最优解输出, 结束整个退火算法。

其中, Step2 中的抽样过程如下:

①令初始状态 $s'(0) = s(i)$, 初始最优解 best₀ = best, 计数参数 $k = 0, q = 0$, acceptn = 0。

②由当前状态个体 $s'(k)$ 和当前主导精英进行单点交叉和非均匀变异, 产生新状态 s 。

计算 $\Delta f = f(s) - f(s'(k))$ 。如果 $\Delta f \leq 0$,则 $s'(k + 1) = s$, best₀ = s , $q = 0$, acceptn = accept + 1 ; 如果 $\Delta f > 0$,

令 $k = k + 1$, 如果以概率 $\exp(-\Delta f/t)$ 接受新状态, 令 $q = 0, s'(k + 1) = s$, acceptn = accept + 1 , 如果新状态没有被接受, 则 $s'(k + 1) = s'(k)$, $q = q + 1$ 。

③如果 $q \geq \text{max2}$ (阈值 2) , 则进入下一步, 否则返回上一步。

④令当前状态 $s' = s'(k)$, 令接受次数为 acceptn , best₀ 为最优解输出, 算法结束。

4 性能仿真实验

文中采用文献[5]中的 15 组无约束标准函数来验证算法的有效性, 以显示对比的公平性。同时, 函数的运行终止条件也为评估次数达到 30 万次后终止。

4.1 SACEA 算法运行结果

表 1 给出了 SACEA 对 15 组函数的运行结果, 参数设置如下: 种群大小 $N = 100$, 精英个体 $M = 20$; 长方体交叉概率 $P_{\text{cu}} = 0.3$; f_1 到 f_{13} 初始 T_0 取 500 , f_{14} 和 f_{15} 取 1 000 ; max₁ 取 200 , max₂ 取 300。计算随机种群比例参数 r 时 r_0 取 0.1 , d 取 0.1。

表 1 SACEA 算法对 15 个函数运行 50 次的仿真结果

函数	全局最优值	平均最优值	标准方差	最好值	最差值	平均评估次数
f_1	0	0	0	0	0	300 031
f_2	0	0	0	0	0	300 033
f_3	0	0	0	0	0	300 029
f_4	0	5.356×10^{-22}	2.386×10^{-22}	0	3.570×10^{-20}	300 031
f_5	0	7.383×10^{-56}	9.736×10^{-59}	0	4.850×10^{-53}	300 034
f_6	0	0	0	0	0	300 028
f_7	0	5.189×10^{-12}	4.063×10^{-12}	3.212×10^{-15}	4.852×10^{-8}	300 036
f_8	-12 569.5	-12 569.488 2	3.559×10^{-6}	-12 569.488 2	-12 569.488 2	300 033
f_9	0	0	0	0	0	300 032
f_{10}	0	0	0	0	0	300 037
f_{11}	0	0	0	0	0	300 029
f_{12}	0	2.107×10^{-33}	8.942×10^{-35}	2.107×10^{-33}	2.107×10^{-33}	300 036
f_{13}	0	0	0	0	0	300 030
f_{14}	-99.60	-99.524 785 6	1.890×10^{-2}	-99.569 380 2	-99.509 752 5	300 038
f_{15}	-78.332 36	-78.332 346 3	2.397×10^{-8}	-78.332 346 3	-78.332 346 3	300 032

从表 1 很容易可以看出, 有 10 个函数完全找到了全局最优值。对于 f_8 、 f_{12} 、 f_{15} , 将函数的理论最优个体代入函数中计算, 可以求得最优解; 这是因为计算机存在截断误差, 所以对于这三个函数, 也可以认定找到了它们的全局最优值。同时, 排除计算机截断的原因, 有 11 个函数的平均最优值达到了全局最优值, 算法具有相当的稳定性。

