

文章编号: 1001-0920(2012)04-0519-06

## 引入补偿采样和平滑算子的概率集群优化算法

王楠, 胡天江, 陈璟, 沈林成

(国防科学技术大学 机电工程与自动化学院, 长沙 410073)

**摘要:** 探讨一种新颖的分布式多智能体优化算法及其在求解约束优化问题上的实现方式. 通过分析该算法在有限采样数下寻优能力所受到的影响, 提出在迭代优化过程中引入补偿采样机制和平滑算子的改进方案, 在保留原算法优点的基础上提高了采样过程对决策空间的覆盖度, 增强了方法的全局和局部邻域搜索能力. 实验结果表明, 引入补偿采样和平滑算子后的概率集群优化算法在收敛速度、解质量和稳定性等方面均得到了明显改善.

**关键词:** 多智能体系统; 混合决策变量; 概率集群; 补偿采样; 平滑算子

中图分类号: TP301.6

文献标识码: A

## Probability collectives with compensate sampling and smooth factors

WANG Nan, HU Tian-jiang, CHEN Jing, SHEN Lin-cheng

(College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, China. Correspondent: WANG Nan, E-mail: acnwang@yahoo.com.cn)

**Abstract:** This paper explores a novel distributed multi-agent optimization algorithm for solving constrained optimization problems. The performance of the algorithm is analyzed and the search capability is found to be influenced by the limitation of the sample number during iteration. Therefore, an improved algorithm is proposed with compensate sampling technique and smooth factors, which maintains the benefits of the original algorithm while improves its global and local search capability. Experimental results show that the proposed algorithm achieves obvious improvement in convergence speed, solution quality and long term stability.

**Key words:** multi-agent system; mixed decision variables; probability collectives; compensation sampling; smooth factor

### 1 引言

复杂系统的决策与优化问题在理论研究和工程实践上均受到了广泛而持久的关注, 各种智能决策及优化手段日益得到研究和应用. 长期以来, 对于复杂系统的优化方法大都以集中式为主, 然而随着系统规模和复杂度的增加, 分布式优化已逐渐成为研究的热点之一. 该类方法将复杂系统分解为易于处理的子系统, 通过对子系统进行分布式优化达到系统整体的最优. 起源于分布式人工智能的多智能体系统(MAS)是典型的分布式优化方法之一, 其思想是将问题的每个子系统作为一个智能体, 通过智能体间的协作与博弈实现对系统整体的优化. 随着多智能体研究的深入, 人们提出了集群智能(COIN)<sup>[1]</sup>理论, 并逐步发展形成了概率集群(PC)<sup>[2]</sup>优化算法, 为基于MAS的优化技术研究开辟了新的研究领域. 概率集群算法以决

策变量的概率分布而非策略值为主要优化对象, 能较好地处理连续、离散及混合型决策变量, 并在收敛速度、跳出局部极值能力和稳定性等方面具有良好表现<sup>[3]</sup>. 文献[4]对该算法的优点进行了较全面的阐述. 此外, 该算法的运行参数较少、易于实现, 是一种具有较强研究与应用价值的分布式优化算法.

目前, 国内外关于PC算法的应用研究尚比较少见. 为此, 本文探讨了利用PC算法求解约束优化问题的具体实现方式, 并分析了采样数的限制对算法的影响. 指出在问题规模较大时, 如果采样数不足够大, 则将降低采样过程对决策空间的覆盖度, 导致收敛缓慢或收敛不到均衡解; 而增大采样数, 又将增加迭代过程的时间开销. 针对该矛盾, 本文对PC算法的采样及更新策略进行了改进, 引入了补偿采样机制和平滑算子, 分别增强了算法的全局搜索能力和局部邻域搜索

收稿日期: 2010-11-07; 修回日期: 2011-01-23.

基金项目: 国家自然科学基金项目(60805037, 61005077).

作者简介: 王楠(1980-), 男, 博士生, 从事优化理论与应用的研究; 沈林成(1965-), 男, 教授, 博士生导师, 从事飞行器导航与任务规划、智能控制等研究.

