基于惩罚机制的自适应交叉粒子群算法

陈晋音 杨东勇 卢 瑾
(浙江工业大学信息工程学院 杭州 310023)

摘 要 粒子群算法存在容易陷入局部收敛的问题,尤其在求解约束条件优化问题时。提出一种基于惩罚机制的自适应交叉粒子群算法,其分3个层次克服局部收敛,获得最优解。首先引入交叉操作,根据粒子群进化过程中的种群多样性模型得到全局最优解。其次为求解约束优化问题,提出了基于惩罚机制的交叉粒子群算法,改进了H策略和简化了P策略惩罚机制。验证了所提算法在算法复杂度没有明显增加的情况下,性能得到了提高。最后分析得出在解决约束条件优化问题时,根据问题本身单峰和多峰的不同特性,粒子群算法的参数对收敛速度和最优解有关键影响。提出用通用公式计算参数,使算法得到最优解,从而推广粒子群算法的应用。

关键词 粒子群算法,交叉操作,收敛模型,自适应,单峰和多峰函数优化,约束优化

中图法分类号 TP18 **文献标识码** A

Self-adaptive Crossover Particle Swarm Optimization Based on Penalty Mechanism

CHEN Jin-yin YANG Dong-yong LU Jin
(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract Particle swarm optimization (PSO) has obvious shortcoming such as local convergence, whose performance of solving constrained optimization problems needs to be improved especially in aspect of convergence speed and optimum value. In this paper, penalty mechanism based self-adaptive crossover PSO was put forward to solve the above two problems by three levels. Aiming at the local convergence problem, crossover operation was brought into PSO. Population diversity model was used to maintain population diversity to achieve global optimum solution. Penalty mechanism based self-adaptive crossover PSO was put forward which improves H strategy and simplifies P strategy. The brought algorithm has better performance without obvious algorithm complexity increasing. According to velocity variety of particles during evolution, particle moving law and the impact of parameters were analyzed. A general calculating formula was put forward to control parameters for optimizing which extends application areas for PSO.

Keywords Particle swarm optimization, Crossover, Convergence model, Self-adaptive, Unimodal and multi-modal function optimizations, Constrained optimization

1 引言

粒子群算法(PSO: Particle swarm optimization)由Kennedy和Eberhart于1995年提出[1],与遗传算法等优化算法相比,PSO算法具有算法复杂度低、操作简单、收敛速度快等优势,但存在容易局部收敛、早熟等问题。动态约束粒子群算法应用于解决动态约束优化问题[2,3],利用粒子的随机搜索解空间能力可有效解决复杂动态约束下的优化问题,但收敛速度较慢。Yuhui Shi等提出的线性递减权重粒子群算法(Linear decreasing weight PSO)[4],利用权重控制收敛速度,收敛到全局最优解。基于混沌理论提出混沌粒子群算法(CPSO: Chaotic PSO)[5],利用随机策略来增加种群的多样性,从而克服PSO算法过早陷入收敛状态,但混沌操作时间复杂度高。利用领域拓扑改善PSO性能[6-8],提出静态邻域拓扑结构Gbest模型和Lbest模型,并基于模型采用可变多簇PSO算法[9],从拓扑设计上通过计算距离或概率对有效控制粒子群算法的收敛速度起了一定作用,但增加了额外的时间耗费。本文提出了一种自适应交叉粒子群算法,该算法在克服局部收敛的基础上,结合有效的惩罚机制解决约束优化。根据优化问题单峰和多峰的不同特性、参数对粒子群算法性能的影响以及相应的拓扑结构[11,12],通过协调全局最优和局部最优使得提出的算法可更好解决单峰和多峰优化问题,推广了基于惩罚机制的自适应交叉粒子群算法的应用领域。

2 自适应交叉粒子群算法

2.1 种群多样性模型

PSO算法在求解优化问题时,依靠记忆的功能来保持本种群最优解 *pbest* 和全局最优解 *gbest*,而粒子本身信息在每一代进化中全部更新,容易将部分粒子优势丢失,发展成为最优解的潜质被遗失,遗传算法利用交叉操作很好地解决了优

到稿日期:2009-06-02 返修日期:2009-08-24

陈晋音(1982—),女,博士生,主要研究领域为生物启发计算、入侵检测等,E-mail:chenjinyin@163.com;杨东勇(1961—),男,博士,教授,主要研究领域为生物计算、全方位视觉、多智能体系统。

秀染色体丢失的问题。因此仿照遗传算法[13]，利用种群多样性分析交叉操作对粒子群算法的影响，得出定理1。

定理1 交叉操作保持粒子群算法的种群多样性，可以克服早熟和局部收敛，获得最优解。

证明：在PSO算法中，特定模式从上一代通过进化保存在下一代的概率等于：

$$p(T_c(A,B)=Y)=\begin{cases}\dfrac{k\cdot p_c}{l}, & Y\neq A\\ (1-p_c)+\dfrac{k\cdot p_c}{l}, & Y=A\end{cases} \tag{1}$$

其中，T_c 表示单点交叉操作的交叉点，k 表示模式 Y 的起始位置，p_c 表示交叉概率，l 表示模式 Y 的长度。我们期望通过可控制的概率来保存模式 Y，使多样化种群避免陷入早熟或局部收敛，粒子群种群多样性定义如下。

$$diversity(s)=\frac{1}{|S|\cdot|L|}\cdot\sum_{i=1}^{|s|}\sqrt{\sum_{j=1}^{N}(p_{ij}-\overline{p_j})^2} \tag{2}$$