4.2 SACEA 和 MECA 比较

文中对 SACEA 算法和 MECA 算法各运行 50 次以后的实验结果对比如表 2 所示。从表 2 可以看出: 排

除计算机截断的影响, MECA 有 5 组函数找到了全局最优值, 而 SACEA 有 13 组; MECA 有 5 组平均最优值达到了全局最优值, 而 SACEA 有 11 组。在标准方差方面, MECA 算法有 5 组函数的标准方差在 10^{-10} 以上, 而 SACEA 算法只有 3 组。综上所述, SACEA 算法在总体的优化效果上明显优于 MECA 算法。

4.3 收敛性和收敛速度分析

4.3.1 收敛性分析

根据文献[14]的证明, 如果保留父代最优个体在子代种群中, 该算法就收敛, 一定可以收敛到全局的最

优解。文中的 SACEA 算法,每一代进化都把父代的最优个体保留到子代中,所以 SACEA 算法也是收敛的。

4.3.2 收敛速度分析

为了对比收敛速度,表 3 给出了以函数最优值达

到全局最优值为终止条件,算法各自运行 10 次,对 7 个函数的平均函数评估次数的对比。从表 3 可以看出,对于这 7 个函数,SACEA 的平均评估次数比 MECA 都有提高,SACEA 收敛速度优于 MECA。

表 2 SACEA 和 MECA50 次仿真结果对比

函数	全局最优值	平均最优值		标准方差		最优值	
		SACEA	MECA	SACEA	MECA	SACEA	MECA
f_1	0	0	4.228×10^{-183}	0	0	0	1.959×10^{-189}
f_2	0	0	1.845×10^{-110}	0	3.113×10^{-110}	0	6.866×10^{-114}
f_3	0	0	3.274×10^{-95}	0	2.313×10^{-94}	0	3.253×10^{-105}
f_4	0	5.356×10^{-22}	5.124×10^{-2}	2.386×10^{-22}	9.732×10^{-2}	0	8.167×10^{-10}
f_5	0	7.383×10^{-56}	7.973×10^{-2}	9.736×10^{-59}	5.638×10^{-1}	0	0
f_6	0	0	0	0	0	0	0
f_7	0	5.189×10^{-12}	4.083×10^{-4}	4.063×10^{-12}		3.212×10^{-15}	3.760×10^{-6}
f_8	-12 569.5	-12 569.488 2	-12 569.486 6	3.559×10^{-6}	7.350×10^{-12}	-12 569.488 2	-12 569.486 6
f_9	0	0	0	0	0	0	0
f_{10}	0	0	0	0	0	0	0
f_{11}	0	0	3.844×10^{-3}	0	7.130×10^{-3}	0	0
f_{12}	0	2.107×10^{-33}	1.571×10^{-32}	8.942×10^{-35}	5.529×10^{-48}	2.107×10^{-33}	1.571×10^{-32}
f_{13}	0	0	1.350×10^{-32}	0	1.106×10^{-47}	0	1.350×10^{-32}
f_{14}	-99.60	-99.524 785 6	-98.709 489 1	1.890×10^{-2}	1.450×10^{-1}	-99.569 380 2	-99.004 382 2
f_{15}	-78.33 236	-78.332 346 3	-78.332 331 4	2.397×10^{-8}	1.005×10^{-13}	-78.332 346 3	-78.332 331 4

表 3 两算法收敛所需平均次数比较

函数	全局最优值	平均函数评价次数	
		SACEA	MECA
f_6	0	4 822	6 852
f_8	-12 569.5	8 027	20 249
f_9	0	13 081	32 171
f_{10}	0	30 876	77 026
f_{12}	0	9 065	56 547
f_{13}	0	11 831	59 233
f_{15}	-78.332 6	18 743	46 937