能力. 经实验验证, 改进型 PC 算法在解决约束及无约束函数优化问题上能够有效克服原 PC 算法的缺陷, 大幅提升优化性能.

## 2 概率集群理论基础

集群智能理论是概率集群的基础和前身. 一个集群由一组具备学习能力的智能体组成, 用  $\{\eta_1, \eta_2, \dots, \eta_N\}$  表示. 每个智能体具有一个候选策略集  $\xi_i$  和个体效益函数  $g_i$ . 智能体的整体效益用  $G$  表示, 与系统描述实际问题的优化指标相对应. 不失一般性, 本文均以最小化  $G$  为指标, 智能体通过博弈优化  $g_i$  并达到均衡. 传统博弈论已经证明, 在  $g_i$  与  $G$  相一致的前提下, 博弈的均衡所对应的  $G$  值至少是原问题的局部最优指标<sup>[1]</sup>. 在集群智能理论框架的基础上, 概率集群结合机器学习、统计物理学和优化领域的知识, 建立了较为完整的数学框架, 以对智能体策略的概率分布进行优化. 在概率的形式下, 令  $q_i(\xi_i)$  表示  $\eta_i$  策略的概率分布, 则  $\eta_i$  的个体效益期望表达如下:

$$E(g_i) = \int d\xi \prod_j q_j(\xi_j) g_i(\xi). \quad (1)$$

若智能体是完全理性的, 则每个智能体可以计算出其他智能体的混合策略, 并根据式 (1) 决定自身最优的混合策略, 博弈的均衡是纳什均衡. 然而, 在绝大多数情况下完全理性的假设不成立, 此时智能体策略的选取还需考虑其所掌握的信息量的多少, 该信息量可以用香农的信息熵衡量, 如下式所示:

$$S(q) = - \int d\xi q(\xi) \ln q(\xi), \quad (2)$$

其中负号表示  $S(q)$  代表  $q(\xi)$  中的信息量. 熵越大, 信息量越少,  $\xi$  的分布越不确定 (例如均匀分布). 概率集群算法指出<sup>[5]</sup>, 在不完全理性条件下, 博弈的均衡是关于智能体策略的概率分布和信息熵的拉格朗日最优解, 定义为极大熵拉格朗日式, 即

$$\min L(q, T) = E_q(G) - T \sum_i S(q_i). \quad (3)$$

其中: 等号右边第 1 项表示在智能体联合策略下的整体期望效益; 第 2 项为智能体的总信息熵,  $T$  为权重系数, 又称为温度系数. 可见, 概率集群算法的显著特点之一是以策略的概率分布为优化对象, 与策略的具体类型 (连续、离散、混合) 无关, 因此在处理优化问题时具有较强的适应能力. 概率集群算法的另一特点是将式 (3) 分解为每个智能体的子问题进行分布式优化. 其中, 智能体  $\eta_i$  所对应的子问题表示如下:

$$\begin{aligned} \min L_i(q, T) &= E_q(g_i) - TS(q_i), \\ \text{s.t.} \quad &\int d\xi_i q(\xi_i) = 1. \end{aligned} \quad (4)$$

根据统计物理学可推导出上式的解  $q_i$  为如下形式的波尔兹曼分布<sup>[5]</sup>:

$$q_i(\xi_i) \propto e^{-(1/T)E_{q_i}(g_i|\xi_i)}. \quad (5)$$

目前, 对该分布尚无法以解析的形式给出最优解, 取而代之的是以迭代优化的方式对其进行逼近. 文献<sup>[2]</sup>列举了多种迭代优化方法, 本文仅对其中采纳较多的牛顿迭代更新法作简单介绍. 在该方法下, 对式 (4) 的求解采用如下形式迭代进行:

$$\begin{aligned} q_i(\xi_{ij}) &\leftarrow q_i(\xi_{ij}) - \Delta q_{ij}, \\ \Delta q_{ij} &= \\ &\alpha q_i(\xi_{ij}) \times \left\{ \frac{E[g_i|\xi_{ij}] - E[g_i]}{T} + S(q_i) + \ln q_i(\xi_{ij}) \right\}. \end{aligned} \quad (6)$$