其中，s 表示整个种群，$|S|$ 代表种群大小，$|L|$ 表示解空间的对角线长度，N 表示需要优化的多峰函数的维数，p_{ij} 表示粒子 i 的第 j 维数值，也可以理解成粒子 i 第 j 维坐标位置。从式(2)可知，种群的多样性与种群大小和解空间大小相关。为克服局部收敛和早熟现象，需要最大化种群多样性，即需要求 $\sum_{i=1}^{|s|}\sqrt{\sum_{j=1}^{N}(p_{ij}-\overline{p_j})^2}$ 的最大值。根据柯西不等式：

$$(\sum_{i=1}^{n}a_ib_i)^2\leqslant\sum_{i=1}^{n}a_i^2\sum_{i=1}^{n}b_i^2 \tag{3}$$

不等式 $\sum_{i=1}^{|s|}\sqrt{\sum_{j=1}^{N}(p_{ij}-\overline{p_j})^2}\leqslant$

$$\sqrt{|s|\cdot(\sum_{j=1}^{N}(p_{1j}-\overline{p_j})^2+\sum_{j=1}^{N}(p_{2j}-\overline{p_j})^2+\cdots+\sum_{j=1}^{N}(p_{|s|j}-\overline{p_j})^2)}$$

取等号当且仅当 $\sum_{j=1}^{N}(p_{1j}-\overline{p_j})^2=\sum_{j=1}^{N}(p_{2j}-\overline{p_j})^2=\cdots=\sum_{j=1}^{N}(p_{|s|j}-\overline{p_j})^2$。假设一个粒子表示成 $[p_{i1}\ p_{i2}\cdots p_{id}]$，$n$ 表示粒子总数，$\sum_{j=1}^{N}(p_{1j}-\overline{p_j})^2=\sum_{j=1}^{N}(p_{2j}-\overline{p_j})^2=\cdots=\sum_{j=1}^{N}(p_{|s|j}-\overline{p_j})^2$，从而实现多样性 $diversity(s)$ 的最大化。而交换 p_{ij} 可以通过交叉操作实现，因此交叉粒子群算法在保证粒子群多样性时克服了早熟和局部收敛，从而得到全局最优解。

2.2 交叉概率

新一代粒子群中保留模式 Y 的概率和交叉概率有关，利用马尔科夫链来分析有限粒子群集合的粒子群算法[14]。假设粒子群算法可建模成一离散时间相关模型，利用有限状态组成的马尔科夫链类模型表示。

定理2 若交叉概率递减，则马尔科夫链中的状态趋于一个稳定值，粒子群算法收敛。

证明：有限状态组成的马尔科夫链类的 $Q(q)$：

$$q_{ij}(k)=q_{ij}^{(k-1,k)}=\Pr\{X^{(k)}=\phi_j\mid X^{(k-1)}=\phi_i\} \tag{4}$$

其中，$X^{(k)}$ 表示第 k 状态，从状态 $X^{(k-1)}$ 转换到状态 $X^{(k)}$ 的概率是 $\Pr\{X^{(k)}=\phi_j\mid X^{(k-1)}=\phi_i\}$。根据文献[14]的证明，遗传算法中转换函数与交叉概率相关，当 n 趋于无穷大时，状态趋于稳定，即 $\forall\varepsilon>0,\exists n,m>N,\lim(q_n-q_m)=0$，因此交叉概率递减时，马尔科夫链中的状态趋于一个稳定值，那么粒子群算法收敛。

根据定理1和定理2，将交叉操作引入PSO中，保持种群多样性，实现交叉粒子群算法。我们分别以实验分析4种递减交叉概率模型：

$$pc_1=pc_{high}-\frac{(pc_{high}-pc_{low})\times(G_{\max}-g)}{G_{\max}}+pc_{low} \tag{5}$$

$$pc_2=-(pc_{high}-pc_{low})\times(\frac{g}{G_{\max}})^2+pc_{high} \tag{6}$$

$$pc_3=(pc_{high}-pc_{low})\cdot(\frac{g}{G_{\max}})^2+(pc_{low}-pc_{high})\cdot(\frac{2\cdot g}{G_{\max}})+pc_{high} \tag{7}$$

$$pc_4=pc_{low}\times(\frac{pc_{high}}{pc_{low}})^{\frac{1}{(1+c\times\frac{g}{G_{\max}})}} \tag{8}$$

其中，$pc_{high}=0.9$，$pc_{low}=0.6$，$G_{\max}$ 表示最大进化代数，g 表示当前代数，$c=10$，c 是一个可调整参数。实验参数设置如下：惯性权重 $w=0.9$，常量 $c_1=c_2=2$，速度范围 $v\in[-1,1]$，粒子群种群大小 $size=200$，最大进化代数 $maximum\ iteration=2000$，维数 $dimension=10$，优化基准测试函数[4]。

2.3 函数优化

定理1和定理2证明自适应交叉粒子群算法(SCPSO: Self-adaptive Crossover PSO)通过保持种群多样性克服了局部收敛，将其最优解、收敛速度、时间复杂度与标准粒子群算法(SPSO)[1]、线性权重递减粒子群算法(LDWPSO)[5]、混沌粒子群算法(CPSO)[9]、综合学习粒子群算法(CLPSO)[11]优化Benchmark函数的进行比较，通过优化测试函数验证自适应交叉粒子群算法克服了局部收敛，同时得到的最优解比其他算法都好。如表1所列，SCPSO优化Sphere，Rosenbrok和Griewanks均获得了最好的最优解，与测试函数的理想最优解最接近；在优化Rastrigin函数时获得了最优解0，其他粒子群算法都没有。表2显示了PSO的时间复杂度比较。

表1 自适应交叉粒子群算法和其他改进的粒子群算法优化基准测试函数结果比较

Function	SPSO	LDWPSO	CPSO	CLPSO	Pc=0.05	Pc=0.5	Pc=0.9	SCPSO
Sphere	0.04	8.03E-28	4.19E-39	1.33E-29	5.04E-39	1.23032E-52	2.34E-54	1.45E-27
Rosenbrock	14.79	3.68	0.76	0.21	2.05	1.02706	2.30	0.10
Rastrigin	13.70	1.98	0.07	0	1.72	3.10	4.97	0
Griewanks	1.00E-5	8.65E-37	1.78E-37	2.27E-123	3.39E-36	1.92E-46	4.67E-45	1.32E-49

