协方差矩阵自适应演化策略学习机制综述

李焕哲[1,2],吴志健[1],汪慎文[2],郭肇禄[3]
(1. 武汉大学计算机学院软件工程国家重点实验室,湖北武汉 430072;
2. 河北地质大学信息工程学院,河北石家庄 050031;3. 江西理工大学理学院,江西赣州 341000)

摘　要： 基于协方差矩阵自适应(CMA)的演化策略算法(ES)是一种优秀的、不依赖于梯度信息的随机局部优化算法. 基于CMA的学习机制使其对搜索空间的任意可逆线性变换具有不变性,对于病态的、高度不可分的问题有优秀的求解能力. CMA学习机制具有较强的数学理论基础,这对设计其他演化算法有很好的借鉴意义. 本文旨在详细分析CMA-ES的各种学习机制,并给出其所依赖的主要理论基础. 最后通过实验比较CMA-ES各种变体的优势与不足,并着重比较本文改进的CMA-ES变体与其它变体在性能上的差异.

关键词： 演化策略；协方差矩阵自适应；自适应学习；多元正态分布
中图分类号： TP18　**文献标识码：** A　**文章编号：** 0372-2112 (2017)01-0238-08
电子学报 URL： http://www.ejournal.org.cn　**DOI：** 10.3969/j.issn.0372-2112.2017.01.033

The Overview of Learning Mechanism of Covariance Matrix Adaptation Evolution Strategy

LI Huan-zhe[1,2], WU Zhi-jian[1], WANG Shen-wen[2], GUO Zhao-lu[3]
(1. *State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan, Hubei* 430072, *China*;
2. *School of Information Engineering, Hebei GEO University, Shijiazhuang, Hebei* 050031, *China*;
3. *School of Science, JiangXi University of Science and Technology, Ganzhou, Jiangxi* 341000, *China*)

Abstract: The evolution strategy (ES) based on covariance matrix adaptation (CMA) is an excellent, gradient-free stochastic local optimization method. The learning mechanism based on CMA enables evolution strategy algorithm to have invariance to any invertible linear transformation of the search space, and to have outstanding capability for solving the ill-conditioned and/or highly non-separable problems. The learning mechanism of CMA has a solid theoretical foundation in mathematics, which may have a certain reference significance to guide the design of other evolutionary algorithms. This paper aims at analyzing the learning mechanisms of CMA-ES in detail, and providing its main mathematical foundations. Finally, the advantages and disadvantages of various CMA-ES variants are compared by a series of experiments, and the difference in performance is compared seriously between our improved variant and other CMA-ES variants.

Key words: evolution strategy; covariance matrix adaptation; adaptive learning; multivariate normal distribution

1　引言

演化策略[1]属于一种随机搜索算法,主要用于解决现实世界中一些传统优化算法难于处理的优化问题,例如不连续,不可微,高维度等. 它通过对已找到的优秀的搜索点增加一个随机向量来改变搜索的步长和方向,这个步骤通常称之为变异. 在优化过程中适当地利用这些变异更新协方差矩阵信息,允许搜索分布学习和使用一个可变尺度的协方差矩阵. 变异对于演化策略来说非常重要,因为它主要通过随机变异来改变搜索点的位置,而不像差分演化算法[2,3]和粒子群优化[4,5]算法通过个体间的差分向量或向优秀个体学习的机制来改变搜索点位置. 变异操作是驱动演化策略算法最重要的动力来源.

为了控制变异操作,用于控制变异分布的策略参数被引入,这些策略参数包括全局(整体)步长、单独步

收稿日期:2015-10-10;修回日期:2016-03-27;责任编辑:李勇锋
基金项目:国家自然科学基金(No. 61364025, No. 61402481);江西省自然科学基金(No. 20151BAB217010);河北省自然科学基金(No. F2015403046);武汉大学软件工程国家重点实验室开放基金(No. SKLSE2014-10-04);河北省科学技术支撑项目(No. 12210319)

长及搜索分布的旋转角度,这些策略参数是内生的,和搜索点的分布密切相关. 优秀的策略参数会提高由它所生成的搜索点被选中的概率,反之,被选中并被遗传到下一代的搜索点也影响对策略参数的调整,这一策略参数调整的过程被称为变异策略参数控制(Mutation Strategy Parameter Control,MSC)[6]. 传统的 MSC 设置往往需要特定的先验知识,而且优良的参数设置会因问题而异,对于通常的现实问题而言,并没有对策略参数最佳选择的特定知识可用,所以策略参数调整很难达到满意的效果. 因此自适应的策略参数应运而生,Schwefel[6]在 1981 年提出了搜索分布旋转角自适应,该方法通常被称为相关变异,通过对搜索分布进行任意角度的旋转及步长控制可生成任意方向的正态变异分布,不过这种方法依赖于坐标系,不具有对搜索空间线性变换的不变性. Ostermeier 等[7]在 1994 年提出了去随机化的步长控制(Derandomized Mutative Step-Size Control,DSC),可以在小种群下自适应控制单独的步长变异. Hansen 和 Ostermeier[8]在 1996 年提出了基于 DSC 概念的协方差矩阵自适应学习机制,协方差矩阵的引入使得搜索分布可以完全独立于坐标系,具有对搜索空间任意可逆线性变换的不变性. Hansen 和 Ostermeier[9]于 2001 年在前面工作的基础上提出了完全去随机化的协方差矩阵自适应学习机制,引入"去随机化"和"积累演化路径"两个概念,以消除随机性对搜索分布波动的影响,使策略参数的改变确定性的与目标参数的变化相联系. 2002 和 2003 年 Müller 和 Hansen 等[10,11]提出了秩-μ 更新机制,该机制可充分利用在较大种群中所包含的信息,通过利用这些信息来减少运行时间和增强全局搜索能力.