其中:  $\alpha$  为牛顿迭代步长,  $\xi_{ij}$  为  $\eta_i$  的第  $j$  个可选策略. 与集群智能不同, 概率集群算法在每次迭代过程中对智能体的全体策略的概率密度都进行更新, 而限于被选取的策略. 式 (6) 可保证智能体的策略分布朝最小化  $L_i$  的方向演化, 但无法确保迭代过程中智能体策略的概率密度总和为 1 以及非负概率的出现. 因此, 需要在每次迭代更新之后对智能体策略的概率密度进行修正, 一般可将负概率设为一极小值, 如  $10^{-4}$ , 并进行归一化处理. 通过分布式迭代与更新, 智能体的策略最终达到均衡. 此时, 各智能体概率密度最大的策略即为原问题的解.

## 3 概率集群算法的实现及分析

考虑如下形式的约束优化问题:

$$\begin{aligned} \min_{\vec{x}} G(\vec{x}); \\ \text{s.t.} \quad c_j(\vec{x}) \leq 0. \end{aligned} \quad (7)$$

其中  $x$  为问题的决策变量集. 应用概率集群算法求解该问题时, 每个智能体  $\eta_i$  代表问题的一个决策变量  $x_i$ , 智能体的策略集  $\xi_i$  即为  $x_i$  的取值空间  $X_i$ . 由  $X_i$  到  $\xi_i$  的映射可以是连续的, 也可以是离散的, 由  $\xi_i$  的类型决定. 在一般的应用中, 可以将  $\xi_i$  取为  $X_i$  上的离散点集, 离散化密度则根据具体问题而定. 智能体仅具有单一的个体效益值, 因此, 需将上述约束优化问题转化为如下形式的无约束优化问题:

$$\min_{\vec{x}} G(\vec{x}) + \sum_j \lambda_j c_j(\vec{x}), \quad (8)$$

其中  $\lambda_j$  为拉格朗日乘子. 智能体的个体效益  $g_i$  可采用“团队效益”(TG)和“完美人生”(WLU)等形式<sup>[6]</sup>, 具体实现可参阅相关文献. 博弈开始之前, 每个智能体要为其策略  $\xi_i$  设定一个初始分布, 在对问题缺乏先验知识的情况下, 可设初始分布为均匀分布. 博弈是以智能体分布式地迭代更新其策略的概率分布的方式进行. 在每次迭代过程中, 每个智能体首先按其策略的概率分布进行联合采样, 得到智能体的联合策略集, 记为  $C$ ; 然后, 智能体  $\eta_i$  根据集合  $C$  评估其在每个策略值  $\xi_{ij}$  下的个体效益. 由于每轮迭代过程中的采

样数有限, 可将历史评估结果纳入评估过程. 此时, 智能体在其策略  $\xi_{ij}$  下的个体效益期望值计算如下:

$$E(g_i|\xi'_i = \xi_{ij}) = \frac{N_{ij}^{(k)}}{D_{ij}^{(k)}} = \frac{\sum_m g_{ij}(\xi'_i = \xi_{ij}, \xi'_{(i)}) \delta(\xi'_i - \xi_{ij}) + \kappa N_{ij}^{k-1}}{\sum_m \delta(\xi'_i - \xi_{ij}) + \kappa D_{ij}^{k-1}}. \quad (9)$$

其中:  $\xi'_i$  为  $\eta_i$  的策略值,  $\xi'_{(i)}$  为除  $\eta_i$  外其余智能体的策略值;  $m$  为采样数;  $\delta(\cdot)$  为狄拉克函数;  $\kappa$  为历史评估结果的权重因子. 根据个体效益期望, 智能体分别按式 (6) 更新其策略的概率分布, 同时根据计算得到的  $c_j$  值, 按下式更新相应的拉格朗日乘子:

$$\lambda_j \leftarrow \lambda_j + \beta \times \frac{\sum c_j(\xi)}{m}. \quad (10)$$

其中  $\beta$  为迭代步长.