表2 交叉粒子群算法和其他粒子群算法的CPU时间消耗比较

Function	PSO	LDWPSO	CPSO	CLPSO	Pc=0.05	Pc=0.5	Pc=0.9	SCPSO
Sphere	2.072	2.1438	18.641	2.937	1.2094	1.9828	2.8312	2.9953
Rosenbrock	3.8782	3.9938	32.326	4.782	2.0828	3.3625	4.789	4.8612
Rastrigin	1.6247	1.6844	13.078	2.453	0.875	1.6688	2.3359	2.6588
Griewanks	1.7438	1.7594	9.5657	0.9578	1.8625	2.8328	3.85	9.5326

3 基于交叉粒子群算法的约束优化问题求解

对于求解约束条件下的优化问题，比较成熟的是利用惩罚机制的遗传算法来求解约束优化问题(CO:Constrained optimization)，本文在提出自适应交叉粒子群算法克服局部收敛的基础上，利用改进的H策略和简化的P策略惩罚机制来求解约束优化问题。在利用惩罚机制解决约束优化问题时，可能会因为惩罚机制中的参数限制而影响优化结果[15]，而文献[16]已经证明扰动策略可以改善粒子群在局部点的寻优能力。

3.1 惩罚机制

约条条件优化问题可以表示成非线性规划问题，最小化 $f(\vec{x})$，$\vec{x}\in S^n$，其中 S 是解空间全集，包括可行解区域和不可行解区域，优化需要满足的制约条件不等式 $E_i(\vec{x})=0, i=1,\cdots,m$ 和等式 $I_i(\vec{x})\leqslant 0, i=m+1,\cdots,p$。在某些特定制约条件下的优化问题，不等式制约条件可表示成 $E_i(\vec{x})-\varepsilon\leqslant 0, i=1,\cdots,m$，其中 ε 表示允许误差(很小的数)，可根据实际问题的精度要求决定。本文利用惩罚机制将制约条件下的优化问题转换成无制约条件的优化问题：

$$F(\vec{x})=\begin{cases} f(\vec{x}), & \vec{x}\in feasible\ \mathrm{Re}\ gion \\ f(\vec{x})+penalty(\vec{x}), & \vec{x}\notin feasible\ \mathrm{Re}\ gion \end{cases} \tag{9}$$

因此优化问题转换成了最小化函数 $f(\vec{x})\quad \vec{x}\in S^n$，研究的惩罚机制策略有H策略、J策略和P策略[15]。

(1) H策略及其改进

惩罚机制主要分为两类：固定的惩罚机制(Stationary)和不固定的惩罚机制(Non-stationary)，固定的惩罚机制在最小化的优化过程中利用固定的惩罚数值，而不固定的惩罚机制则惩罚数值是动态改变的。H策略是一种固定惩罚机制：

$$F(\vec{x})=f(\vec{x})+P(\vec{x}) \tag{10}$$

$$P(\vec{x})=\begin{cases} 0, & \vec{x}\in feasible\ \mathrm{Re}\ gion \\ \sum_{i=1}^{p} R_i H_i^2(\vec{x}), & \vec{x}\notin feasible\ \mathrm{Re}\ gion \end{cases} \tag{11}$$

其中，i 是等式制约和非等式制约的个数，$H_i=\max\{0, g_i(\vec{x})\}, i=1,2,\cdots,p$，$g_i(\vec{x})$ 是违反制约的大小。R 定义为各个制约条件的权重，根据实际而定。平衡各种不同制约条件对惩罚机制的影响，可利用 $g_i(\vec{x})$ 来实现。

$$g_i(\vec{x})=\begin{cases} |E_i(\vec{x})|, & \text{if } E_i(\vec{x})\neq 0 \\ I_i(\vec{x}), & \text{if } I_i(x)>0 \end{cases} \tag{12}$$

m 表示所有不等式和等式制约条件个数，约束条件 R_i 是一个分段函数，定义为各个制约条件的权重，利用 R_i 可以实现对不同制约条件的不同权重。本文提出将 R_i 的数值根据目标函数和惩罚机制的比例来决定最合适的取值。

$$Scale=\frac{f(\vec{x})}{\sum_{i=1}^{p} H_i(\vec{x})} \tag{13}$$

其中，f 表示目标函数的数值，$\sum_{i=1}^{p} H_i(\vec{x})$ 表示所有违反制约条件的总和，为保证最优解不违反任何一个制约条件，则需要满足 f 的数值比 $\sum_{i=1}^{p} H_i(\vec{x})$ 小得多，考虑 $Scale$ 等于 0.1，此时惩罚机制就能发挥其作用。

(2) J策略

J策略是一种动态、不固定的(non-stationary)惩罚机制，因此在粒子群算法进化过程中，惩罚机制的数值会变化，具体参考文献[15]。

(3) P策略及其简化

P策略提出的基本思想就是最差的可行解比最好的不可行解的适应度高，这是一种不固定的惩罚机制，在P策略中不仅需要考虑惩罚机制的数值，而且还需要将满足制约条件的最次解和不满足制约条件的最优解之间的距离进行比较，因此计算如下：

$$p(\vec{x})=r\sum_{j=1}^{p} f_j(\vec{x})+\rho(\vec{x},t)$$

$$\rho(\vec{x},t)=\max\{0,\max_{x\in F}\{f(\vec{x})\}-\min_{x\notin F}\{f(\vec{x})\}+\sum_{j=1}^{p} f_j(\vec{x})\} \tag{14}$$