在协方差矩阵更新过程中,需要用到对协方差矩阵的特征分解. 一个普通协方差矩阵特征分解需要 $\Theta(n^3)$步,其中 n 为问题维度. 为了把特征分解的复杂度降为二次的,Hansen 等[9]使用了对协方差矩阵进行延迟分解的方法,提出了(μ_w,λ)-PUd-CMA 算法,但是这种方法会使用过时的变异分布. 2006 年 Igel 等[12]把 Cholesky 因式分解方法引入协方差矩阵更新,把对协方差矩阵的显式更新改为对协方差矩阵 Cholesky 因子的更新,这种方法不需要显式更新协方差矩阵,而通过对 Cholesky 因子的更新来隐式更新协方差矩阵,该方法把算法计算复杂度从 $\Theta(n^3)$降为了 $\Theta(n^2)$. 他们提出的这种增量更新技术属于秩-1 更新,被经常用于卡尔曼滤波领域[13]. Igel 等把这种 Cholesky 因式分解方法应用到了$(1+1)$-CMA-ES 中,由此提出了$(1+1)$-Cholesky-CMA-ES 算法. 在该算法中取消了演化路径的使用,这导致$(1+1)$-Cholesky-CMA-ES 算法在求解某些问题时学习效率变慢. 2007 年 Igel 等[14]直接把 Cholesky 因式分解方法应用于$(1+\lambda)$-CMA-ES 算法,从而提出了$(1+\lambda)$-Cholesky-CMA-ES,并将该算法应用于求解多目标优化问题. 2009 年 Suttorp 等[15]在 Igel 的工作基础上进行改进,在 Cholesky 因式分解方法中解决了演化路径计算的问题,使得具有秩-1 更新的(μ_w,λ)-CMA-ES 算法的计算复杂度也降为了 $\Theta(n^2)$,这很大程度上提高了(μ_w,λ)-CMA-ES 算法在高维问题上的运行效率. 2008 年 Ros 等[16]提出一种以对角协方差矩阵替换常规协方差矩阵的方法,该方法把算法时间和空间复杂性降到了线性的. 由于对角矩阵的引入导致该方法失去了对搜索空间的旋转不变性,以致在某些不可分的问题上性能下降严重,但是在一些高维可分的问题上却性能优异,甚至在某些高维多峰问题上也性能良好. 经过多年完善改进,CMA 学习机制已经成为演化策略中对策略参数控制最行之有效的一种方法,并成功应于全局优化[17,18]、多峰优化[19,20]、多目标优化[14,21]、大规模优化[22,23]等领域.

CMA-ES 中的学习机制主要包括协方差矩阵自适应和整体步长自适应. 这两种学习机制相互独立的进行自适应学习,整体步长的自适应并不依赖于协方差矩阵的自适应,这样做的目的是为了加快收敛的速度,但是这两种学习机制都依赖于对已选择的多元正态随机变量的学习.

2　理论基础

为了把协方差矩阵自适应学习机制阐述清楚,本文把协方差矩阵自适应学习的主要理论基础总结为定理 1 到定理 5 (篇幅关系证明部分被省略). 另外,定理 6 和定理 7 为有关 Cholesky 因子增量更新的相关定理,引自文献[12,15].

2.1　协方差矩阵自适应学习的主要理论基础

定理 1　若 n 维随机向量 $\boldsymbol{x}$ 符合多元正态分布 $N(\boldsymbol{0},\boldsymbol{I})$,则它的任意线性组合 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{\mu}$ 服从多元正态分布 $N(\boldsymbol{\mu},\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})\sim \boldsymbol{A}N(\boldsymbol{0},\boldsymbol{I})+\boldsymbol{\mu}$,其中 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$.

定理 2　如果有限个 n 维向量 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\in\mathbb{R}^n$,给每个向量乘以一个 $N(0,1)$正态分布随机数,则 $N(0,1)\boldsymbol{x}_1+\cdots+N(0,1)\boldsymbol{x}_m$ 是一个均值为零向量的多元正态分布 $N(\boldsymbol{0},\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}})$.

定理 3　一个实对称正定矩阵 $\boldsymbol{C}$ 可特征分解为 $\boldsymbol{B}\boldsymbol{D}^2\boldsymbol{B}^{\mathrm{T}}$,$\boldsymbol{B}$ 为正交矩阵,$\boldsymbol{B}$ 的每一列对应 $\boldsymbol{C}$ 的一个特征向量;$\boldsymbol{D}$ 为对角矩阵,$\boldsymbol{D}$ 的每个对角元素对应 $\boldsymbol{C}$ 的一个特征值的平方根,并且每个对角元素与 $\boldsymbol{B}$ 中的相应列对应.

定理 4　对于非零向量 $\boldsymbol{v}=[v_1,\cdots,v_n]^{\mathrm{T}}$,$n\geqslant 2$,令 $\boldsymbol{A}=\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}$,则 $\boldsymbol{A}$ 满足以下结论:

a. 是秩为 1 的 n 阶对称矩阵.

b. 它是一个非负定矩阵,只有一个非零的特征值 $\lambda_n=\|\boldsymbol{v}\|^2$,且 v 是特征值 λ_n 对应的特征向量.

推论 1 对于非零向量 $\boldsymbol{v}=[v_1,\cdots,v_n]^{\mathrm{T}},n\geqslant 2$,如果乘以一个 $N(0,1)$ 正态分布随机数,则 $N(0,1)\boldsymbol{v}$ 为线性分布,且该分布为所有均值为零的正态分布中具有最大的概率产生向量 $\boldsymbol{v}$.

定理 5 m 个相互独立的 n 维向量 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots\boldsymbol{v}_m,n\geqslant 2$,矩阵 $\sum_{i=1}^{m}\boldsymbol{v}_i\boldsymbol{v}_i^{\mathrm{T}}$ 为秩为 $\min(m,n)$ 的对称矩阵.

2.2 Cholesky 因子增量更新的理论基础

在文献[12]中提出了定理 6,该定理给出了 Cholesky 因子如何进行增量更新,在文献[15]中提出了定理 7 和推论 2,定理 7 给出了 Cholesky 因子的逆矩阵如何进行增量更新,推论 2 指出了如何通过协方差矩阵的隐式更新把计算复杂度降为 $\Theta(n^2)$.