采用概率集群算法求解约束优化问题的具体流程如图 1 所示.

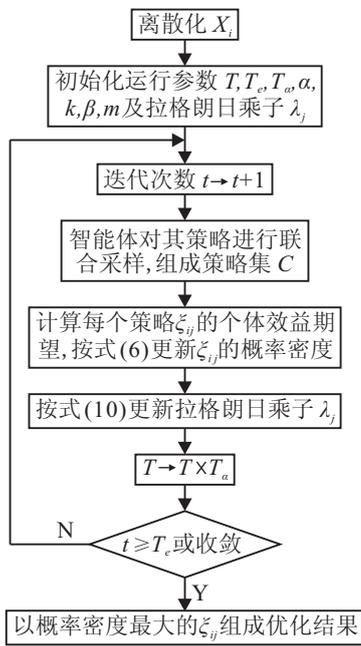


图 1 概率集群优化算法运行流程

下面分析优化过程. 博弈开始时, 温度系数  $T$  较高, 根据式 (5) 可知, 此时  $\ln q_i(\xi_{ij})$  对  $\Delta q_{ij}$  起主要影响. 由于  $\ln q_i$  在  $(0, 1]$  上是关于  $q_i$  的增函数,  $q_i(\xi_{ij})$  越小,  $-\Delta q_{ij}$  越大, 即概率小的策略概率密度增幅较大, 此时智能体将尽可能地探索未选择的策略空间. 随着迭代次数  $t$  的增加,  $T$  逐步降低,  $E(g_i|\xi_{ij}) - E(g_i)$  开始起主导作用, 此时智能体将侧重于选择最优化  $g_i$  的策略. 虽然在理论上当  $g_i$  与  $G$  一致时博弈的均衡至少是问题的局部最优解, 但这是建立在对  $E(g_i|\xi_{ij})$  的无偏估计的基础上的. 由于在实际优化过程中, 智能体采用基于概率的随机采样方式对策略空间进行搜索, 并

受采样数  $m$  的限制, 若智能体的最优或局部最优联合策略  $\xi^*$  的某一分量  $\xi_j^*$  在  $T$  较高时未被命中或命中时的评估效益值较差, 则随着  $T$  的降低, 其被命中的概率也逐步降低. 如果在其被命中前算法已收敛, 则博弈的均衡实际上并未达到, 所得结果并不是局部最优解. 通过分析概率集群方法的原理, 造成这一问题的原因主要有以下两方面:

1) 与采样数  $m$ , 温度系数  $T$  及衰减因子  $T_\alpha$  的取值有关. 取值较小, 则将导致优化过程对决策空间的覆盖度低, 从而错过潜在的最优策略. 增大  $m, T$  或  $T_\alpha$  可以在一定程度上解决该问题, 但由于决策空间随决策量数目以指数的形式增长, 在决策量较多时仍无法涵盖各种可能的情况, 并将导致迭代时间长、收敛缓慢等新问题的出现.

2) 主要与策略概率密度的更新过程有关. 由式 (5) 可以看出, 智能体的每个  $\xi_{ij}$  的概率密度均独立更新, 相互间几乎没有影响, 因此该方法不具备邻域搜索能力, 即使找到了最优解的相邻区域, 其搜索到最优解的概率与在全局范围内搜索到最优解的概率并无明显差别.

## 4 改进概率集群优化算法

基于上述分析, 本文对概率集群优化算法进行两方面改进, 在迭代博弈过程分别引入补偿采样机制和平滑算子, 以提高算法的全局和局部邻域搜索能力.