其中，$\rho(\vec{x},t)$ 是一个时间函数。利用 $\max_{x\in F}\{f(\vec{x})\}-\min_{x\notin F}\{f(\vec{x})\}$ 和违反制约条件的比较来评价解 x 的适应度，这是一种动态的惩罚机制，可利用比较解违反制约条件的大小和其他解之间的距离来实现。本文提出的简化P策略可以在满足消耗较少计算时间的基础上，达到较好的优化效果。首先 $\rho(\vec{x},t)$ 的变化情况分为3种情况进行讨论。

(1) 假设随着 $\max_{x\in F}\{f(\vec{x})\}-\min_{x\notin F}\{f(\vec{x})\}$ 的递减 $\sum_{j=1}^{p} f_j(\vec{x})$ 没有改变，那么当前代的解 x 一定比前一代的解 x 优秀，因为 $\rho(\vec{x},t)$ 减少了。

(2) 假设随着 $\sum_{j=1}^{p} f_j(\vec{x})$ 的递减 $\max_{x\in F}\{f(\vec{x})\}-\min_{x\notin F}\{f(\vec{x})\}$ 没有变化，那么与情况(1)类似，因为 $\rho(\vec{x},t)$ 减少了，所以当前代的解 x 一定比前一代的解 x 优秀。

(3) 假设 $\max_{x\in F}\{f(\vec{x})\}-\min_{x\notin F}\{f(\vec{x})\}$ 和 $\sum_{j=1}^{p} f_j(\vec{x})$ 都随着时间递减，那么解 x 的惩罚机制 $p(\vec{x})=r\sum_{j=1}^{p} f_j(\vec{x})+\rho(\vec{x},t)$ 数值比前一代解 x 的惩罚机制数值小，因此情况(1)仍然成立。

总结以上3种情况，分析提出简化的计算方式：

$$p(\vec{x})=r\sum_{j=1}^{p} f_j(\vec{x})+\rho(\vec{x},t)$$

$$\rho(\vec{x},t)=\max\{0,\sum_{j=1}^{p} f_j(\vec{x})-\min_{x\notin F}\{\sum_{j=1}^{p} f_j(\vec{x})\}\} \tag{15}$$

其中，r 是常量，本文中设定 $r=2$。

3.2 实验分析

自适应交叉粒子群算法能有效克服局部收敛问题，将基于改进的惩罚机制的自适应交叉粒子群算法用于解决约束优化问题，通过对比优化测试函数与遗传算法，验证所提算法具有更好的优化性能。仿真实验采用各种不同的基准函数[15]，分别是基于遗传算法、改进H策略、J策略和简化P策略的粒子群算法优化基准测试函数，记录得到的最优解、平均值、最差解和标准差如表3所列。实验结果，主要是几个特征：

(1) 变量个数 D，根据 g 系列的测试函数，对于 $2<D<7$ 的函数，遗传算法和粒子群算法基本能稳定的优化，而 $D>7$ 的函数 $g01$ 和 $g07$，遗传算法的优化精度很低，与最优值 -15 和 24.3 相比差距很大，可见遗传算法利用惩罚机制处理多变量函数优化能力比较弱。而基于惩罚机制的SCPSO算法克服了这个问题，优化的精确度较高。

(2) 函数分为二次型、非线性和线性，从优化 $g01-g13$ 和 $f01-f06$ 来看，优化结果不受影响。因此，不管是遗传算法还是基于改进 H 策略、J 策略和简化 P 策略的 SCPSO 算法都能优化这 3 种类型制约条件下的目标函数。

(3) 制约条件的类型，等式和不等式。对于有等式制约条件的函数，例如 $g05$ 和 $g13$，由于等式在优化过程中采用 $|f(x)-f^*(x)|<1e-10$，在利用基于惩罚机制的 SCPSO 算法进行优化时，表现出比较好的收敛速度，但是对于遗传算法由于操作复杂，收敛速度比较慢，相比于不等式，等式在较多的制约条件下函数优化表现出一定的难度。

表 3　优化基准测试函数的遗传算法、改进 H 策略、J 策略和改进 P 策略粒子群算法优化结果比较

Algorithm	Index	g01	g05	g06	g07	g09	g10	g11	g13
GA	Best	−13.60	5231.37	−6340.27	25.68	685.36	7122.36	0.76	0.096
	Mean	−13.57	5255.41	−6337.47	25.69	702.11	7240.01	0.82	0.122
	Worst	−12.40	5280.24	−6333.25	28.12	713.41	7266.12	0.88	0.152
	Dev	3.11	20.33	10.24	5.33	33.52	56.77	0.13	0.23
H SCPSO	Best	−15.0	5126.51	−6961.81	24.36	680.63	7131.39	0.75	0.069
	Mean	−15.0	5316.59	−6961.81	24.72	680.63	7134.70	0.75	0.29
	Worst	−15.0	5965.49	−6961.81	25.02	680.63	7139.70	0.75	0.98
	Dev	0.0	52.46	0.0	0.64	0.0	10.21	0.0	0.72
J SCPSO	Best	−15.0	5126.59	−6961.81	24.69	680.63	7049.70	0.75	0.056
	Mean	−15.0	5133.78	−6961.81	25.11	680.63	7052.44	0.77	0.133
	Worst	−15.0	5140.02	−6961.81	25.32	680.63	7058.23	0.81	0.159
	Dev	0.52	10.32	0.0	1.40	0.0	9.58	0.08	0.16
P SCPSO	Best	−15.0	5226.23	−6961.81	24.32	680.63	7049.69	0.75	0.098
	Mean	−13.24	5250.06	−6961.81	24.37	688.25	7049.80	0.75	0.13
	Worst	−10.41	5310.45	−6961.81	24.44	697.12	7050.11	0.5	0.29
	Dev	4.01	31.27	0.0	0.12	8.03	0.61	0.0	0.21

