

# 基于云模型的进化算法

张光卫<sup>1)</sup> 何 锐<sup>1)</sup> 刘 禹<sup>1)</sup> 李德毅<sup>2)</sup> 陈桂生<sup>2)</sup>

<sup>1)</sup>(北京航空航天大学计算机学院 北京 100083)

<sup>2)</sup>(中国电子系统工程研究所 北京 100840)

**摘 要** 基于云模型在非规范知识的定性、定量表示及其相互转换过程中的优良特征,结合进化计算的基本思想,提出一种基于云模型的进化算法.该算法利用云模型对物种的遗传变异进化统一建模,能够自适应控制遗传变异的程度和搜索空间的范围,从而可以快速收敛到最优解,较好地避免了传统遗传算法易陷入局部最优解和选择压力过大造成的早熟收敛等问题.仿真结果表明:该算法具有精度高、收敛速度快等优点.云模型和进化计算思想的有效结合一方面拓宽了云模型的应用领域,也为进化计算的研究进行了新的探索和尝试.

**关键词** 云模型;进化计算;遗传算法;数值优化;人工智能

中图法分类号 TP18

## An Evolutionary Algorithm Based on Cloud Model

ZHANG Guang-Wei<sup>1)</sup> HE Rui<sup>1)</sup> LIU Yu<sup>1)</sup> LI De-Yi<sup>2)</sup> CHEN Gui-Sheng<sup>2)</sup>

<sup>1)</sup>(School of Computer Science and Engineering, Beihang University, Beijing 100083)

<sup>2)</sup>(Institute of Electronic System Engineering of China, Beijing 100840)

**Abstract** Based on the outstanding characteristics of the cloud model on the process of transforming a qualitative concept to a set of quantitative numerical values, and integrating with the basic principle of evolutionary computation, the authors propose a novel rapid evolutionary algorithm, namely Cloud Model Based Evolutionary Algorithm or CBEA for short. With the cloud model, inheritance and mutation of species can be modeled naturally and uniformly, which make it easy and nature to control the scale of mutation and inheritance, and the scope of searching space. This enables CBEA to be able to find accurate numerical solutions within a short time. Numerical optimization experiments are carried out to verify the algorithm. With some of the typical test functions, the performance of CBEA is studies and also be compared with other algorithms. The results prove the high quality of the algorithm on precision, stability and convergence rate. In addition, the successful integrating of the cloud model and evolution principle expands the research fields of Cloud Theory and also indicates a new way for the research of evolutionary computation.

**Keywords** cloud model; evolutionary computation; genetic algorithm; numerical optimization; artificial intelligence

收稿日期:2006-04-24;最终修改稿收到日期:2007-07-12. 本课题得到国家自然科学基金(60496323,60375016)及国家“九七三”重点基础研究发展规划项目“现代设计大型应用软件的共性基础”(2004CB719401)资助. 张光卫,男,1970年生,博士研究生,研究方向为人工智能. E-mail: ezhang@263.net. 何 锐,男,1976年生,博士,讲师,研究方向为进化计算、移动计算. 刘 禹,男,1980年生,博士研究生,研究方向为人工智能. 李德毅,男,1944年生,博士生导师,中国工程院院士,主要研究领域为人工智能、指挥自动化. 陈桂生,男,1965年生,博士,讲师,研究方向人工智能、复杂网络、指挥自动化.

# 1 引言

遗传算法(GA)<sup>[1-5]①</sup>是智能计算的重要分支. 问题解按照一定编码方案被表示成遗传空间的串结构个体, 用适应度函数进行评价以选择出优秀个体, 通过交换和变异操作产生新一代更适应环境的种群. 系统初始化为一组随机解, 经过逐代演化, 收敛到一个最适应环境的串上, 求得问题的最优解.

云模型是李德毅院士提出的一种定性知识描述和定性概念与其定量数值表示之间的不确定性转换模型<sup>[6-9]</sup>, 已经在智能控制、模糊评测等多个领域得到应用. 正态云模型是最重要的一种云模型, 由于其具有良好的数学性质, 可以表示自然科学、社会科学中大量的不确定现象<sup>[7]</sup>.

云模型在知识表达时具有不确定中带有确定性、稳定之中又有变化的特点, 体现了自然界物种进化的基本原理. 对于云模型  $C(Ex, En, He)$ ,  $Ex$  可以代表父代个体遗传的优良特征, 是子代对父代的继承,  $En$  和  $He$  可以表示继承过程的不确定性和模糊性, 表现了物种进化过程中的变异特征. 用正向云算子可以完成概念空间到数值空间的转换, 在 3 个参数的控制下产生子代种群, 完成遗传操作. 一方面这种转换是确定的和精确的, 因为数值空间的每一个云滴都是定性概念的一次量化实现, 都在一定程度上是该定性概念的代表; 另一方面, 这种转换又是随机的和模糊的, 每一次变换得到不同的云滴集合, 而且同一定性概念可用云滴集合中的任何一个代表, 不同的云滴代表该概念的确定程度不同.

基于云模型的优良特性, 结合进化计算的基本原理, 本文提出一种基于云模型的进化算法, 该算法不但比传统遗传算法精度高, 而且能够很好地避免遗传算法易陷入局部最优解和选择压力过大造成的早熟收敛等问题.

本文首先对云模型进行简单介绍; 然后重点给出了 CBEA 的思想和详细算法流程; 仿真实验部分对本文提出的算法进行测试和对比分析; 最后总结全文并给出下一步的研究方向.

# 2 云模型简介

**定义 1.** 云和云滴. 设  $U$  是一个用数值表示的定量论域,  $C$  是  $U$  上的定性概念, 若定量值  $x \in U$  是定性概念  $C$  的一次随机实现,  $x$  对  $C$  的确定度

$\mu(x) \in [0, 1]$  是有稳定倾向的随机数:  $\mu: U \rightarrow [0, 1]$ ,  $\forall x \in U, x \rightarrow \mu(x)$ , 则  $x$  在论域  $U$  上的分布称为云 (Cloud), 记为云  $C(X)$ . 每一个  $x$  称为一个云滴<sup>[6]</sup>. 如果概念对应的论域是  $n$  维空间, 那么可以推广至  $n$  维云.