### 4.1 补偿采样机制

补偿采样是在迭代过程中增加额外的样本参与评估机制. 引入补偿采样的目的是对博弈初期未命中或评估效益较差的策略空间进行探索, 以克服采样数的限制, 使概率密度较小的策略值也能被采样, 在概率意义上对决策空间形成覆盖. 补偿采样过程首先建立策略空间命中次数的统计信息, 以智能体  $\eta_i$  为例, 用  $M_{ij}$  表示其策略  $\xi_{ij}$  在迭代过程的命中次数, 即

$$M_{ij} = \sum_t \left( \sum_m \delta(\xi'_i - \xi_{ij}) + \sum_{m_e \times N} \delta(\xi'_i - \xi_{ij}) \right). \quad (11)$$

其中:  $t$  为迭代次数,  $m_e$  为每个智能体的补偿采样数. 在每次迭代过程中,  $\eta_i$  根据  $M_{ij}$  按

$$q_{ij}^e = \frac{\max(0, M - M_{ij})}{\sum_j \max(0, M - M_{ij})}, \quad M = \frac{\sum_j M_{ij}}{\text{size}(\xi_i)} \quad (12)$$

计算其每个策略  $\xi_{ij}$  的补偿采样概率, 得到补偿采样分布  $q_i^e$ . 在该分布中, 命中次数大于平均值的策略, 其补偿采样概率为 0. 然后,  $\eta_i$  按该分布进行采样, 得到  $m_e$  个补偿采样策略. 同时, 其余智能体  $\eta_{(i)}$  根据相应的  $q_{(i)}$  进行联合采样, 得到  $m_e$  组补充采样策略, 补

偿采样策略与补充采样策略的组合即为  $\eta_i$  的联合补偿采样策略. 各智能体的全部联合补偿采样策略便组成了联合补偿采样策略集, 记为  $C^e$ .

由于补偿采样在形式上违背了概率集群算法的采样原则, 对于  $C^e$  的评估结果, 若直接参与计算个体期望效益, 则可能对  $E(g_i)$  的估计形成较大扰动. 事实上, 引入补偿采样机制的出发点是探索潜在的最优策略, 只有当补偿采样策略的个体效益评估值优于由  $C$  计算得到的个体效益期望时, 补偿采样才有意义. 因此, 对  $C^e$  中的每组策略  $\xi^e$ , 按下式更新其相关策略值的期望评估效益:

$$E'(g_i|\xi_{ij}) = \begin{cases} \frac{N_{ij}^k + g_i(\xi_i^e = \xi_{ij}, \xi_{(i)}^e)}{D_{ij}^k + 1}, \\ g_i(\xi_i^e = \xi_{ij}, \xi_{(i)}^e) < E(g_i|\xi_{ij}); \\ E(g_i|\xi_{ij}), \text{ else.} \end{cases} \quad (13)$$

其中:  $\xi_i^e$  为  $\xi^e$  中与  $\eta_i$  对应的策略,  $\xi_{(i)}^e$  为  $\xi^e$  中除  $\eta_i$  外其他智能体的策略. 由式 (13) 可知, 若  $C^e$  中存在较优的策略, 则该策略的个体效益期望将朝较优的方向调整, 从而减小该策略的概率密度变化量  $\Delta q$ , 使该策略在下一轮迭代中能够以更高的概率被采样. 该机制有利于避免算法过早陷入局部最优, 增强算法的全局搜索能力.

## 4.2 平滑算子

平滑算子是在智能体更新其策略  $\xi_{ij}$  的概率密度  $q_i(\xi_{ij})$  的过程中, 将概率增幅  $-\Delta q_{ij}$  平滑到  $\xi_{ij}$  的邻域, 使  $q_i(\xi_{ij})$  增加的同时, 与  $\xi_{ij}$  相邻的策略  $\xi_i^l$  的概率密度  $q_i(\xi_i^l)$  也相应增加, 从而在概率意义上覆盖较优解的邻域, 增强算法的邻域搜索能力. 平滑过程在智能体  $\eta_i$  按式 (6) 对其策略的概率分布更新完毕后进行. 令  $r$  表示邻域半径, 则策略  $\xi_{ij}$  的平滑算子可表示为

$$\Delta q_{ij}^l(\xi_i^l = \xi_{ik}) = \min(\Delta q_i(\xi_{ij}) \times \mu^{-\|\xi_{ij} - \xi_{ik}\|^2/r}, 0). \quad (14)$$