Algorithm	Index	f01	f02	f03	f04	f05	f06
GA	Best	1.37	−6952.33	677.50	−30672.00	−31443.20	−213
	Mean	1.35	−6950.12	677.81	−30672.00	−31421.54	−213
	Worst	1.34	−6944.12	678.34	−30672.00	−31413.97	−213
	Dev	0.25	7.04	0.22	0.0	25.66	0.0
H SCPSO	Best	1.39	−6961.81	680.63	−30672.00	−31028.9	−213
	Mean	1.39	−6961.81	680.67	−30672.00	−31028.9	−213
	Worst	1.40	−6961.81	680.68	−30672.00	−31028.9	−213
	Dev	0.01	0.0	0.07	0.0	0.0	0.0
J SCPSO	Best	1.39	−6961.81	680.69	−30672.00	−31523.86	−312
	Mean	1.40	−6920.23	682.34	−30672.00	−31510.22	−312
	Worst	1.43	−6892.11	684.12	−30672.00	−31501.56	−312
	Dev	0.04	53.74	4.03	0.0	18.42	0.0
P SCPSO	Best	1.39	−6961.81	680.68	−30672.00	−31512.85	−312
	Mean	1.39	−6961.33	680.77	−30672.00	−31512.23	−312
	Worst	1.39	−6955.25	685.30	−30672.00	−31510.11	−312
	Dev	0.0	7.44	4.78	0.0	1.37	0.0

3.3　算法复杂度分析

除了比较最优解，还要分析基于惩罚机制 SCPSO 的算法复杂度，它包含 3 个参数：种群规模 M、变量个数 N、制约条件个数 S。针对改进的 H 策略，第 n 代迭代中算法计算的总次数 P_n 满足：

$$P_n \leqslant M+M(N+N+N)+\frac{1}{2}M(M-1)+S\times N=M+3MN+\frac{1}{2}M(M-1)+S\cdot N \tag{16}$$

由于参数在取值时 M 远远大于 N，因此可以推出基于改进 H 策略的时间复杂度最高级为 $O(M^2)$。针对 J 策略，惩罚机制和改进 H 策略的惩罚机制是同构的，所以复杂度是一样的，也是 $O(M^2)$。针对改进 P 策略，惩罚机制中要求可行解的最大适应度和不可行解的最小适应度两个，则需要遍历两次种群，所以第 n 代迭代中算法计算的总次数 P_n 满足：

$$P_n \leqslant M+M(N+N+N)+\frac{1}{2}M(M-1)+S\times N\times[\frac{1}{2}M(M-1)]$$

$$=M+3M\cdot N+\frac{1}{2}M(M-1)+\frac{1}{2}S\cdot N\cdot M(M-1)$$

$$=M+3M\cdot N+\frac{1}{2}(M+S\cdot N)(M-1) \tag{17}$$

所以简化 P 策略的 SCPSO 计算最高级为 $O(M^2+SNM)$，因为 SN 远小于 M，所以时间复杂度可以简化成 $O(M^2)$。因此基于改进 H 策略的 SCPSO 和基于简化 P 策略的 SCPSO 时间复杂度均为 $O(M^2)$，而遗传算法的算法复杂度为 $O(T\times M\times N)$[16]，本文提出的算法比遗传算法复杂度低。

3.4　仿真结果及收敛性分析

本文首先提出自适应交叉操作粒子群算法(SCPSO)，并实现基于惩罚机制的 SCPSO 用于求解约束优化问题，分析求解结果，如图 1—图 3 所示。

3 种策略下的 SCPSO 算法在求解 $g05$，$g09$ 和 $f2$，$f3$ 等非线性问题时，对可能存在的多极值优化问题收敛速度比较慢，但在求解 $g06$ 和 $g09$ 时收敛快，因此无法从图中分析得到 SCPSO 的收敛规律，为了深入了解 SCPSO 中粒子进化和算法收敛的关系，应推广基于惩罚机制的 SCPSO 来解决多种约束优化问题，在寻求全局最优解时加快收敛速度，针对优化问题本身特性与 SCPSO 粒子进化的关系做进一步的研究。