云模型所表达概念的整体特性可以用云的数字特征来反映, 云用期望  $Ex$  (Expected value)、熵  $En$  (Entropy)、超熵  $He$  (Hyper entropy) 这 3 个数字特征来整体表征一个概念, 多维云模型的整体特征可由多组数字特征表示. 期望  $Ex$  是云滴在论域空间分布的期望, 是最能够代表定性概念的点, 或者说是这个概念量化的最典型样本; 熵  $En$  代表定性概念的可度量粒度, 通常熵越大概念越宏观, 它也是定性概念不确定性的度量, 由概念的随机性和模糊性共同决定. 一方面  $En$  是定性概念随机性的度量, 反映了能够代表这个定性概念的云滴的离散程度; 另一方面又是定性概念亦此亦彼性的度量, 反映了在论域空间可被概念接受的云滴的取值范围; 超熵  $He$  (Hyper entropy) 是熵的不确定性度量, 即熵的熵, 由熵的随机性和模糊性共同决定. 用 3 个数字特征表示的定性概念的整体特征记作  $C(Ex, En, He)$ .

**定义 2.** 一维正态云算子  $Ar^{\text{Forward}}(C(Ex, En, He))$  是一个把定性概念的整体特征变换为定量表示的映射  $\pi: C \rightarrow \Pi$ , 满足以下条件:

- (1)  $\Theta = \{t_i \mid \text{Norm}(En, He) \text{ 的一次实现, } i = 1..N\}$ ;
- (2)  $X = \{x_i \mid x_i \text{ 为 } \text{Norm}(Ex, t_i) \text{ 的一次实现, } t_i \in \Theta, i = 1..N\}$ ;
- (3)  $\Pi = \{(x_i, y_i) \mid x_i \in X, t_i \in \Theta, y_i = e^{-(x_i - Ex)^2 / (2t_i^2)}\}$ .

其中  $\text{Norm}(\mu, \delta)$  为期望为  $\mu$  方差为  $\delta$  的正态随机变量,  $N$  为云滴的个数.

利用正态云算子, 就可以把定性概念  $C(Ex, En, He)$  变换为数值表示的云滴集合, 实现了概念空间到数值空间的转换. 一维正态云算子可以推广至  $n$  维正态云算子.

# 3 基于云模型的进化算法(CBEA)

通过正向云算子根据 3 个数字特征产生云滴的过程能够刻画遗传和变异的思想, 以一维正态云模

① Heirkotter J, Beasley D. The hitch-hiker's guide to evolutionary computation. FAQ in Comp Ai Genetic, 1995. <http://www.cse.dmu.ac.uk/~rij/gafaq/top.htm>

型  $C(0, En, He)$  为例, 它表达的概念为“坐标轴原点附近”, 以不同的熵和超熵可以把这个定性概念转化为定量表示, 如图 1 所示分别做出通过正向云算子得到的 500 个云滴的图形. 图形上的每个云滴可以理解以 0 (期望) 为种子个体, 以不同的熵和超熵得到的后代个体. 云滴形成的每个云相当于一个种群, 可以看到 4 个种群在覆盖范围和离散程度上的明显差异, 即熵越大, 个体的覆盖范围越大, 超熵越大, 种群中的个体越离散.

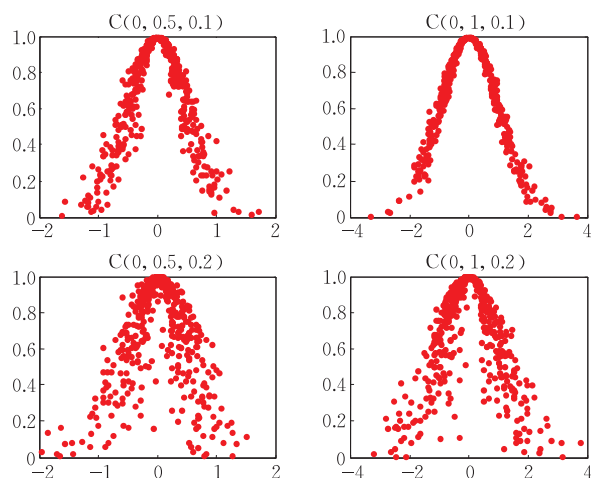


图 1 用云表达概念“坐标轴原点附近”

可见期望体现了遗传的稳定性, 熵体现了变异的范围, 可以表示搜索的广度, 超熵体现了进化的稳定程度, 可以表示求精的粒度. 根据人类搜索的经验, “当前优秀个体周围往往存在更加优秀的个体”, 制定下面的搜索过程: 首先, 求出优秀个体, 对一个进化代所有的个体 (云滴) 根据被优化的函数进行适应度计算 (fitness value), 根据适应度选择得到前  $m$  个最适应的优秀个体, 称为优秀个体向量; 其次, 以优秀个体为母体繁殖下一代个体, 即分别以  $m$  个个体为种子产生一定数量的云滴作为下一代个体, 给定规则使得越优秀的种子产生的下一代个体数量越多, 这样得到新一代的个体中存在更加优秀的个体的可能性较大, 反复叠代便可逐步寻优; 另外, 根据历史进化代的情况可以对下一个进化代熵和超熵的值进行自适应控制 (放大或缩小), 以控制搜索的范围和求精的粒度. 这个控制过程一方面要防止算法陷入局部最优和早熟收敛, 另一方面还要满足在可能存在最优解的范围内迅速求精, 当然这两个方面是矛盾的, 需要制定合适的控制策略.

在这个过程中, 遗传和变异是统一在一起的, 都融合在新个体的产生过程中, 因为由正向云算子产生的子代个体 (云滴) 都会聚集在母体 (云期望) 周

围, 充分体现了“龙生龙, 凤生凤”的遗传特征, 同时这也是一个充满不确定性的过程, 是随机的和模糊的, 每一次变换得到不同的云滴集合, 体现了“一母生九子九子各不同”的变异特征. 当算法在若干进化代都不能产生新的优于历史个体的最优个体时, 就要考虑突变操作, 生物学认为突变是能够影响生物遗传性状的变异, 但它并不经常发生, 本算法中最深度的突变操作是系统重新进行初始化.

以上面的思想为基础, 本文提出一种新颖的演化算法——CBEA (Cloud Based Evolutionary Algorithm). 类似大多数演化计算技术遵循的过程, CBEA 的计算过程为

1. 系统初始化为一组随机解, 即随机初始化群落中个体的值.
2. 计算所有个体的适应度, 并选择出适应度最好的前  $m$  个最优秀个体, 形成优秀个体向量.
3. 前  $m$  个个体分别繁殖一个种群.
4. 如果达到演化代数则算法停止, 最优秀的个体即为最优解, 否则转步 2.