其中:  $\Delta q_{ij}^l(\xi_i^l)$  为  $\Delta q_{ij}$  在相邻策略  $\xi_i^l$  上的平滑,  $\mu$  为平滑系数,  $\|\cdot\|^2$  为欧几里德距离. 在智能体的策略被等间隔离散化的情况下, 平滑算子可简化为如下形式:

$$\Delta q_{ij}^l(\xi_i^l = \xi_{ik}) = \min(\Delta q_i(\xi_{ij}) \times \mu_d^{|j-k|}, 0), \quad (15)$$

其中  $\mu_d$  为平滑衰减乘子. 智能体  $\eta_i$  的所有策略在  $\xi_{ik}$  上的平滑效应按下式计算:

$$q_i'(\xi_{ik}) \leftarrow q_i'(\xi_{ik}) - \Delta q_i^l(\xi_{ik}), \\ \Delta q_i^l(\xi_{ik}) = \sum_{\|\xi_{ij} - \xi_{ik}\|^2 < r} \Delta q_{ij}^l(\xi_{ik}) \sum_{|j-k| < d} \Delta q_{ij}^{ld}(\xi_{ik}). \quad (16)$$

其中:  $q_i'(\xi_{ik})$  表示按式 (6) 更新后  $\xi_{ik}$  的概率密度,  $d$  为邻域搜索步长. 可见, 引入平滑算子并没有削弱

$-\Delta q_{ij}$ , 因此并没有违背概率集群算法的基本原理. 但由于平滑算子增加了  $q_i(\xi_i^l)$ , 在总体上会降低  $-\Delta q_{ij}$  为  $\xi_{ij}$  带来的领先优势, 从而在表面上会对收敛过程造成一定的延缓. 然而, 引入平滑算子可为概率集群算法带来前所未有的邻域搜索能力, 有利于在较少的迭代次数内从较优解出发找到最优解. 在实际应用和测试中, 引入补偿机制和平滑算子的改进型概率集群优化算法, 在收敛速度、解的质量和稳定性等方面均得到了明显改善.

## 4.3 运行流程

引入补偿采样机制和平滑算子后, 概率集群优化算法的搜索能力和收敛性等均得到了明显改善. 下面给出改进型概率集群算法的运行流程, 如图 2 所示.

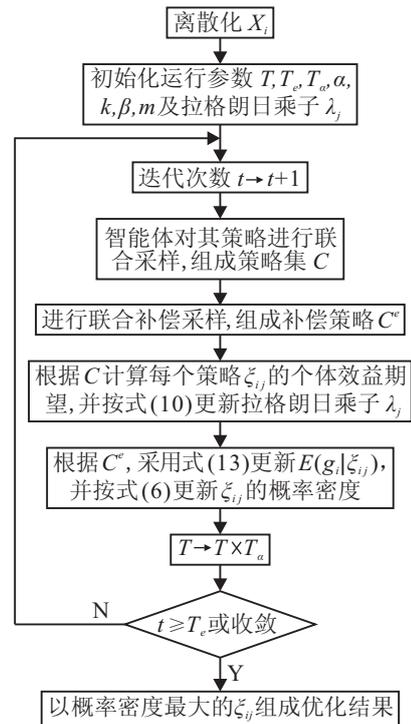


图 2 改进型概率集群优化算法运行流程

## 5 实验与分析

为验证改进型概率集群优化算法的有效性, 本文在无约束及约束优化 Benchmark 问题中选择了多个具有不同特点的实例进行仿真测试, 包括 Schaffer  $F_7$  函数, Rosenbrock 函数, Ackley 函数和 Shankar 问题. 仿真实验环境基于 Matlab 2008 a 构建, 实验过程中对标准 PC, 改进型 PC 以及遗传算法 (GA) 进行了比较. 为真实反映方法间的性能差异, 以优化过程的函数评估次数而非迭代次数作为基准量, 比较在相同函数评估次数下各方法所找到的历史最优解质量. 每组实验重复 50 次, 采用 best-so-far 图表示比较结果, 其中曲线上的点表示 50 次实验的平均历史最优指标值, 垂直条带则表示最优指标的 95% 置信区间. 各算法的主要参数设置如表 1 所示.