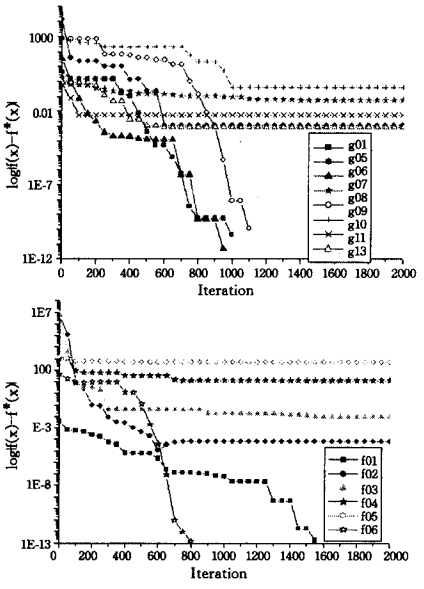

图 1　基于改进 H 策略的自适应交叉粒子群算法优化函数收敛图

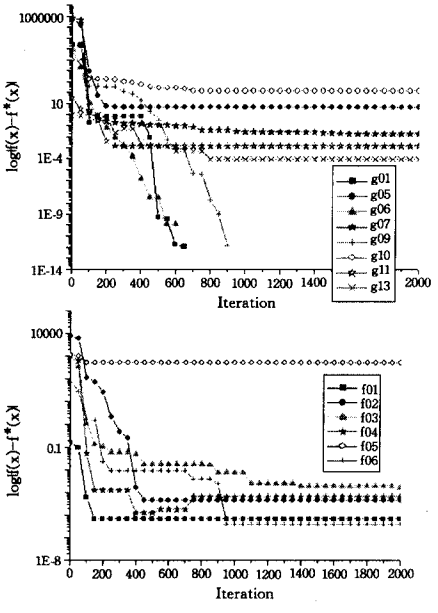

图 2　基于 J 策略的自适应交叉粒子群算法优化函数收敛图

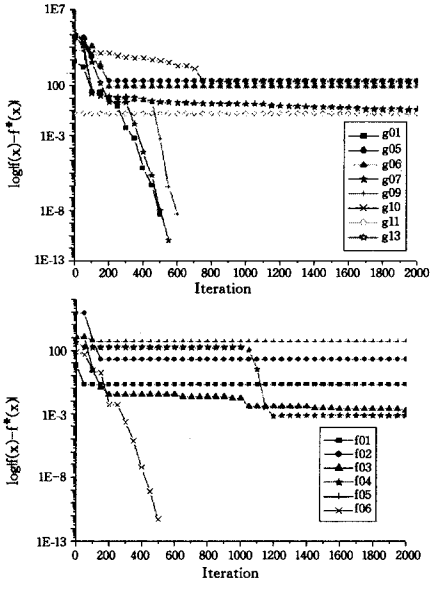

图 3　基于简化 P 策略自适应交叉粒子群算法优化函数收敛图

4　SCPSO 参数优化

4.1　粒子运动分析

首先粒子群算法中粒子的移动规律由当前位置和移动速度决定，假设求解问题是一维的，粒子的移动过程可以在 X 轴上表示，如图 4 所示。

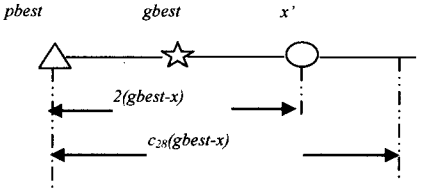

图 4　当 $x=pbest$ 时粒子的运动趋势和运动边界，其中三角形表示粒子历史最优位置，星形表示全局最优位置，圆形表示当前位置关于全局最优位置的对称点

(1) 当 $x=pbest$ 时，粒子的当前位置是 x，而 x 刚好是粒子历史记录中的最优，$pbest_i^k - X_i^k = 0$，即粒子历史最优位置不影响粒子位置的更新，$gbest^k - X_i^k \neq 0$。x' 是当前位置关于全局最优位置的对称位置，因此 x 与 x' 的距离等于 $2(gbest-x)$。对于速度更新公式，速度的更新范围是 $c_2(gbest-x)$，因此 $\frac{2(gbest-x)}{c_2(gbest-x)}$ 决定了粒子移动的范围，将其定义为参数 p 决定了粒子搜索的范围，利用参数 p 描述粒子的搜索能力。

$$p=\frac{2\times|gbest-x|}{c_2\times|gbest-x|}=\frac{2}{c_2} \tag{18}$$

(2) 当 $x\neq pbest$ 时，粒子的当前位置 x 不与粒子历史最优位置 $pbest$ 重合，$x-x'$ 表示当前位置 x 关于全局最优位置 $gbest$ 的对称位置。与情况(1)相比，粒子的移动范围发生了变化，因为当前位置不与历史最优位置重合，则在更新粒子位置时需要考虑 $pbest_i^k - X_i^k \neq 0$ 的影响，因此粒子经过更新以后的位置是 $c_1\times(pbest-x)+c_2\times(gbest-x)$，如图 5 所示，则此时有：

$$p=\frac{2\times|gbest-x|}{c_1\times|pbest-x|+c_2\times|gbest-x|} \tag{19}$$

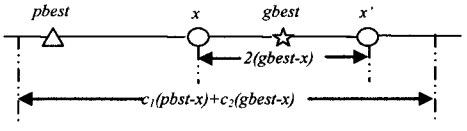

图 5　当 $x\neq pbest$ 时粒子的运动趋势和运动边界 $c_1(pbst-x)+c_2(gbest-x)$

总结以上(1)(2)，定义 p 描述粒子搜索能力，分子表示粒子到全局最优解的搜索范围，进化到最优解的理想搜索范围；分母表示粒子实际在进化过程中的搜索范围，p 等于两者的比，表示理想搜索范围与实际搜索范围比。

4.2　参数调整

根据 p 的定义，式(1)参数 c_1 和 c_2 通过以下公式计算。

$$\begin{cases} c_1=anyValue, c_2=\dfrac{2}{p}, & x=pbest \\ c_1=c_2\times\left|\dfrac{gbest-x}{pbest-x}\right|, c_2=\dfrac{1}{p}, & x\neq pbest \end{cases} \tag{20}$$

根据 Kenney 最早提出粒子群算法模仿鸟群的生物启发原理[1]：$pbest$ 代表粒子本身在搜索过程中找到的最优值，是一个私有因素(private part)，$gbest$ 是所有粒子搜索到的全局最优值，是一个社会因素(social part)。我们分析到：粒子如果对自身的信赖程度高，更有自信通过自身进化找到最优解，

则加大 $c_1(pbest-x)$ 的权重，如果信任社会全体共同得到的最优解，粒子更愿意向全局最优解靠近，则加大 $c_2(pbest-x)$ 的权重。描述粒子对 *pbest* 和 *gbest* 的信赖程度，引入 q 参数为粒子的自信因子。

$$\begin{cases} p=\dfrac{2\times|gbest-x|}{c_1\times|pbest-x|+c_2\times|gbest-x|} \\ q=\dfrac{c_1\times|pbest-x|}{c_2\times|gbest-x|} \end{cases} \tag{21}$$

其中，$x\neq pbest, x\neq gbest$，则 c_1 和 c_2 通过 p 和 q 计算：

$$\begin{cases} c_1=\dfrac{2q\times|gbest-x|}{p(q+1)\times|pbest-x|}, c_2=\dfrac{2}{p(q+1)}, & x\neq pbest \\ c_1=anyValue, c_2=\dfrac{2}{p(q+1)}, & x=pbest \end{cases} \tag{22}$$