表 4 给出了在相同的仿真环境下,以评估次数超过 300 000 次为结束条件,各算法各自运行 50 次的平均运行时间的对比图。从表 4 可以看出,达到相同的

表 4 两算法仿真 50 次运行时间的比较

函数	平均运行时间/s	
	SACEA	MECA
f_2	0.62	1.01
f_6	0.83	1.32
f_9	0.92	1.38
f_{11}	1.01	1.60
f_{14}	3.72	8.74
f_{15}	1.09	5.83

一定评估次数,SACEA 算法所需要的时间比 MECA 算法要少。这从另一方面证明了 SACEA 算法的收敛速度比 MECA 算法有所提高。

4.4 多样性分析

因为优化函数较多,所以文中选取 f_9 为代表函数来分析算法的多样性,如图 1 所示。从图中可知,在整个收敛过程中,算法很好地保持了种群的多样性,避免了早熟现象,增强了算法的全局寻优性能。

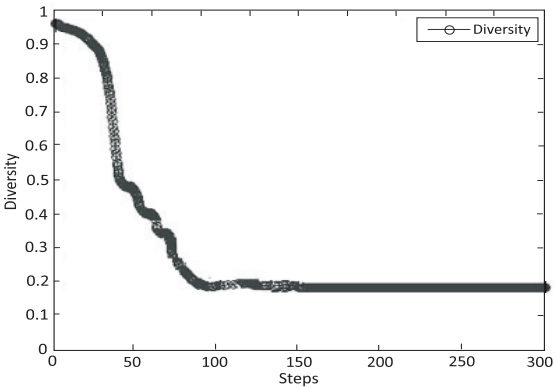


图 1 收敛过程中种群多样性

5 结束语

SACEA 算法的提出是结合了精英协同进化和模拟退火的精髓。算法的提出是为了获得更好的全局寻

优性能,并且获得相对快的收敛速度,所以算法将改进的快速模拟退火机制引入到精英协同进化中,并减少了精英操作算子;同时 SACEA 中随机种群的加入增加了种群的多样性,更好地防止了早熟收敛。文中通过对 15 组标准测试函数的仿真,并与 MECA 算法作对比,体现了 SACEA 算法在全局寻优性能、收敛性以及收敛速率方面的更好表现。

参考文献:

[1] Potter M A, de Jong K A. A cooperative coevolutionary approach to function optimization [C]//Proc of the parallel problem solving from nature. [s. l.] : [s. n.], 1994 : 249 – 257.

[2] Danoy G, Bouvry P, Martins T. hLCGA: a hybrid competitive coevolutionary genetic algorithm [C]//Proc of the 6th international conference on hybrid intelligent systems. Auckland: IEEE Computer Society Press, 2006: 48–51.

[3] 王 磊, 刘小勇. 协同人工免疫计算模型的研究 [J]. 电子学报, 2009, 37 (8) : 1739–1745.

[4] Meng W, Han X D, Hong B R. Bee evolutionary genetic algorithm [J]. Acta Electronica Sinica, 2006, 34 (7) : 1294–1300.

[5] Mu Caihong, Jiao Licheng, Liu Yi. M-Elite coevolutionary al-

gorithm for numerical optimization [J]. Journal of Software, 2009, 20 (11) : 2925–2938.

[6] 刘 全, 王晓燕, 傅启明, 等. 双精英协同进化遗传算法 [J]. 软件学报, 2012, 23 (4) : 765–775.

[7] 周昌乐, 游 维, 丁晓君. 一种宋词自动生成的遗传算法及其机器实现 [J]. 软件学报, 2010, 21 (3) : 427–437.

[8] 应伟勤, 李元香, Shen Philip CY, 等. 演化多目标优化中的几何热力学选择 [J]. 计算机学报, 2010, 33 (4) : 755–767.

[9] 朱颢东, 钟 勇. 一种改进的模拟退火算法 [J]. 计算机技术与发展, 2009, 19 (6) : 32–35.

[10] 王庆荣, 袁占亨, 张秋余. 基于改进遗传—模拟退火算法的公交排班优化研究 [J]. 计算机应用研究, 2012, 29 (7) : 2461–2463.