CBEA 与粒子群算法<sup>[10-11]</sup> (Particle Swarm Optimization, PSO) 和遗传算法有很多共同之处, 都是基于叠代的优化过程, 都随机初始化种群, 而且都使用适应值来评价系统, 进而根据适应值来进行一定的随机搜索, 它们都不能保证一定能找到最优解.

CBEA 与遗传算法的不同之处在于, 虽然也是通过叠代搜寻最优值, 但没有遗传算法的二进制编码工作, 也没有相应的交叉 (crossover)、变异 (mutation) 等操作, 而是通过正向云算子完成新一代种群的产生, 通过熵、超熵控制产生子代种群的位置、范围 (搜索的范围) 以及子代种群集聚程度 (求精的粒度), 算法的实现也较遗传算法简单.

CBEA 和 PSO 都采用实数编码, 都没有相应的交叉、变异操作, PSO 根据粒子的速度来决定搜索, 粒子们追随当前的最优粒子在当前解空间中搜索, 而 CBEA 在各个进化代是个体不断产生和被淘汰的过程, 既体现了进化论的思想, 也体现了人类搜索的特点.

假设参加叠代寻优的个体总数为  $n$ , 称为一个群落  $C$ , 群落中的个体被分成  $m$  个种群,  $m$  也称为群落的丰富度. 群落和种群的关系为  $C = \bigcup_{i=1}^m P_i$ , 其中  $P_i$  表示第  $i$  个种群, 各个种群包括的个体的数目可以不同, 个体被编码为一个  $k$  元组,  $x_{ij}^k$  表示第  $i$  个种群的第  $j$  个个体的第  $k$  个分量, 由于个体能且只能属于一个种群, 所以  $n = j_1 + j_2 + \dots + j_m$ , 种群是个

体的集合,表示为

$$\begin{aligned} P_1 &= \{ (x_{11}^1, x_{11}^2, \dots, x_{11}^k), (x_{12}^1, x_{12}^2, \dots, x_{12}^k), \dots, \\ &\quad (x_{1j_1}^1, x_{1j_1}^2, \dots, x_{1j_1}^k) \}, \\ P_2 &= \{ (x_{21}^1, x_{21}^2, \dots, x_{21}^k), (x_{22}^1, x_{22}^2, \dots, x_{22}^k), \dots, \\ &\quad (x_{2j_2}^1, x_{2j_2}^2, \dots, x_{2j_2}^k) \}, \\ &\dots \\ P_m &= \{ (x_{m1}^1, x_{m1}^2, \dots, x_{m1}^k), (x_{m2}^1, x_{m2}^2, \dots, x_{m2}^k), \dots, \\ &\quad (x_{mj_m}^1, x_{mj_m}^2, \dots, x_{mj_m}^k) \}. \end{aligned}$$

按照群落中种群的规模对种群的地位进行划分,分为建群种、优势种和劣势种,其中优势种是拥有较多个体的种群,建群种为优势种中的最优者. 一个群落中建群种只有一个,是群落中的最具代表性的种群,优势种和劣势种则可以有多多个,其中劣势种规模最小但个数最多. 根据当前最优个体周围存在更加最优个体的概率较大的原则,越优秀的个体将产生更多的子代个体. 在每一个进化代,挑选适应度最好的前  $m$  个最优个体,  $m$  为群落中种群的个数,淘汰群落中所有其余个体,以  $m$  个最优个体为母体产生  $m$  个种群,构成新一代群落,最优秀的个体产生建群种,依次类推使得越优秀的个体产生的种群规模越大.

比如假设群落  $C$  的个体总数  $n=1000$ ,种群数  $m=10$ ,种群大小依次为  $[300, 200, 150, 50, 50, 50, 50, 50, 50, 50]$ ,在每个进化代的优秀个体向量大小为 10,算法在每一个进化代根据个体的适应度挑选出 10 个优秀个体,下一代的种群由这 10 个优秀个体产生,其中最优秀的个体将作为母体产生建群种,其它次优秀的个体将依次产生优势种和劣势种群,则种群结构与优秀个体的关系如图 2 所示.

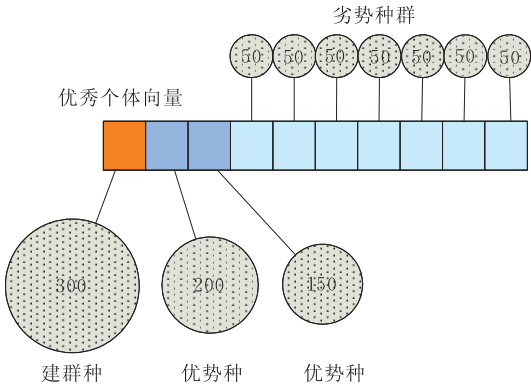


图 2 种群结构与优秀个体的关系

基于云模型的进化算法不像遗传算法那样注重父代与子代的遗传细节(通过基因表达遗传操作)上的联系,而是通过云模型把进化和遗传作为一种定性知识进行描述,进而通过不确定的定性定量转换,

把定性知识转化成若干定量实现(子代个体). 在每一个进化代,一方面通过对子代个体的适应性评估得到优秀个体,另一方面根据历史进化情况确定下一步进化的范围,从而形成新的知识,为下一步进化做好了准备. 为了便于表述算法思想首先定义以下基本概念.

**定义 3.** 进化模型(evolution pattern).  $EP(Ex, En, He)$ 是用云模型表达的进化模型. 其中  $Ex$  称为种子个体,表达祖先遗传的优良特性; $En$  称为进化熵,代表变异的大概范围; $He$  称为进化超熵表示进化的稳定性,  $He$  越大则不确定性越强. 进化模型包括一维模型和多维模型,多维模型用  $EP(Ex_1, En_1, He_1, \dots, Ex_n, En_n, He_n)$ 表示,其中维数  $n$  为大于等于 1 的整数. 给定父代个体( $Ex$ )作为母体,指定熵和超熵( $En, He$ ),利用正向云算子便可以产生任意数量的云滴,所有云滴均是该母体的后代个体,形成一个种群,因而进化模型可以看成是种群的产生模型.