其中，$p\in(0,1]$。SCPSO 调节 p 和 q 设置 c_1 和 c_2，实现优化单峰函数在找到的最优解收敛，而优化多峰函数放慢收敛速度，扩大搜索的范围找到全局最优，通过设置不同的 p 和 q，调节 c_1 和 c_2，实现对单峰和多峰优化问题的合理求解。Kennedy 提出 $c_1=c_2=2$[1]，其实是本文式(22)的一个特例，当 $\begin{cases} p=\dfrac{|gbest-x|}{1+|gbest-x|} \\ q=\dfrac{1}{|gbest-x|} \end{cases}$ 时，$c_1=c_2=2$，因此本文提出的利用 p 和 q 调节 c_1 和 c_2 是对粒子群算法的推广，为更多应用做准备。

表 4 利用 SCPSO 基于调节 p 和 q 来求解 Sphere 和 Rastrigin 函数，前者是单峰函数，设置较大 p 值可以快速获得最优解；后者是多峰函数，设置较小 p 值可以有效控制收敛速度获得最优解，最后一行是利用 Kennedy 的方法设置 $c_1=c_2=2$，最优解没有本文提出的调节 p 和 q 获得的最优解优秀。因此利用 SCPSO 通过调节 p 和 q 可以有效解决单峰和多峰优化问题。

结束语 本文提出用基于惩罚机制的自适应交叉粒子群算法求解约束条件优化问题，首先针对粒子群算法容易陷入局部收敛的问题，提出自适应交叉粒子群算法，定理证明和实验验证了算法在收敛速度和最优解比已有粒子群算法都好；其次，实现了基于惩罚机制的自适应交叉粒子群算法，改进了 H 策略，简化了 P 策略，结合提出的 SCPSO 有效优化了 Benchmark 函数。实验发现：优化单峰和多峰函数与 SCPSO 的收敛速度密切相关，因此应深入分析粒子的移动规律，定义参数 p 和 q 控制算法的收敛速度，最后通过实验验证了通过调节 p 和 q 能有效解决单峰和多峰优化问题，相比固定 c_1 和 c_2 为常数，所提算法能获得更好的最优解，推广了粒子群算法的应用。

表 4 调整 p 和 q 优化 Sphere 函数和 Rastrigin 函数，运行 10 次统计最优解、平均解和最差解

p	q	Sphere			Rastrigin		
		Best	Mean	Worst	Best	Mean	Worst
0.3	0.5	0.0150104	0.0643553	0.13411	0	0	0
	1	0.0428935	0.091605	0.137418	0	0	0
	2	0.0581127	0.111001	0.170852	1.251e−50	4.589e−40	5.654e-32
0.5	0.5	8.0029e-20	1.9814e-17	8.7702e-17	5.32907e-15	5.321e-13	9.532e-10
	1	1.1935e-19	2.6582e-18	7.8745e-18	3.2010e-15	4.2526e-12	4.216e-8
	2	3.9442e-11	1.7619e-10	2.8828e-10	5.86198e-13	1.6541e-10	5.684e-8
0.7	0.5	4.8446e-64	4.6573e-64	2.6844e-63	6.851e-10	2.8396e-8	9.12301e-6
	1	1.0261e-67	6.1296e-65	5.1076e-64	6.884e-6	6.556e-5	5.384e-4
	2	1.7199e-50	7.0436e-48	2.2163e-47	2.8396e-6	6.6912e-5	6.8970e-4
0.9	0.5	9.5936e-95	1.0566e-93	1.0322e-92	0.006251	0.0652	0.026591
	1	3.7872e-97	3.7594e-95	2.5115e-94	0.0021221	0.05710	0.06284
	2	3.3776e-80	2.3373e-76	2.2912e-75	0.010238	0.05841	0.08412
$p=\dfrac{\|gbest-x\|}{1+\|gbest-x\|}$	$q=\dfrac{1}{\|gbest-x\|}$	0.0431	0.1203	1.3255	1.98992	2.0421	2.2657

参 考 文 献

[1] Kennedy J, Eberhart R C. Particle swarm optimization [A]// Proc. IEEE International Conf. on Neural Networks[C]. Perth: IEEE Piscataway, 1995: 1942-1948

[2] Wimalajeewa T, Jayaweera S K. PSO for Constrained Optimization: Optimal Power Scheduling for Correlated Data Fusion in Wireless Sensor Networks [C]// Personal, Indoor and Mobile Radio Communications, PIMRC 2007. IEEE 18th International Symposium, 2007, 3(7): 1-5

[3] Liu Chun-an. New dynamic constrained optimization PSO algorithm [C]// Proceedings of the 2008 Fourth Internetional Conference on Natural Computation. 2008, 7: 650-653

[4] Shi Yuhui, Eberhart R. A modified particle swarm optimizer [A]// IEEE World Congress on Computational Intelligence[C]. Anchorage: IEEE, 1998: 69-73

[5] Chen Guimin, Jia Jianyuan, Han Qi. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm [J]. Journal of xi'an jiaotong university, 2006, 40(1): 53-61

[6] Mendes R. Population topologies and their influence in particle swarm performance [D]. Portugal: Escola de Engenharia: Universidade do Minho, 2004

[7] Kennedy J, Mendes R. Neighborhood topologies in fully in - formed and best-of-neighborhood particle swarms [J]. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2006, 36(4): 515-519

[8] Kennedy J, Mendes R. Population structure and particle swarm performance[C]// Proc. of the Congress on Computational Intelligence. Honolul: IEEE Press, 2002: 1671-1676

[9] Ni Q J, Zhang Z Z, Wang Z Z, et al. Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure [J]. Journal of Software, 2009, 20(2): 339-349

[10] Clerc M, Kennedy J. The particle swarm Explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Trans. on Evolutionary Computation, 2002, 6(1): 58-73

[11] Liang J J, Qin A K. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transaction on Evolutionary Computation, 2006, 10(3): 281-295

(下转第 284 页)

(a) (b) (c)

图 5 AWV-Shift 算法的人脸探测结果

实验结果表明，AWV-Shift 算法能够快速准确地探测出视频中的人脸目标，并且跟踪窗的尺寸能够随着人脸目标尺寸的变化自适应地进行调整。该算法在探测速度方面与目前广泛采用的 Adaboost 算法接近，并且在一定程度上降低了 Adaboost 算法的误判率。