3 协方差矩阵自适应学习机制

3.1 多元正态分布

演化策略采用的搜索分布是多元正态分布,多元正态分布 $N(\boldsymbol{m},\boldsymbol{C})$ 由均值 $\boldsymbol{m}\in\mathbb{R}^n$ 和对称的正定协方差矩阵 $\boldsymbol{C}\in\mathbb{R}^{n\times n}$ 唯一确定,均值 $\boldsymbol{m}$ 确定分布的中心位置,协方差矩阵 $\boldsymbol{C}$ 确定分布的形状、方向和范围。根据二次型的知识,每个协方差矩阵都可以由唯一的一个超椭球体 $\{\boldsymbol{x}\in\mathbb{R}^n\mid\boldsymbol{x}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{x}=1\}$ 标识[24]. 根据定理 3 可知 $\boldsymbol{C}=\boldsymbol{B}\boldsymbol{D}^2\boldsymbol{B}^{\mathrm{T}}$,$\boldsymbol{B}$ 的所有列构成一个正交坐标系统,$\boldsymbol{B}$ 的每一列向量代表椭球体的一个主轴方向,这种类型的正态分布是不依赖于坐标轴的。协方差矩阵的特征分解可看成是对协方差矩阵的主成份分析,$\boldsymbol{B}$ 的列向量指明各成份的方向,$\boldsymbol{D}$ 的对角元素指明各成份的大小。

椭球体的主轴对应矩阵 $\boldsymbol{C}$ 的特征向量,主轴长度的平方对应矩阵 $\boldsymbol{C}$ 的特征值。正态分布 $N(\boldsymbol{m},\boldsymbol{C})$ 根据定理 1 可以被写成以下不同的形式:

$$\begin{aligned}N(\boldsymbol{m},\boldsymbol{C})&\sim\boldsymbol{m}+N(\boldsymbol{0},\boldsymbol{C})\\&\sim\boldsymbol{m}+N(\boldsymbol{0},\boldsymbol{B}\boldsymbol{D}^2\boldsymbol{B}^{\mathrm{T}})\\&\sim\boldsymbol{m}+N(\boldsymbol{0},(\boldsymbol{B}\boldsymbol{D})(\boldsymbol{B}\boldsymbol{D})^{\mathrm{T}})\\&\sim\boldsymbol{m}+\boldsymbol{B}\underbrace{\boldsymbol{D}N(\boldsymbol{0},\boldsymbol{I})}_{\sim N(\boldsymbol{0},\boldsymbol{D}^2)}\end{aligned}\tag{1}$$

3.2 协方差矩阵自适应的原理

协方差矩阵自适应是一个完全去随机化的自适应方案[9,24]. 协方差矩阵自适应的一般策略是改变协方差矩阵以使得那些有希望产生更大适应度进步的变异步能够更多的被生成. 实现协方差矩阵自适应较常见的方法是直接对协方差矩阵进行增量式更新,其形式为:

$$\boldsymbol{C}^{(g+1)}=\alpha\boldsymbol{C}^{(g)}+\beta\boldsymbol{V}^{(g)}\tag{2}$$

矩阵 $\boldsymbol{C}^{(g)}\in\mathbb{R}^{n\times n}$ 和 $\boldsymbol{V}^{(g)}\in\mathbb{R}^{n\times n}$ 必须保证是正定的,$\alpha,\beta\in\mathbb{R}^+$ 是权重系数. 令 $\boldsymbol{v}^{(g)}\in\mathbb{R}$ 为搜索空间中有希望产生更大适应度进步的变异步,在下一次迭代中要增加产生变异步 $\boldsymbol{v}^{(g)}$ 的概率,则可采用秩 -1 更新形式:

$$\boldsymbol{C}^{(g+1)}=\alpha\boldsymbol{C}^{(g)}+\beta\boldsymbol{v}^{(g)}\boldsymbol{v}^{(g)^{\mathrm{T}}}\tag{3}$$

$N(\boldsymbol{0},\boldsymbol{C}^{(g+1)})$ 可被写成 $N(\boldsymbol{0},\alpha\boldsymbol{C}^{(g)})+N(\boldsymbol{0},\beta\boldsymbol{v}^{(g)}\boldsymbol{v}^{(g)^{\mathrm{T}}})$ 的形式,其中 $N(\boldsymbol{0},\alpha\boldsymbol{C}^{(g)})$ 为上一次迭代的正态分布,权重系数 α 用于减少旧的正态分布信息;根据推论 1 可知 $N(\boldsymbol{0},\beta\boldsymbol{v}^{(g)}\boldsymbol{v}^{(g)^{\mathrm{T}}})$ 为一个线性分布,它具有最大的概率产生向量 $\boldsymbol{v}^{(g)}$,这种变异分布倾向于产生在过去演化代中选择生成的变异步. 权重系数 β 用于降低新的正态分布信息对旧正态分布信息的影响. 这种更新规则使得旧的变异分布向线性分布 $N(\boldsymbol{0},\beta\boldsymbol{v}^{(g)}\boldsymbol{v}^{(g)^{\mathrm{T}}})$ 移动. 通过添加不同的 $\boldsymbol{v}_i,i=1,\cdots,m$,由定理 2 可知,可实现均值为零向量的任意多元正态分布. 由于变异向量 $\boldsymbol{v}^{(g)}$ 所生成的协方差矩阵被加到了 $\boldsymbol{C}^{(g+1)}$ 中,所以在下一代生成向量 $\boldsymbol{v}^{(g)}$ 所在方向的变异的概率会增加。最后,这种更新会导致选择前和选择后分布上的一致。如果两个分布相类似,在随机选择的情况下,最终的期望是协方差矩阵的变异分布将不会有任何改变发生。

3.3 演化策略的基本公式

在一个标准的演化策略算法中,第 g 代生成的第 i 个后代的目标向量 $\boldsymbol{x}_i^{(g+1)}\in\mathbb{R}^n$ 可按如下公式生成[15]:

$$\boldsymbol{x}_i^{(g+1)}\leftarrow\underbrace{\boldsymbol{c}_i^{(g)}}_{\text{重组}}+\underbrace{\boldsymbol{v}_i^{(g)}}_{\text{变异}}$$