**定义 4.** 进化(evolution). 它是指以父代种群中适应度较好的优秀个体为母体,按照进化模型产生新种群的操作. 进化操作包含有一定的变异成分,因为个体既继承了母体的优良特征,又与母体有一定程度的不同,由于这种变异为稳中求变,故称为进化式变异. 进化式变异的变化程度主要受进化模型的参数  $En$  控制,其变化程度可能较小( $En$  较小时),也可能较大( $En$  较大时).

**定义 5.** 突变(mutation). 它是指进化过程中全部或部分抛弃父代种群的优秀个体,并按照一定策略生成新的个体作为母体产生新种群的操作. 经过突变产生的群落相对于父代群落差异较大.

**定义 6.** 进化代(evolution generation). 进化过程中新群落的一次产生称为一个进化代.

**定义 7.** 适应度(fitness). 它是指群落中各个个体对环境的适应程度,用于衡量个体优劣的程度. 根据适应度的大小,可以决定某个个体是被保留还是被淘汰. 适应度通常是费用、盈利、方差等目标的表达式,适应度函数的构造可以参考遗传算法中适应度评估函数的构造方法.

**定义 8.** 精英个体(elite individual). 它是指进化过程中得到的适应能力最强的个体,分为当代精英和跨代精英. 当代精英指一个进化代的所有个体中适应性最强的个体,跨代精英指多个进化代中适应性最强的个体,进化过程(算法)的最终结果即为所有进化代中的最优跨代精英个体. 出现跨代精英的进化代称为非平凡进化代,没有出现跨代精英的



进化代称为平凡进化代,两个跨代精英个体之间相隔的进化代数称为连续平凡代数,是连续没有出现跨代精英个体的进化代数.连续出现跨代精英的进化代数称为连续非平凡代数.

连续平凡代数和连续非平凡代数是进化过程中进行自适应调节的重要数据.较大的连续平凡代数说明目前搜索的邻域中难以发现更加优秀的个体,那么此时算法可以自适应地采取突变操作.连续非平凡代的出现则表明当前进化代是有效的,算法可以自适应地控制进化过程中的变异程度(求精操作).

**定义 9.** 进化策略(evolutionary strategy).它是指进化过程中进化操作的控制策略,亦即通过调整进化模型的参数  $E_n$  和  $H_e$  来优化子代种群产生的策略.通过制定进化策略可以解决两方面问题:(1)局部求精.当出现了跨代精英个体时,算法可能找到了新的极值邻域,或更加逼近了老的极值邻域,此时需要求精操作,方法是降低进化范围(减小  $E_n$ )和增加稳定性(减小  $H_e$ ),从而加大搜索的精度和稳定度以达到快速局部求精的目的,比如可以简单地把  $E_n$  和  $H_e$  减小为原来的  $1/K$ ,其中  $K$  为大于 1 的实数,称为进化系数.(2)局部求变(进化式变异).当若干进化代没有发现新的跨代精英,即连续平凡代数达到一定的阈值  $\lambda_{local}$  时,算法可能陷入了一个局部最优邻域,此时需要跳出这个小局部,并在该局部附近尝试寻找新的局部最优.方法是提高  $E_n$  和  $H_e$ ,比如简单地提高为原来的  $L$  倍, $L$  称为进化式变异系数, $L \leq K$ ,可取  $L = \lceil \sqrt{K} \rceil$ .当函数的局部最优值非常邻近时,进化式变异可以在众多邻近局部最优值中寻找全局最优.

**定义 10.** 变异策略(mutation strategy).它是指进化过程中对突变操作的控制策略,是算法摆脱局部的保证.当经过若干代进化没有得到适应性更加优异的个体,而且进化式变异没有效果时,算法有可能陷入局部,需要进行一次突变操作.进行局部求变和突变的连续平凡代数阈值之间的关系为  $\lambda_{global} > \lambda_{local}$ .突变方法有两种,一种是取历史跨代精英个体的平均值,另一种是取历史当代精英个体的平均值,熵可取为相应历史精英个体的方差.

在 CBEA 中进化和变异是统一的,进化式变异是进化和变异融合,可以用来进行局部求精或跳出小局部,而突变则用来在全局范围内寻找新的极值搜索区域.算法可以判断出当前的进化状况,进而可以自适应地进行调整.

相比传统遗传算法,当种群中多数个体的适应值相差不大时交叉操作就显得无能为力,此时算法容易陷入局部解而不能通过交换解决,突然变异能够使之摆脱局部收敛而跃出局部解,但后期的变异可能破坏已产生的对形成最优解有建设性作用的模块.CBEA 可以有效避免遗传算法的这个缺点,因为进化式变异和突变均利用了历史搜索结果.进化算法的详细结构如图 3 所示.

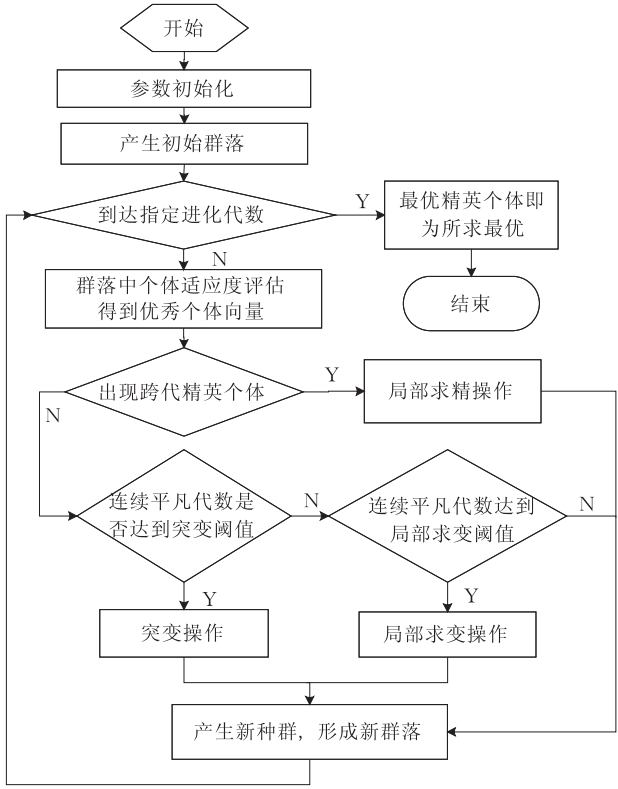


图 3 进化算法结构图

4 仿真实验及分析

为了验证算法的有效性,我们选用如下经典非约束类测试函数进行对比实验.