表1 PC, 改进PC和GA的主要参数

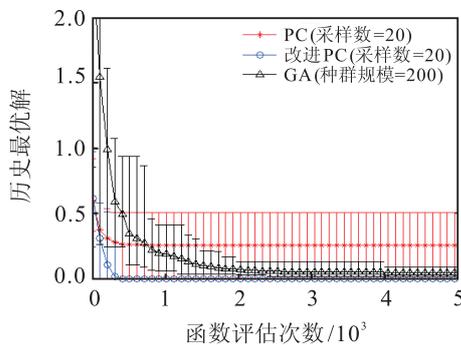
算法	参数名	参数值
PC, 改进PC	$T$	0.1
	$T_\alpha$	0.95
	$\kappa$	0.5
	$\alpha$	0.2
改进PC	$m_e$	5
	$d$	5
	$\mu_d$	0.4
GA	种群规模	200
	变异概率	0.2
	交叉概率	0.9
	精英个体	10

### 5.1 Schaffer 函数

Schaffer  $F_7$  函数<sup>[3]</sup>定义如下:

$$F_1(X) = (x_1^2 + x_2^2)^{0.25} \times [\sin^2(50((x_1^2 + x_2^2)^{0.1})) + 1], \quad (17)$$

其中  $x_1, x_2 \in [-1, 1]$ . 该函数在  $(0, 0)$  处取得极小值 0, 其特点是具有数量多、分布广的局部极值“陷阱”, “陷阱”周围被高代价的障碍区所包围, 许多优化算法如遗传算法等都很容易陷入并收敛到其中某一局部极值点. 采用 PC 和改进型 PC 优化该函数时,  $x_1$  和  $x_2$  在其定义域内的离散间隔取 0.04, 离散数为 51. 实验结果如图 3 所示. 其中: 对于改进型 PC, 平均在函数评估 500 次左右收敛于全局最优解, 在收敛速度、平均解质量和稳定性等方面均表现较好; 对于 PC 和 GA, 在少数情况下能找到最优解, 但在大多数情况下均收敛于局部最优解, 尤其 PC 在平均解质量和稳定性等方面表现较差.

图3 Schaffer  $F_7$  函数优化结果比较

### 5.2 Rosenbrock 函数

Rosenbrock 函数<sup>[7]</sup>是另一个常用的无约束优化测试函数, 其定义如下:

$$F_2(X) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2], \quad (18)$$

其中  $x_1, x_2 \in [-5.12, 5.12]$ . 该函数在  $x_i = 1$  处取得极小值, 其特点是全局最优解隐藏在一个狭长的“山谷”中. 在搜索过程中, 大多数优化算法都能很快找到该“山谷”, 即局部最优解的分布区域, 但很容易收敛

于“山谷”边缘而无法找到全局最优解. 本例取  $N=5$ , 采用 PC 和改进型 PC 优化该函数时, 令  $x_i$  的离散间隔为 0.1, 离散数为 101. 实验结果如图 4 所示. 其中: 对于改进型 PC, 平均在函数评估 1 000 次左右收敛于全局最优解, 在收敛速度、平均解质量和稳定性等方面均表现较好; GA 在大多数情况下可收敛于最优解较小的邻域内; 而 PC 的解质量普遍不高, 且在稳定性上表现较差.