向量 $\boldsymbol{c}_i^{(g)}$ 依赖于重组方式,重组方式可有全局离散重组、全局中值重组和全局加权重组. 在通常情况下多采用加权重组方式,让质量好的后代个体具有更大的权重系数. 变异向量 $\boldsymbol{v}_i^{(g)}$ 是一个符合 $N(\boldsymbol{0},\boldsymbol{C}^{(g)})$ 分布的多元正态分布随机向量. 要生成一个 $\boldsymbol{v}_i^{(g)}$,可先生成一个符合 $N(\boldsymbol{0},\boldsymbol{I})$ 分布的随机向量 $\boldsymbol{z}_i^{(g)}$,然后通过一个线性变换 $\boldsymbol{A}_i^{(g)}\in\mathbb{R}^{n\times n}$ 对 $\boldsymbol{z}_i^{(g)}$ 进行缩放和旋转,其中 $\boldsymbol{A}_i^{(g)}$ 为矩阵 $\boldsymbol{C}^{(g)}$ 的一个 Cholesky 因子($\boldsymbol{C}^{(g)}=\boldsymbol{A}^{(g)}\boldsymbol{A}^{(g)\mathrm{T}}$),即 $\boldsymbol{A}_i^{(g)}\boldsymbol{z}_i^{(g)}\sim N(\boldsymbol{0},\boldsymbol{C}_i^{(g)})$,$\boldsymbol{z}_i^{(g)}\sim N(\boldsymbol{0},\boldsymbol{I})$. 则生成抽样搜索点的基本公式可写成如下形式:

$$\boldsymbol{x}_k^{(g+1)}=\boldsymbol{m}_k^{(g)}+\sigma^{(g)}\underbrace{N(\boldsymbol{0},\boldsymbol{C}^{(g)})}_{\boldsymbol{BD}N(\boldsymbol{0},\boldsymbol{I})}\quad k=1,\cdots,\lambda\tag{4}$$

其中,$\boldsymbol{x}_k^{(g+1)}$ 第 $g+1$ 代中的第 k 个后代. $\boldsymbol{m}^{(g)}\in\mathbb{R}^n$ 第 g 代搜索分布中已选择个体的加权平均位置. $\boldsymbol{\sigma}^{(g)}\in\mathbb{R}^+$ 第 g 代整体标准偏差,即第 g 代的全局步长. $\boldsymbol{C}^{(g)}\in\mathbb{R}^{n\times n}$ 第 g 代的搜索分布的协方差矩阵,大小取决于标量因子 $\boldsymbol{\sigma}^{(g)^2}$. $\lambda\geqslant 2$ 为种群大小.

3.4 秩-1 更新

根据上面 3.2 节的式(3)可知,秩 -1 更新机制[9]通过向旧的协方差矩阵上累加新的矩阵增量实现更新,矩阵增量 $\boldsymbol{v}^{(g)}\boldsymbol{v}^{(g)^{\mathrm{T}}}$ 中的 $\boldsymbol{v}^{(g)}$ 为选择的变异步. 变异步长公式为 $\boldsymbol{y}^{(g+1)}=(\boldsymbol{m}^{(g+1)}-\boldsymbol{m}^{(g)})/\sigma^{(g)}$,$\boldsymbol{m}^{(g+1)}-\boldsymbol{m}^{(g)}$ 为

两代上已选择个体加权均值之差,其中除以 $\sigma^{(g)}$ 的目的是为了消除整体步长对变异步长的影响. 为了使用步长符号信息及充分挖掘一系列演化代之间的相关信息来进行协方差矩阵更新,Hansen 和 Ostermeier 引入了所谓的“演化路径”的概念[8,9]. 演化路径是指一系列连续演化代的变异步长之和,这个总和也被称为“积累路径”[9]. 演化路径公式如下:

$$\boldsymbol{p}_c^{(g+1)} = (1-c_c)\boldsymbol{p}_c^{(g)} + \sqrt{c_c(2-c_c)}\underbrace{\frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}}(\boldsymbol{m}^{(g+1)} - \boldsymbol{m}^{(g)})}_{\sqrt{\mu_{eff}}\boldsymbol{B}^{(g)}\boldsymbol{D}^{(g)}\langle \boldsymbol{z}\rangle_w^{(g+1)}} \tag{5}$$

$\sqrt{\mu_{eff}}$为方差有效选择质量,$\langle \boldsymbol{z}\rangle_w^{(g+1)}$ 为 u 个已选择随机向量的加权平均值,$\boldsymbol{0} \leqslant c_c \leqslant 1$ 为权重系数. 根据 3.2 节的式(3),“秩-1 更新”公式通过演化路径 $\boldsymbol{p}_c^{(g+1)}$ 可构造为:

$$\boldsymbol{C}^{(g+1)} = (1-c_1)\boldsymbol{C}^{(g)} + c_1\boldsymbol{p}_c^{(g+1)}\boldsymbol{p}_c^{(g+1)^{\mathrm{T}}} \tag{6}$$

c_1 是协方差矩阵学习率. 式(6)使用了协方差矩阵的显示更新,在式(4)和式(5)中都用到了 $\boldsymbol{B}^{(g)}$ 和 $\boldsymbol{D}^{(g)}$,这需要对 $\boldsymbol{C}^{(g)}$ 进行特征分解. 普通矩阵特征分解的计算复杂度为 $\Theta(n^3)$,为了把计算复杂度从 $\Theta(n^3)$ 降到 $\Theta(n^2)$,Igel 和 Suttorp 等[12,15]提出了隐式协方差矩阵更新机制,在这种新机制中不需要计算 $\boldsymbol{B}^{(g)}$ 和 $\boldsymbol{D}^{(g)}$,也就不需要矩阵的特征分解.

3.5　秩-μ 更新

秩-1 更新适合于小种群的情况下使用,而大种群的情况下,秩-1 更新不能充分利用大种群所提供的可用信息. 为了解决这一问题,Müller 和 Hansen 等提出了秩-μ 更新[10,11]. 秩-μ 更新是指选择子种群中最好的 μ 个个体进行协方差矩阵更新. 因为 μ 个个体是相互独立的,所以根据定理 5 知这 μ 个个体的秩最大为 min(n,u),这就是称为秩-μ 更新的原因. 引入秩-μ 更新有以下三点优势:(1)降低了噪声的影响;(2)增加了全局搜索能力;(3)可实现算法在并行机上的运行[10]. 对 μ 个个体的协方差矩阵按式(7)进行估计,$\boldsymbol{x}_{i:\lambda}^{(g+1)} - \boldsymbol{m}^{(g)}$ 为已选择个体 i 的实际变异步长.