$$f_1 = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, -100 \leq x_i \leq 100, i = \{1, 2\}$$
,即 Schaffer 函数,其在  $(0, 0)$  处取得全局最小值 0.

$$f_2 = \left(4 - 2.1x_1^2 + \frac{1}{3}x_1^4\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2, -3 \leq x_1 \leq 3, -2 \leq x_2 \leq 2$$
,也称 Six-Hump Camel Back 函数,有两个全局最小值  $-1.03162845348988$ .

$$f_3 = \left\{ \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \right\} \times \left\{ \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \right\}, -10 \leq x_j \leq 10, j = \{1, 2\}$$
,也称 Shubert

函数,有 18 个全局最小值  $-186.7309088310239$ .

$f_4 = -\cos(x_1)\cos(x_2) \cdot \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$ ,  $-100 \leq x_i \leq 100, x = \{1, 2\}$ , 也称 Eazom 函数,在  $(x_1, x_2) = (\pi, \pi)$  时取得全局最小值  $-1$ .

$f_5 = \sum_{i=1}^n x_i^2$ ,  $-100 \leq x_i \leq 100, i = 1, 2, \dots, n, n = 10$ , 即 Sphere 函数,在  $(0, 0, \dots, 0)$  处取得全局最小值 0.

$f_6 = 20 + \exp(1) - 20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(-\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$ ,  $-32.768 < x_i < 32.768$ ,  $i = 1, 2, \dots, n, n = 10$ , 即 Ackley 函数,在  $(0, 0, \dots, 0)$  处有全局最小值 0.

$f_7 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ ,  $-600 < x_i < 600, i = 1, 2, \dots, n, n = 10$ , 即 Griewank 函数,在  $(0, 0, \dots, 0)$  处有全局最小值 0.

$f_8 = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$ ,  $-5.12 < x_i < 5.12, i = 1, 2, \dots, n, n = 10$ , 即 Rastrigin 函数,在  $(0, 0, \dots, 0)$  处有全局最小值 0.

#### 实验 1. CBEA 算法搜索过程和执行效率分析.

以函数  $f_1$  为例,实验的初始值设置:群落规模 500,群落丰富度 10,群落中种群的规模为  $[150, 100, 75, 25, 25, 25, 25, 25, 25, 25]$ ,适应度函数取为目标函数  $fitness = f_1$ ,进化代数 20,阈值  $\lambda_{local} = 2$ ,  $\lambda_{global} = 10$ ,进化系数  $K = 10$ ,进化式变异系数  $L = \lceil \sqrt{K} \rceil$ ,进化模式初始取值为  $EP(E_{x_1}, E_{n_1}, He_1, E_{x_2}, E_{n_2}, He_2) = (40, 100, 0.01, 20, 100, 0.01)$ .

以一次随机运行为例分析结果的收敛情况,最优点

$$f_1(8.814651527942963e-005, 1.489411529440795e-004) = 2.998322828906552e-008,$$

分别做出  $x_1, x_2$  和函数值的收敛曲线,横坐标为代数,纵坐标为函数值或  $x_1, x_2$  的值,如图 4 所示. 分析如下: (1) 算法共运行 20 代,在第 15 代找到了全局最优点. (2) 算法在第 1, 2, 3, 11 进化代 4 次寻找新的变化较大的局部极值点邻域,经过求精后迅速从第 1, 2 个逃出,在第 3 个局部极值点邻域连续求精 8 个进化代,其中找到 5 个跨代精英. 由于阈值  $\lambda_{local} = 2$ ,所以当算法在连续两代没有发现跨代精英的时候,就要进行变异,第 11 代进行变异,并立即找到了新的跨代精英,表明了算法具有良好的变异特性,使得算法能快速跳出局部极值邻域并进入新的

求精阶段. (3) 第 18, 19, 20 代为连续变异,没有找到跨代精英,算法结束. (4) 观察连续非平凡代数,大量实验表明连续非平凡代数一般不超过 5,表明算法有很好的局部求精能力.

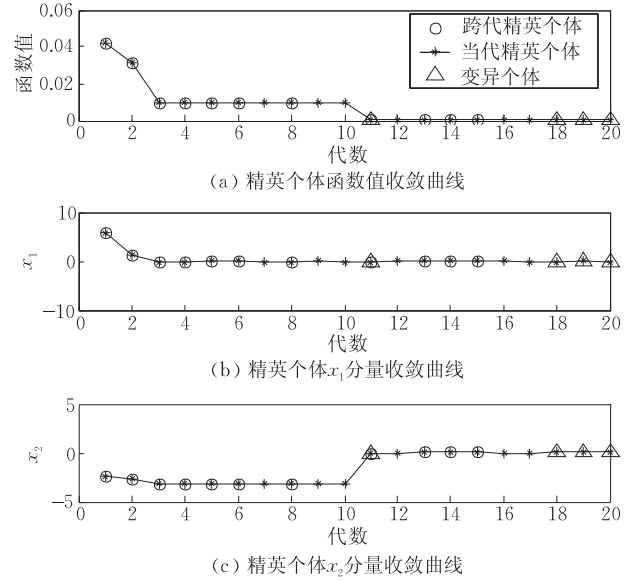


图 4 进化过程中各项指标的收敛曲线

以上分析表明算法不但能够快速定位全局极值点所在邻域,而且具有高效的局部求精能力,能够较好地避免遗传算法易陷入局部最优解和选择压力过大造成的早熟收敛等问题.

为了更好地了解算法的执行情况,下面对  $x_1$  和  $x_2$  的搜索空间的演变进行分析.  $x_1$  的变化情况如图 5 所示,图中给出了第 1, 3, 5, 14 进化代  $x_1$  的搜索空间,算法在第 3 代,建群种已经接近了最优解邻域,第 5 代的时候群落中除了有两个种群外其余的种群都回归到最优解附近,数据表明在第 5 进化代  $x_1$  已经很逼近最优值了.