[11] 高 鹰, 谢胜利. 基于模拟退火的粒子群优化算法 [J]. 计算机工程与应用, 2004, 40 (1) : 47–50.

[12] Metropolis N, Rosenbluth A W, Rosenbluth M N, et al. Equation of state calculations by fast computing machines [J]. Journal of Chemical Physics, 1953, 21 (6) : 1087–1092.

[13] Kirkpatrick S, Gelatt J C D, Vecchi M P. Optimization by simulated annealing [J]. Science, 1983, 220 (4598) : 671–680.

[14] Rudolph G. Convergence analysis of canonical genetic algorithms [J]. IEEE Transactions on Neural Networks, 1994, 5 (1) : 96–101.

(上接第 90 页)

参考文献:

[1] Qian F S A. Application of fast and elitist non-dominated sorting generic algorithm in multi-objective reactive power optimization [J]. Transactions of China Electro Technical Society, 2007, 12: 24–24.

[2] Han Min, Fan Jianchao. Particle swarm optimization using dynamic neighborhood topology for large scale optimization [C]//Proc of 2010 8th world congress on intelligent control and automation. Jinan: IEEE, 2010, 3138–3142.

[3] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3) : 256–279.

[4] Jiao Jian, Wang Xianjia, Zhang Liubo. Applying particle swarm optimization in multiobjective optimization and hybrid optimization [C]//Proc of 2010 second international conference on computational intelligence and natural computing. [s. l.] : [s. n.], 2010.

[5] Zhang Jun, Chung H S H, Lo W L, et al. Pseudo co-evolutionary genetic algorithms for power electronic circuits optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2006, 36 (4) : 590 –

598.

[6] 李 宁, 邹 彤, 孙德宝, 等. 基于粒子群的多目标优化算法 [J]. 计算机工程与应用, 2005, 41 (23) : 43–46.

[7] 刘衍民, 牛 奔, 赵庆祯. 多目标优化问题的粒子群算法仿真研究 [J]. 计算机应用研究, 2011, 28 (2) : 458–460.

[8] 胡广浩, 毛志忠, 何大阔. 基于两阶段领导的多目标粒子群优化算法 [J]. 控制与决策, 2010, 25 (3) : 404–410.

[9] 刘衍民, 牛 奔, 赵庆祯. 基于交叉和变异的多目标粒子群算法 [J]. 计算机应用, 2011, 31 (1) : 82–84.

[10] 盛跃宾, 陈定昌, 穆 森, 等. 有等式约束优化问题的粒子群优化算法 [J]. 计算机工程与设计, 2006, 27 (13) : 2412–2413.

[11] 陈民铀, 张聪誉, 罗辞勇. 自适应进化多目标粒子群优化算法 [J]. 控制与决策, 2009, 24 (12) : 1851–1855.

[12] 陶新民, 徐 晶, 杨立标, 等. 一种改进的粒子群和 K 均值混合聚类算法 [J]. 电子与信息学报, 2010, 32 (1) : 92–97.

[13] 徐 刚, 杨玉群, 黄先玖. 一种非线性权重的自适应粒子群优化算法 [J]. 计算机工程与应用, 2010, 46 (35) : 49–51.

[14] Zitzler E, Ded K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results [J]. Evolutionary Computation, 2002, 8 (2) : 173–195.

基于模拟退火机制的精英协同进化算法

作者：[贺玫璐](#)，[罗杰](#)，[HE Mei-lu](#)，[LUO Jie](#)
作者单位：[南京邮电大学 自动化学院, 江苏 南京, 210046](#)
刊名：[计算机技术与发展](#)[ISTIC](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2015(1)

引用本文格式：[贺玫璐](#), [罗杰](#), [HE Mei-lu](#), [LUO Jie](#) [基于模拟退火机制的精英协同进化算法](#)[期刊论文]-[计算机技术与发展](#) 2015(1)