$$\boldsymbol{C}_\mu^{(g+1)} = \sum_{i=1}^{\mu} w_i(\boldsymbol{x}_{i:\lambda}^{(g+1)} - \boldsymbol{m}^{(g)})(\boldsymbol{x}_{i:\lambda}^{(g+1)} - \boldsymbol{m}^{(g)})^{\mathrm{T}} \tag{7}$$

w_i 为个体 $\boldsymbol{x}_{i:\lambda}^{(g+1)}$ 的权重系数. 按照式(2)的形式,则秩-μ 更新公式可写为:

$$\boldsymbol{C}^{(g+1)} = (1-c_u)\boldsymbol{C}^{(g)} + c_u\sum_{i=1}^{\mu} w_i\boldsymbol{y}_{i:\lambda}^{(g+1)}\boldsymbol{y}_{i:\lambda}^{(g+1)^{\mathrm{T}}} \tag{8}$$

$\boldsymbol{y}_{i:\lambda}^{(g+1)} = (\boldsymbol{x}_{i:\lambda}^{(g+1)} - \boldsymbol{m}^{(g)})/\sigma^{(g)}$,$c_\mu$ 为协方差矩阵的学习率. “秩-μ 更新”利用学习先前代的信息以减小协方差矩阵更新对种群大小的依赖,尤其在种群较小时可增加更新的稳定性和可靠性,达到了协方差矩阵更新去随机化的目的.

3.6　合并秩-1 和秩-μ 更新

最终的协方差矩阵更新公式合并式(6)和式(8),以适应不同情况下协方差矩阵的更新[24].

$$\boldsymbol{C}^{(g+1)} = (1-c_1-c_u)\boldsymbol{C}^{(g)} + \underbrace{c_1\boldsymbol{p}_c^{(g+1)}\boldsymbol{p}_c^{(g+1)^{\mathrm{T}}}}_{\text{秩1更新}} + \underbrace{c_u\sum_{i=1}^{\mu} w_i\boldsymbol{y}_{i:\lambda}^{(g+1)}\boldsymbol{y}_{i:\lambda}^{(g+1)^{\mathrm{T}}}}_{\text{秩}\mu\text{更新}} \tag{9}$$

式(9)结合了式(8)和式(6)的优点. 一方面,“秩-μ 更新”有效利用了一代中所有已选择搜索点的信息,生成稳定可靠的协方差矩阵更新信息;另一方面,“秩-1 更新”有效利用了连续代之间步长的相关关系,以增加协方差矩阵更新的改变率. 前者适用于较大种群,后者对于小种群尤其重要.

4　整体步长自适应学习机制

式(9)并没有显性控制协方差矩阵分布的整体缩放,在协方差矩阵自适应的过程中,对于每一个被选择的变异步,协方差矩阵仅会在变异步方向增加信息,并且以因子 $1-c_1-c_\mu$ 来消退旧的信息以使协方差矩阵收缩. 这种隐性步长控制使得协方差矩阵整体缩放效率不高,无法满足算法快速收敛的要求.

为了控制步长 $\sigma^{(g)}$,Hansen 和 Ostermeier[9]引入上节中用到的演化路径来进行步长控制. 这种步长控制独立于协方差矩阵更新并被称为“积累路径长度控制”. 与式(5)相同的技术被用于构建演化路径 $\boldsymbol{p}_\sigma$,$\boldsymbol{p}_\sigma$ 近似为 $\boldsymbol{p}_c$ 的一个共轭演化路径[24]. 初始化 $\boldsymbol{p}_\sigma = \boldsymbol{0}$,则演化路径 $\boldsymbol{p}_\sigma^{(g+1)}$ 可写为:

$$\boldsymbol{p}_\sigma^{(g+1)} = (1-c_\sigma)\boldsymbol{p}_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)\mu_{\mathrm{eff}}}\underbrace{\boldsymbol{C}^{(g)^{-\frac{1}{2}}}\frac{\boldsymbol{m}^{(g+1)} - \boldsymbol{m}^{(g)}}{\sigma^{(g)}}}_{\boldsymbol{B}^{(g)}\langle \boldsymbol{z}\rangle_w^{(g+1)}} \tag{10}$$

c_σ 代表 $\boldsymbol{p}_\sigma$ 的学习率. $\boldsymbol{p}_\sigma^{(g+1)}$ 的长度和随机选择下的期望长度是可比较的. 更新 $\sigma^{(g)}$,只需比较 $\|\boldsymbol{p}_\sigma^{(g+1)}\|$和它的期望长度 $E\|N(\boldsymbol{0},\boldsymbol{I})\|$,则有以下全局步长公式:

$$\sigma^{(g+1)} = \sigma^{(g)}\exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p}_\sigma^{(g+1)}\|}{E\|N(\boldsymbol{0},\boldsymbol{I})\|} - 1\right)\right) \tag{11}$$

d_σ 是阻尼系数,$E\|N(\boldsymbol{0},\boldsymbol{I})\|$为 $N(\boldsymbol{0},\boldsymbol{I})$分布的随机向量的欧氏范数的期望.

5　(μ_w,λ)-Cholesky-CMA-ES

Igel 和 Suttorp 等[12,15]把 Cholesky 因式分解的方法应用于(1+1)-CMA-ES 和(μ_w,λ)-CMA-ES,使相关算法的复杂性降低为 $\Theta(n^2)$. 但 Suttorp 仅对秩-1 更新给出了该方法的算法描述,缺少对秩-μ 更新进行 Cholesky 因式分解的具体实现方法及相关性能测试. 基于 Suttorp 等[15]的工作,本文给出一个完整的(μ_w,λ)-Cholesky-

CMA-ES 算法. 该算法实现了同时对秩-1 和秩-μ 更新进行 *Cholesky* 因子增量更新，把算法的时间复杂度从 $\Theta(n^3)$ 降到 $\Theta((1+\mu)n^2)$，$\mu \ll n$. 协方差矩阵可按以下公式进行隐式更新：

$$\begin{aligned}\boldsymbol{C}^{(g+1)} &= \alpha\boldsymbol{C}^{(g)} + \beta\boldsymbol{p}_c^{(g+1)}\boldsymbol{p}_c^{(g+1)^{\mathrm{T}}} + \gamma\sum_{i=1}^{\mu} w_i\boldsymbol{y}_{i:\lambda}^{(g+1)}\boldsymbol{y}_{i:\lambda}^{(g+1)^{\mathrm{T}}} \\ &= \left[\begin{matrix}[[\alpha\boldsymbol{C}^{(g)} + \beta\boldsymbol{p}_c^{(g+1)}\boldsymbol{p}_c^{(g+1)^{\mathrm{T}}}] + \gamma w_1\boldsymbol{y}_{1:\lambda}^{(g+1)}\boldsymbol{y}_{1:\lambda}^{(g+1)^{\mathrm{T}}}] \\ + \gamma w_2\boldsymbol{y}_{2:\lambda}^{(g+1)}\boldsymbol{y}_{2:\lambda}^{(g+1)^{\mathrm{T}}} + \cdots\end{matrix}\right]\end{aligned}$$

其中，$\alpha = 1 - c_1 - c_\mu$，$\beta = c_1$，$\gamma = c_\mu$.