$x_2$  的搜索空间的变化情况如图 6 所示,图中给出了第 1, 6, 12, 15 进化代  $x_2$  的搜索空间,直到第 6 代时  $x_2$  仍局限在局部最优邻域  $-3.14$  附近,在第 11 代出现变异,第 12 代的时候其建群种开始接近了最优解邻域,基本确定了下一步演化的方向,第 15 代时群落中所有 10 个种群都回归到最优解附近,正是在 15 代算法取得了最优解(见图 4).

为了考查 CBEA 的执行时间和计算精度,针对函数  $f_1$ ,算法根据不同的总代数各运行 100 次,由于运算结果呈非对称偏态分布,不适合用平均值作为测度,因而做出每次运行结果的频度统计,如表 1 所示. 算法的配置为:群落大小为 500,  $\lambda_{local} = 2$ ,  $\lambda_{global} = 10$ ,  $K = 10$ ,  $L = \lceil \sqrt{K} \rceil$ . CBEA 的执行时间受运行的总代数、适应度函数的复杂度以及群落

大小的影响,对于函数  $f_1$  每次运行的平均时间如表 1 所示. 实验机器为赛扬笔记本 1.5GHz 主频,

512MB 内存,仿真软件为 Matlab7. 1.

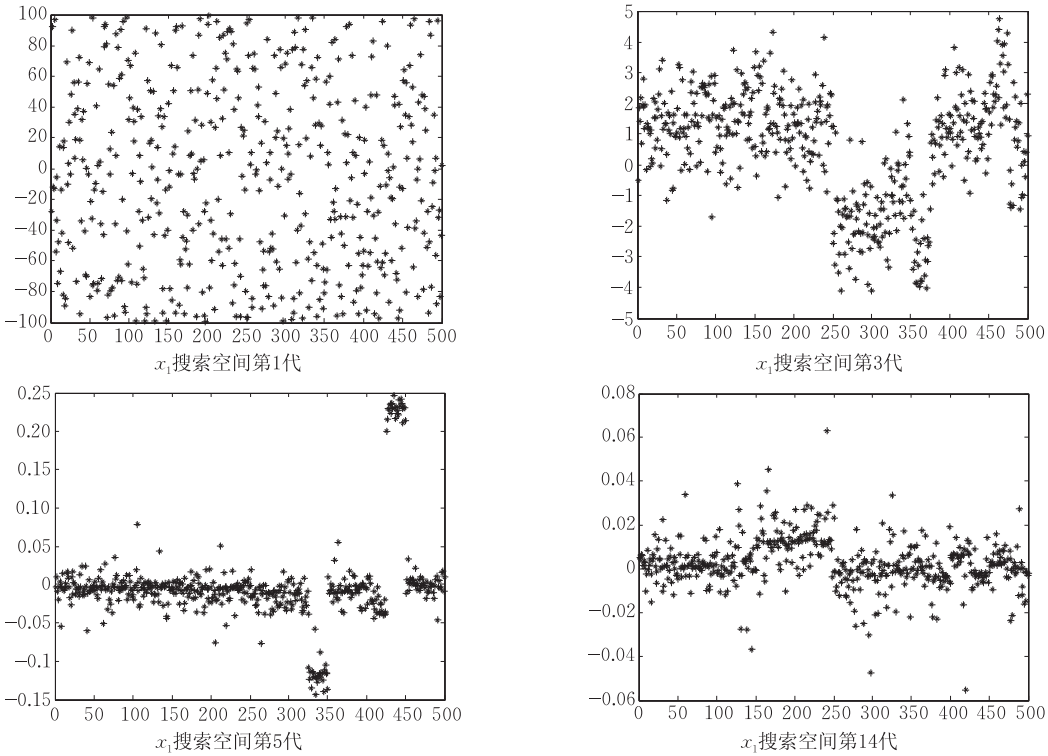


图 5  $5 \times 1$  的搜索空间变化情况(横轴为个体编号,纵轴为个体的适应值)

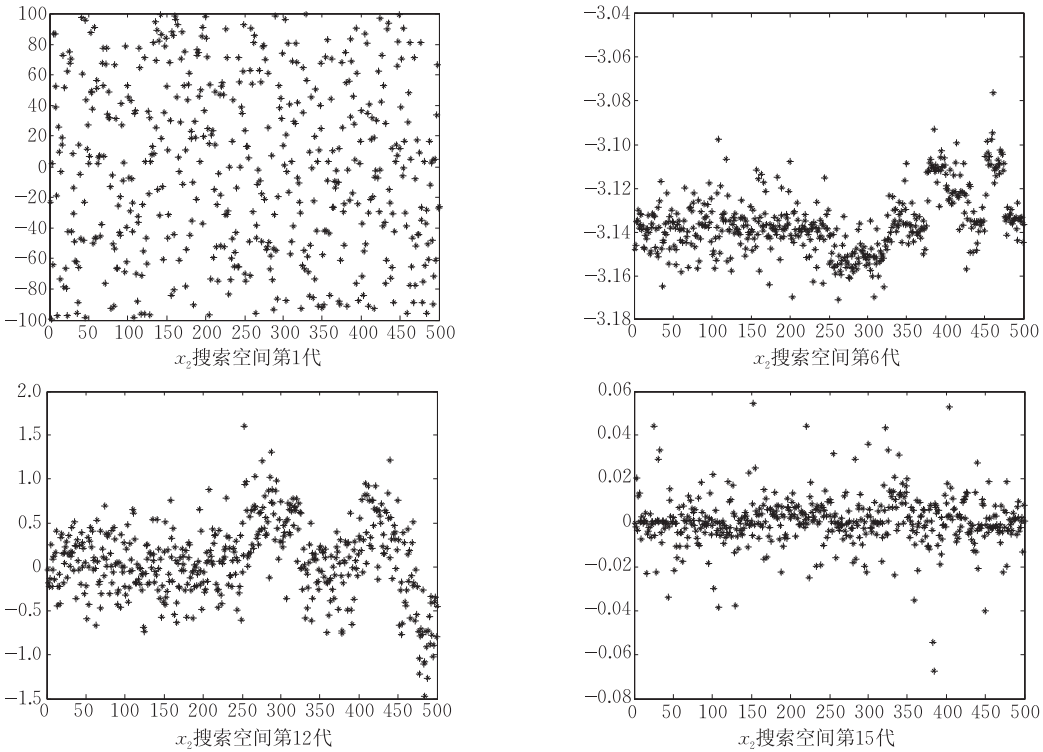


图 6  $6 \times 2$  的搜索空间变化情况(横轴为个体编号,纵轴为个体的适应值)