结束语 本文将光照补偿和肤色探测作为修正人脸位置的后期手段。光照补偿和肤色探测由于仅在跟踪框内进行，因此能够以很小的计算成本提高传统 Mean-shift 算法的探测精度，在探测速度上也能够满足实时性要求。并且，本算法探测过程完全不需要人工参与，不像 OpenCV 中的连续自适应 Mean-shift 算法那样，在每次跟踪前还得手工标定人脸区域样本。

参 考 文 献

[1] Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003, 25(5): 564-577

[2] Yang C J, Duraiswami R, Davis L. Efficient spatial feature tracking via the Mean2Shift and a new similarity measure[A]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition[C]. San Diego, USA, 2005: 293-300

[3] Deguchi K, Kawanaka O, Okatani T. Object tracking by the Mean-Shift of regional color distribution combined with the particle-filter algorithm[A]// Proceedings of the 17 th International Conference on Pattern Recognition[C]. Cambridge, UK, 2004: 506-509

[4] 左军毅，梁彦，赵春晖，等. Mean Shift 跟踪算法中尺度自适应策略的研究[J]. 中国图象图形学，2008，13(9)：1750-1757

[5] Open Source Computer Vision Library , Intel Corporation [OL]. http://developer. intel. com

[6] Hsu R-L, Abdel-Mottaleb M, Jain A K. Face Detection In Color Images[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, 24(5): 696-706

(上接第 254 页)

[12] Zahara, Erwie, Hu, et al. Solving constrained optimization problems with hybrid particle swarm optimization [J]. Engineering Optimization, 2008, 40(11): 1031-1049

[13] Cao Y J, Wu Q H. Convergence analysis of adaptive genetic algorithm, genetic algorithms in engineering systems [J]. Innovations and applications, September 1997: 446-450

[14] Powell D, Skolnick M. Using genetic algorithms in engineering design optimization with non-linear constraints [C]// Proceedings of the fifth international conference on genetic algorithms. 1993: 270-271

[15] Kuri-Morales A F, Gutierrez-Garia J. Penalty function methods for constrained optimization with genetic algorithms: a statistical analysis [M]. Berlin Heidelberg: Springer-Verlag, 2002: 187-200

[16] Li Ai-guo. Particle swarms cooperative optimizer [J]. Journal of Fudan University: Natural Science, 2004, 43(5)

[17] 许艳. 基于改进遗传算法的自动组卷研究 [J]. 计算机与信息技术，2008，7(2)

(上接第 257 页)

[4] Fei Sheng-wei, Sun Yu. Forecasting dissolved gases content in power transformer oil based on support vector machine with genetic algorithm[J]. Electric Power Systems Research, 2008, 78 (3): 507-514

[5] Pai Ping-feng. System reliability forecasting by support vector machines with genetic algorithms[J]. Mathematical and Computer Modelling, 2006, 43(3): 262-274

[6] 张袅娜，张德江，冯勇. 基于混沌遗传算法的柔性机械手滑模控制器优化设计[J]. 控制理论与应用，2008，25(3)：451-455

[7] Schölkopf B, Smola A. Learning with kernels: support vector machines, regularization, and beyond [R]. Cambridge, MA: MIT Press, 2002

[8] Wang Wenjian, Xia Zongben, Lu Weizhen, et al. Determination of the spread parameter in the Gaussian kernel for classification and regression[J]. Neurocomputing, 2003, 55(3): 643-663

[9] 杜京义，侯媛彬. 基于遗传算法的支持向量回归机参数选取[J]. 系统工程与电子技术，2006，28(9)：1430-1433

[10] 姚俊峰，梅炽，彭小奇，等. 混沌遗传算法及其应用[J]. 系统工程，2001，19(1)：70-74

[11] Tavazoei M S, Haeri M. Comparison of different one - dimensional maps as chaotic search pattern in chaos optimization algorithms [J]. Applied Mathematics and Computation, 2007, 187(2): 1076-1085

[12] Yang Dixiong, Li Gang, Cheng Gengdong. On the efficiency of chaos optimization algorithms for global optimization [J]. Chaos, Solitons & Fractals, 2007, 34(4): 1366-1375

(上接第 260 页)

参 考 文 献

[1] Samad T, Bay J S, Godbole D. Network-Centric Systems for Military Operations in Urban Terrain: The Role of UAVs[J]. Proceedings of the IEEE, 2007, 95(1): 92-107

[2] Nolle L, Wong K C P, Hopgood A A. DARBS: A Distributed Blackboard System[M]// Bramer, Coenen, Preece, eds. Research and Development in Intelligent Systems XVIII. Springer, 2001: 161-170

[3] A Hovercraft Testbed for Decentralized and Cooperative Control [C]// Proceeding of the 2004 American Control Conference. Boston, Massachusetts, 2004

[4] Popp R. MTE Ground Station Testbed - A Battlefield Awareness Asset for GMTI Exploitation

[5] Shields J F. The Formation Control Testbed Celestial Sensor: Overview, Modelling, and Calibrated Performance

[6] King E, Kuwata Y, Alighanbari M, et al. Coordination and Control Experiments on a Multi-vehicle Testbed

[7] McLain T W, Randal W. Beard Unmanned Air Vehicle Testbed for Cooperative Control Experiments

[8] Niland W M. The Migration of a Collaborative UAV Testbed into the Flames Simulation Environment[C]// Proceedings of the 2006 Winter Simulation Conference

[9] Wise R, Rysdyk R, Seattle, et al. Multi-vehicle Cooperative Control Flight Test[C]// Anawat Pongpunwattana, MO. IEEE 25th Digital Avionics Systems Conference. October 2006

[10] Hoppe H. Terrain Rendering Using GPU-Based Geometry Clipmaps Arual Asirvatham, Microsoft Research, 2004