6 实验与分析

为了比较各种变体的优缺点，本节选择了 4 种常见 CMA-ES 变体与本文改进的(μ_w,λ)-Cholesky-CMA 变体进行比较。选择的 4 种变体分别是 $(1+1)$-Cholesky-CMA、$(1+\lambda)$-Cholesky-CMA、(μ_w,λ)-PUd-CMA 和 (μ_w,λ)-sep-CMA. 所有变体在 10 个测试函数上进行了对比实验，这些测试函数分别选自 CEC2005[25] 和 CEC2013[26] 国际会议上提供的标准测试函数，选择的测试函数具体如表 1 所示.

50 维及以下问题的最大评价次数设为 3.0E+5，其余问题的最大评价次数设为 1.0E+6，求解精度设为 1E-8. 代码由 matlab R2014a 实现，CPU 为 Intel Core i7-4790，内存为 8G. 表 2 给出了 5 种算法在 10 个函数上分别运行 25 次后求得的平均最优值，最优结果用粗体显示. 为了统计上的显著性，对求得的 25 组最优值，在 (μ_w,λ)-Cholesky-CMA 和其它算法之间以 0.05 为显著性水平执行 Wilcoxon 秩和检验. 检验两组样本之间是否存在显著性差异，若两组样本之间存在显著性差异，则比较两组样本的平均最优值，以确定两组样本的优劣. "−"、"+"和"≈"分别代表相应算法的性能劣于、优于和近似于(μ_w,λ)-Cholesky-CMA 算法.

从表 2 的实验结果可以看出，本文改进的变体(μ_w,λ)-Cholesky-CMA 的性能明显优于其它变体，证明了秩-μ 更新对算法性能的重要性. 因为在每一代都进行协方差矩阵更新，从而保证了协方差矩阵的准确性，所以(μ_w,λ)-Cholesky-CMA 的结果比(μ_w,λ)-PUd-CMA 好。各种变体的具体表现如下：

$(1+1)$-Cholesky-CMA：在单峰和部分可分问题上表现较好，在多峰或有噪声的函数上表现较差，其中在有噪声函数上表现最差. 这与精英选择机制与关，精英选择机制在有噪声的情况下容易造成系统性高估[15,27]，不适用于求解有噪声问题.

$(1+\lambda)$-Cholesky-CMA：整体表现不好，$1+\lambda$ 精英选择策略并没有给该算法带来大的优势，选择单一个体作为父个体相当于取消了秩-μ 更新，不能充分利用种群信息引导搜索. 另外，单一个体作为父个体也增大了演化路径的随机性，导致协方差矩阵更新的波动性加大.

(μ_w,λ)-Cholesky-CMA：在整体性能上表见最好，尤其在多峰、不可分、高条件数和有噪声函数上表现较好，但在简单可分或部分可分的单峰问题上表现差于(μ_w,λ)-sep-CMA 或者$(1+1)$-Cho-CMA.

表 1 测试函数

函数	函数名称	维度
F_1	可分的单峰 Sphere 函数	200
F_2	旋转高条件数的 Elliptic 函数	200
F_3	有噪声的 Schwefel 函数	30
F_4	偏移的 Rosenbrock 函数	200
F_5	偏移旋转的 Rastrigin 函数	30
F_6	多峰可分复合函数	10
F_7	可分高条件数 Elliptic 函数	200
F_8	可分多峰 Rastrigin 函数	500
F_9	部分可分 Elliptic 函数	1000
F_{10}	完全不可分 Schwefel 函数	200

表 2 独立运行 25 次求得的平均最优值及标准偏差

函数	$(1+1)$-Cho-CMA	$(1+\lambda)$-Cho-CMA	(μ_w,λ)-Cho-CMA	(μ_w,λ)-PUd-CMA	(μ_w,λ)-sep-CMA
F_1	**−4.50e+2 ±0.00e+0**$^{\approx}$	**−4.50e+2 ±0.00e+0**$^{\approx}$	**−4.50e+2 ±0.00e+0**	**−4.50e+2 ±0.00e+0**$^{\approx}$	**−4.50e+2 ±0.00e+0**$^{\approx}$
F_2	8.96e+2 ±3.76e+3$^-$	2.78e+4 ±1.73e+4$^-$	**−4.50e+2 ±0.00e+0**	−4.48e+2 ±6.95e−1$^-$	1.38e+6 ±2.89e+5$^-$
F_3	9.34e+4 ±2.76e+4$^-$	2.52e+4 ±8.51e+3$^-$	**7.33e+1 ±7.33e+2**	2.01e+3 ±5.05e+3$^-$	1.62e+4 ±1.19e+4$^-$
F_4	5.16e+2 ±4.33e+1$^-$	5.75e+2 ±5.51e+1$^-$	4.78e+2 ±1.56e+1	4.81e+2 ±6.49e+0$^{\approx}$	**4.11e+2 ±1.99e+1**$^+$
F_5	−8.64e+1 ±7.15e+1$^-$	−1.96e+2 ±3.45e+1$^-$	**−2.83e+2 ±1.12e+1**	−2.82e+2 ±1.22e+1$^{\approx}$	−2.78e+2 ±1.45e+1$^{\approx}$
F_6	6.66e+2 ±1.42e+2$^-$	5.60e+2 ±2.13e+2$^-$	**4.52e+2 ±1.81e+2**	4.57e+2 ±1.33e+2$^{\approx}$	5.16e+2 ±2.38e+2$^{\approx}$
F_7	1.65e+0 ±1.22e+0$^+$	4.38e+4 ±1.38e+4$^-$	1.67e+1 ±2.38e+1	5.45e+1 ±1.70e+2$^{\approx}$	**9.68e−9 ±2.99e−10**$^+$
F_8	1.07e+4 ±8.38e+2$^-$	7.41e+3 ±4.62e+2$^-$	**1.83e+3 ±9.42e+1**	1.93e+3 ±1.20e+2$^-$	2.00e+3 ±1.66e+2$^-$
F_9	**6.64e+8 ±5.22e+7**$^+$	6.18e+9 ±6.33e+8$^+$	1.57e+10 ±1.93e+9	2.32e+10 ±2.18e+9$^-$	9.95e+8 ±2.86e+8$^+$
F_{10}	1.78e+5 ±7.47e+4$^-$	2.03e+5 ±4.45e+4$^-$	**7.59e−4 ±1.20e−3**	2.50e−3 ±3.80e−3$^-$	1.64e+3 ±1.49e+3$^-$