在上面的初始配置下,对于函数  $f_1$  当运行总代数大于 50 时,成功率基本达到 100%,且平均精度

优于  $10^{-10}$  数量级. 当运行总代数大于 50 时,算法得到精确解的比率大于 90%. 如果提高群落的大小,

那么能够进一步提高成功率. 当算法运行的总代数 行效率还是相当高的.  
小于 100 时算法能够在 0.1s 内完成寻优, 算法的执

表 1 结果频度统计表

运行次数	每次运行代数	运行结果频度统计(次)						每次运行平均时间/s
		0 (精确解)	$f_1$ ([1E-17, 1E-13])	$f_1$ ([1E-12, 1E-10])	$f_1$ ([1E-09, 1E-06])	$f_1$ ([1E-05, 1E-04])	$f_1$ (1E-03) (失败)	
100	30	40	45	7	5	0	3	0.02266
100	50	95	2	2	1	0	0	0.04125
100	80	95	3	2	0	0	0	0.06297
100	100	97	3	0	0	0	0	0.07781
100	150	99	1	0	0	0	0	0.11562

实验 2. CBEA 收敛性实验.

为了说明 CBEA 的收敛性, 我们利用所选取的所有测试函数对算法的演化情况进行了跟踪并绘制各自的演化曲线. CBEA 对每个函数都计算 50 次, 每次演化 100 代; 各个函数的演化曲线在某一代的值是 50 次实验中算法演化到该代时所有搜索到的最优解的平均值. 实验中, CBEA 的群落大小均取 1000, 群落丰富度为 10, 群落中种群的规模为 [300, 200, 150, 50, 50, 50, 50, 50, 50, 50].  
算法对所有函数演化曲线如图 7 所示(由于各

个搜索范围和精确解位置不尽不同, 故未将所有收敛曲线画在同一坐标系中).

由图 7 可知, 无论测试函数是 2 维还是 10 维, CBEA 都能够很快地收敛到最优解的水平, 除了  $f_3$  之外, 其它函数都能够在前 10 代逼近最优解, 而  $f_3$  也能够 在 25 代左右逼近其最优解. 事实上, 实验数据表明, 所有函数都能够 在 30 代左右到达或非常接近最优解. 不过由于显示粒度的原因不能在图中完全表现出来.

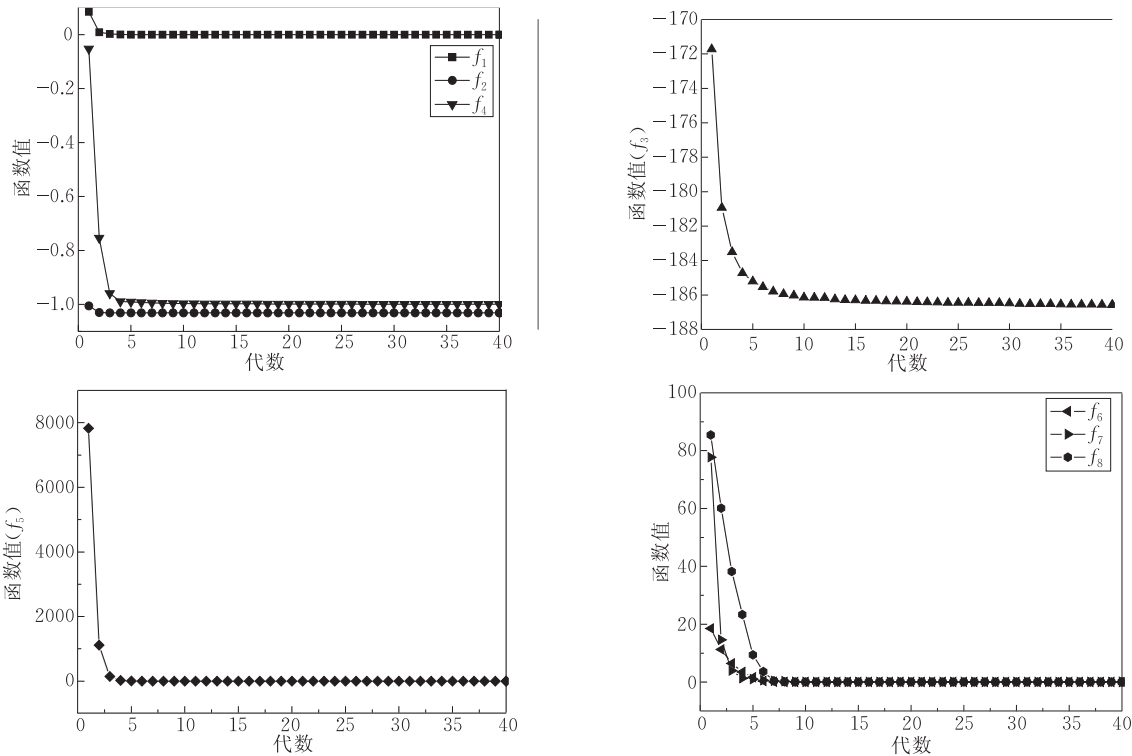


图 7 CBEA 计算函数  $f_1 \sim f_8$  的收敛曲线

实验 3. 全局搜索能力对比分析.

为了说明 CBEA 的全局搜索能力, 我们分别与遗传算法和 PSO 算法进行了比较. 前者是基于函数

$f_1 \sim f_4$ , 后者是基于  $f_5 \sim f_8$ .

遗传算法方面, 选取了传统 GA<sup>[1]</sup> (采用浮点编码、非一致变异算子和算术交叉算子) 和直接用局部



快速微调遗传算法 (FLAGA)<sup>[12]</sup> 和 CBEA 进行比较. CBEA 的参数设置同实验 2. 实验结果的对比情况如表 2 所示, 其中 GA 与 FLAGA 的数据来自文献[12].