(μ_w,λ)-PUd-CMA:在整体性能上劣于(μ_w,λ)-Cholesky-CMA-ES算法,但是它们在5个问题上性能是相似的.其原因是两个算法的大体机制是相同的,区别在于(μ_w,λ)-PUd-CMA的推迟更新机制使得协方差矩阵不能及时得到更新,导致新生成的搜索分布可能用到的是旧的协方差矩阵信息.另外,(μ_w,λ)-PUd-CMA采用特征分解的方式进行协方差矩阵更新,该方法的数值精度要高于Cholesky因式分解的方法.

(μ_w,λ)-sep-CMA:整体上在多峰、不可分、有噪声的函数上表现较差,在单峰、可分的问题上表现较好,这是因为以对角阵代替常规协方差矩阵导致失去了对搜索空间的旋转不变性.这种简化了的协方差矩阵更新方法使该算法在运行时间上有显著优势,尤其在高维问题上甚至达到了1个或几个数量级的优势.所以用该算法求解高维的单峰可分问题是一个较好的选择,该算法甚至在高维多峰的Rosenbrock函数上也取得了最好结果.

7 讨论及展望

CMA学习机制是一种基于统计的自适应学习方法,它通过对演化过程中所构造的概率分布进行随机抽样来生成新的种群个体,优化过程中生成的概率分布刻画了被优化目标函数的特性.这种机制与分布估计算法[28,29](Estimation of Distribution Algorithms, EDAs)的搜索方式存在着很大的相似性,它们都是根据先前代有潜在希望的个体构造的概率分布模型来产生下一代的新个体.区别是CMA-ES通过统计的方式来确定搜索参数(协方差矩阵和全局步长),搜索参数确定了新个体变异的方向和强度;而EDAs是通过统计的方式来建立概率采样模型,利用概率模型在搜索空间中采样生成新个体.

CMA-ES被认为是实值优化中最具竞争力的演化算法之一,对于非线性优化问题,由于传统的优化方法,如拟牛顿法(BFGS)、共轭梯度下降法等,在非凸和崎岖不平的搜索景观上(如尖峰、不连续、异常值、噪声和局部最优值)会导致搜索失败,所以CMA-ES在这类问题上是一种非常具有吸引力的选择.CMA-ES算法具有以下优点:

(1)CMA-ES对于求解具有高条件数的凸二次函数有内在的优势.

(2)CMA-ES具有对搜索空间任意可逆线性变换的不变性,对处理平移的、旋转的和严重不等比例缩放的问题也具有内在的优势.

(3)目标参数及策略参数变异的无偏性是随机搜索算法的一个重要设计准则[30],CMA-ES算法的种群均值$\boldsymbol{m}$、协方差矩阵$\boldsymbol{C}$和全局步长的对数$\ln(\sigma)$都是无偏的,这可以提高算法的稳定性,减少种群的发散或早熟.

(4)CMA-ES的种群大小是内生的,与问题的维度具有较小的相关性,不会随问题维度的增长而导致种群大小急剧增加.

CMA-ES作为一种局部搜索算法,其未来研究方向或许可从以下5点进行:

(1)对于Cholesky因式分解方法,在非精英选择的CMA-ES算法中,计算共轭演化路径$\boldsymbol{p}_\sigma$的公式有轻微改变.一般来说Cholesky因子$\boldsymbol{A}$是不对称的,Cholesky因式分解方法无法得到正交矩阵$\boldsymbol{B}$,所以在计算$\boldsymbol{p}_\sigma$的公式中取消了$\boldsymbol{B}$.如果能够把Cholesky因子$\boldsymbol{A}$分解成对称的形式$\boldsymbol{A}=\boldsymbol{BDB}^{\mathrm{T}}$,则上述问题可通过$\sigma^{-1}\boldsymbol{A}_{\mathrm{inv}}(\langle\boldsymbol{x}\rangle_w^{(g+1)}-\langle\boldsymbol{x}\rangle_w^{(g)})$替换$\boldsymbol{B}\langle z\rangle_w$来解决[15].但是如何获得一个对称的Cholesky因式分解仍是一个需解决的问题.

(2)(μ_w,λ)-Cholesky-CMA-ES算法虽然在一定程度上降低了时间复杂度.但是在大规模优化问题上,如何降低其时间和空间复杂度仍是一个需要解决的问题.

(3)在CMA-ES算法中由于协方差矩阵自适应和全局步长自适应是相互独立进行的,不适当的全局步长值在多峰函数或有噪声函数上容易造成种群发散或早熟,这是需要解决的问题.

(4)CMA-ES是一种局部优化算法,在全局搜索能力上较弱,目前一般通过增大种群或增大全局搜索步长的方式来增强全局搜索能力.寻找能够有效增强CMA-ES全局搜索能力的新方法也是今后研究的一个方向.

(5)基于CMA-ES的特点,探索CMA-ES的新的应用领域,把它应用到具体的实际问题中去是今后的一个重点研究方向.

参考文献

[1] Bäck T, Foussette C, Krause P. Contemporary Evolution Strategies[M]. Berlin Heidelberg: Springer, 2015. 7-86.

[2] Das S, Suganthan P N. Differential evolution: a survey of the state-of-the-art[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.

[3] 彭虎,吴志健,等.基于精英区域学习的动态差分进化算法[J].电子学报,2014,42(8):1522-1530.
Peng Hu, Wu Zhi-jian, et al. Dynamic differential evolution algorithm based on elite local learning[J]. Acta Electronica Sinica, 2014, 42(8): 1522- 1530. (in Chinese)

[4] 周新宇,吴志健,等.一种精英反向学习的粒子群优化算法[J].电子学报,2013,41(8):1647-1652.
Zhou Xin-yu, Wu Zhi-jian, et al. Elite opposition-based

particle swarm optimization[J]. Acta Electronica Sinica, 2013,41(8):1647 – 1652.(in Chinese)

[5] 喻飞,李元香,等.透镜成像反学习策略在粒子群算法中的应用[J].电子学报,2014,42(2):230 – 235.
Yu Fei, Li Yuan-xiang, et al. The application of a novel OBL based on lens imaging principle in PSO[J]. Acta Electronica Sinica,2014,42(2):230-235.(in Chinese)

[6] Schwefel H. Numerical Optimization of Computer Models [M]. New York:John Wiley & Sons Inc,1981.1 – 389.

[7] Ostermeier A, Gawelczyk A, Hansen N. A derandomized approach to self-adaptation of evolution strategies[J]. Evolutionary Computation,1994,2(4):369 – 380.

[8] Hansen N, Ostermeier A. Adapting arbitrary normal mutation distributions in evolution strategies:the covariance matrix adaptation[A]. Proc of IEEE Conference on Evolutionary Computation[C]. IEEE,1996.312 – 317.

[9] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies[J]. Evolutionary Computation,2001,9(2):159 – 195.

[10] Hansen N, Muller S D, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)[J]. Evolutionary Computation,2003,11(1):1 – 18.

[11] Müller S D, Hansen N, et al. Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations[A]. Parallel Problem Solving from Nature—PPSN VII[C]. Granada:Springer,2002.422 – 431.

[12] Igel C, Suttorp T, Hansen N. A computational efficient covariance matrix update and a (1 + 1)-CMA for evolution strategies[A]. Proc of the 8th Annual Conference on Genetic and Evolutionary Computation[C]. Washington: ACM,2006.453 – 460.

[13] Grewal M, Andrews A. Kalman Filtering:Theory and Practice Using MATLAB[M]. New York:John Wiley & Sons Inc,2001.25 – 347.

[14] Igel C, Hansen N, Roth S. Covariance matrix adaptation for multi-objective optimization[J]. Evolutionary Computation,2007,15(1):1 – 28.

[15] Suttorp T, Hansen N, Igel C. Efficient covariance matrix update for variable metric evolution strategies[J]. Machine Learning,2009,75(2):167 – 197.

[16] Ros R, Hansen N. A simple modification in CMA-ES achieving linear time and space complexity[A]. Parallel Problem Solving from Nature-PPSN X[C]. Berlin Heidelberg:Springer,2008.296 – 305.

[17] Chen L, Zheng Z, et al. An evolutionary algorithm based on covariance matrix learning and searching preference for solving CEC 2014 benchmark problems[A]. Proc of IEEE Congress on Evolutionary Computation[C]. Beijing:IEEE,2014.2672 – 2677.

[18] Ghosh S, Das S, Roy S, et al. A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization[J]. Information Sciences,2012,182(1): 199 – 219.

[19] Preuss M. Niching the CMA-ES via nearest-better clustering[A]. Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation [C]. Portland:ACM,2010.1711 – 1718.

[20] Li L X, Tang K. History-based topological speciation for multimodal optimization[J]. IEEE Transactions on Evolutionary Computation,2015,19(1):136 – 150.

[21] 杨咚咚,焦李成,等.求解偏好多目标优化的克隆选择算法[J].软件学报,2010,21(01):14 – 33.
Yang Dong-dong, Jiao Li-cheng, et al. Clone selection algorithm to solve preference multi-objective optimization [J]. Journal of Software,2010,21(01):14 – 33.(in Chinese)

[22] Loshchilov I. A computationally efficient limited memory CMA-ES for large scale optimization[A]. Proc of the 2014 Conference on Genetic and Evolutionary Computation[C]. Vancouver:ACM,2014.397 – 404.

[23] Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution[J]. Information Sciences,2008,178(15):2985 – 2999.

[24] Hansen N. The CMA Evolution Strategy:A Tutorial[Z]. 2011.1 – 34

[25] Suganthan P N, Hansen N, et al. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization [R]. Singapore: Nanyang Technological University,2005.1 – 50.

[26] Li X, Tang K, et al. Benchmark Functions for the CEC′ 2013 Special Session and Competition on Large-Scale Global Optimization[R]. Cancun:IEEE,2013.1 – 23.

[27] Arnold D V. Noisy Optimization with Evolution Strategies [Z]. Boston:Kluwer Academic Publishers,2002.1 – 20.

[28] Chen T, Tang K, Chen G, et al. Analysis of computational time of simple estimation of distribution algorithms[J]. IEEE Transactions on Evolutionary Computation,2010,14 (1):1 – 22.

[29] Yang P, Tang K, Lu X. Improving estimation of distribution algorithm on multimodal problems by detecting promising areas[J]. IEEE Transactions on Cybernetics,2015, 45(8):1438 – 1449.

[30] Beyer H, Deb K. On self-adaptive features in real-parameter evolutionary algorithms[J]. IEEE Transactions on Evolutionary Computation,2001,5(3):250 – 270.

作者简介

李焕哲　男,1975 年生,河北唐山人,武汉大学计算机学院博士研究生,研究方向:智能计算、机器学习.
E-mail:lihuanzhe@ whu. edu. cn

吴志健　男,1963 年生,教授,博士生导师,武汉大学软件工程国家重点实验室副主任,研究方向:智能计算、并行计算和智能信息处理.
E-mail:zhijianwu@ whu. edu. cn

汪慎文　男,1979 年生,副教授,博士后,研究方向为智能计算与机器学习等.
E-mail:wangshenwen@ whu. edu. cn

郭肇禄　男,1984 年生,江西南康人,博士,研究方向为智能计算、并行计算和机器学习.
E-mail:gzl@ whu. edu. cn