表 2 本文算法与遗传算法比较

函数	GA T	GA 平均值	GA 成功次数	FLAGA T	FLAGA 平均值	FLAGA 成功次数	CBEA T	CBEA 平均值	CBEA 成功次数
$f_1$	300	0.0819251999945	15	300	7.396492233593399E-6	50	100	2.220446049250313E-18	50
$f_2$	300	-1.03131046699639	50	200	-1.03162845348226	50	100	-1.03162845348986	50
$f_3$	200	-186.366281815724	48	300	-186.7309088006962	50	100	-186.7309088310227	50
$f_4$	300	-0.60547253510119	34	300	-0.999999999999999	50	100	-1.0	50

从表 2 可以看出, CBEA 在所有测试函数中都表现得更好. 它用比其它算法更少的进化代数取得了更好的函数平均值, 并且 50 次的成功次数 (即全部成功) 也说明算法具有良好的稳定性. 此外, 实验中 CBEA 的达到精确解的比例相当高, 由于篇幅限制没有列出解的频度统计表.

PSO 算法方面, 选取了经典的 PSO 算法<sup>[13]</sup> 和 Liang 等新近提出的 PSO 改进算法 CLPSO<sup>[14]</sup> 作为比较对象. CBEA 的参数设置同实验 2. 为了便于比较, 使用了与文献[14]相同的统计参数, PSO 和 CLPSO 的数据也来自文献[14]. 实验结果对比如表 3 所示.

表 3 本文算法与 PSO 算法比较

函数	PSO		CLPSO		CBEA	
	平均值	标准方差	平均值	标准方差	平均值	标准方差
$f_5$	7.96E-51	3.56E-50	5.15E-29	2.16E-28	0	0
$f_6$	1.58E-14	1.6E-14	4.32E-14	2.55E-14	0	0
$f_7$	9.69E-02	5.01E-02	4.56E-03	4.81E-03	0	0
$f_8$	5.82E+00	2.96E+00	0	0	0	0

由表 3 可知, 对函数  $f_5 \sim f_8$ , CBEA 在准确性和稳定性都具有较为显著的优势. 事实上对于这 4 个测试函数, CBEA 每次都能够找到精确最优解, 因此其平均解即为精确最优解且所有解的标准方差为 0; 而 PSO 对这些函数都不能找到最优解, CLPSO 也仅在测试函数  $f_8$  上具有相同的精确性和稳定性.

进化母体.

云模型和进化计算思想的成功结合拓宽了云模型的应用领域, 也为进化计算的研究进行了新的探索和尝试. 本文的研究工作仅是这一交叉研究领域的开始, 如何拓宽算法在优化问题中的应用范围和对算法迭代过程的理论分析均是有意义, 也是我们下一步的研究方向.

5 结论与展望

基于云模型在定性概念与其定量数值表示之间转换过程中的优良特性, 结合进化计算的基本思想, 本文提出一种基于云模型的进化算法 (CBEA). 该算法能够自适应控制遗传、变异的程度和搜索空间的范围, 从而可以快速使算法收敛到最优, 较好地避免了遗传算法易陷入局部最优解和选择压力过大造成的早熟收敛等问题.

CBEA 不像遗传算法那样注重父代与子代的遗传细节 (通过基因表达遗传操作) 上的联系, 而是通过云模型把进化和遗传作为一种定性知识进行描述, 进而通过不确定的定性定量转换, 把定性知识转化成若干定量实现 (子代个体). 在每一个进化代, 通过对个体的适应性评估得到优秀个体, 作为下一代

参 考 文 献

[1] Wang Zheng-Zhi, Bo Tao. Evolutionary Computing. Changsha: National Univeristy of Defence Technology Press, 2000 (in Chinese)  
(王正志, 薄涛. 进化计算. 长沙: 国防科技大学出版社, 2000)

[2] Holland J H. Adoption in Natural and Artificial System. Ann Arbor: The University of Michigan Press, 1975

[3] Goldberg D E. Genetic Algorithm in Search, Optimization and Machine Learning. MA: Addison Wesley, 1985

[4] Davis L. Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold, 1991

[5] Schwefel H P. Numerical Optimization of Computer Models. Chichester: Wiley, 1981

[6] Li De-Yi, Du Yi. Artificial Intelligence with Uncertainty. Beijing: National Defense Industry Press, 2005(in Chinese)

(李德毅, 杜鹁. 不确定性人工智能. 北京: 国防工业出版社, 2005)

[7] Li De-Yi, Liu Chang-Yu. Study on the universality of the normal cloud model. Engineer and Science of China, 2004, 6(8): 28-33(in Chinese)

(李德毅, 刘常昱. 论正态云模型的普适性. 中国工程科学, 2004, 6(8): 28-33)

[8] Li De-Yi, Liu Chang-Yu, Du Yi, Han Xu. Artificial intelligence with uncertainty. Journal of Software, 2004, 15(9): 1583-592(in Chinese)

(李德毅, 刘常昱, 杜鹁, 韩旭. 不确定性人工智能. 软件学报, 2004, 15(9): 1583-592)

[9] Li De-Yi. Uncertainty in knowledge representation. Engineer and Science of China, 2000, 2(10): 73-79(in Chinese)

(李德毅. 知识表示中的不确定性. 中国工程科学, 2000, 2(10): 73-79)

[10] Eberhart R, Kennedy J. A new optimizer using particle swarm theory//Proceedings of the 6th International Symposium on Micro Machine and Human Science. Piscataway, Nagoya, Japan, 1995: 39-43

[11] Kennedy J, Eberhart R. Particle swarm optimization//Proceedings of the IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948

[12] Liu Xi-Chun, Yu Shou-Yi. A genetic algorithm with fast local adjustment. Chinese Journal of Computers, 2006, 29(1): 100-105(in Chinese)

(刘习春, 喻寿益. 局部快速微调遗传算法. 计算机学报, 2006, 29(1): 100-105)

[13] Shi Y, Eberhart R C. A modified particle swarm optimizer//Proceedings of the IEEE International Conference on Evolutionary Computation. Perth, Australia, 1998: 69-73

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



**ZHANG Guang-Wei**, born in 1970, Ph. D.. His research interests include evolutionary computing, artificial intelligence, machine learning, the next network.

**HE Rui**, born in 1976, Ph. D., lecturer. His research interests include evolutionary computing, mobile computing.

**LIU Yu**, born in 1980, Ph. D. candidate. His research interests include artificial intelligence.

**LI De-Yi**, born in 1944, Ph. D. supervisor, member of Chinese Academy of Engineering. His research interests include artificial intelligence and command automation.

**CHEN Gui-Sheng**, born in 1965, Ph. D., lecturer. His research interests include artificial intelligence, complex network and command automation.