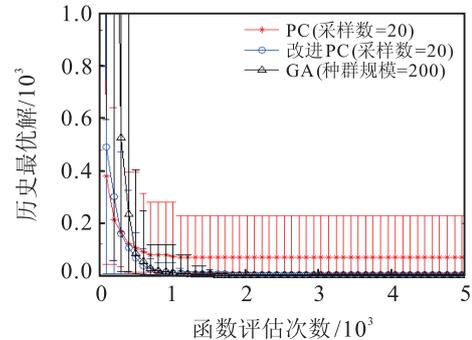


图4 Rosenbrock 函数优化结果比较

### 5.3 Ackley 函数

Ackley 函数<sup>[7]</sup>是一个被广泛采用的多模态测试函数, 其定义如下:

$$F_3(X) = -a e^{-b(\sum_i x_i^2/N)^{0.5}} - e^{\sum_i \cos(cx_i)/N} + a + e. \quad (19)$$

其中:  $a = 20, b = 0.2, c = 2\pi; x_i$  在  $[-32.768, 32.768]$  间取值. 该函数在  $x_i = 0$  处取极小值 0, 其特点与 Schaffer  $F_7$  类似, 但更为复杂. 取  $N=5$ , 采用 PC 和改进型 PC 优化该函数时, 令  $x_i$  的离散间隔为 1.3, 离散数为 51. 实验结果如图 5 所示. 可见, 随着问题复杂度的增加, 收敛速度开始降低. 其中: 对于改进型 PC, 平均在函数评估 4 000 次左右收敛于全局最优解, 且比较稳定; GA 和 PC 均无法在 5 000 次内逼近全局最优解, 尤其 PC 还一度陷入局部最优, 整体表现较差.

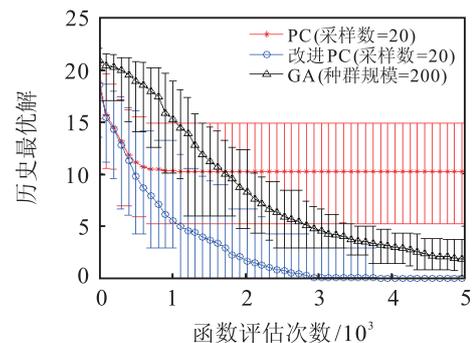


图5 Ackley 函数优化结果比较

### 5.4 Shankar 问题

Shankar 问题<sup>[6]</sup>是一个典型的约束优化问题, 其定义如下:

$$\min F_4(X) = x_1^2 + x_2^2;$$

$$\begin{aligned} \text{s.t. } c_1 &= x_1 + 0.5x_2 < 4, \\ c_2 &= 0.5x_1 + x_2 > 2, \\ x_1, x_2 &\in [-1, 4]. \end{aligned} \quad (20)$$

本文用其测试改进型PC方法在求解带约束优化问题时的性能. 该问题的最优可行解为(0.8, 1.6), 其特点是具有两个相互耦合的约束, 可以对不利用梯度信息的随机优化方法产生较大的干扰, 从而影响优化效率和精度. 采用PC和改进型PC求解该问题时, 令 $x_1$ 和 $x_2$ 的离散间隔为0.1, 离散数为51. 实验结果如图6所示. 为便于衡量可行性, 以罚函数形式将约束加入指标中. 其中: 对于改进型PC, 平均在函数评估1000次左右稳定收敛于最优可行解, 其迭代优化

过程 $x_1$ 和 $x_2$ 的概率分布如图7所示; 对于GA, 平均在函数评估1300次左右收敛于最优可行解; 而PC在少数情况下能找到最优可行解, 且稳定性较差.

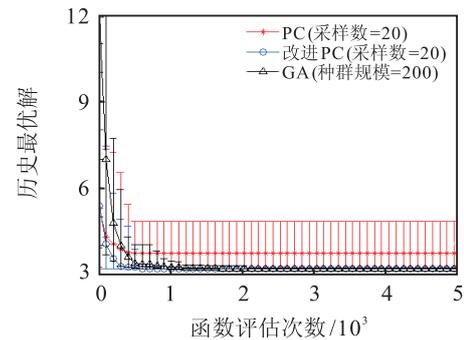


图6 Shankar 约束优化问题优化结果比较

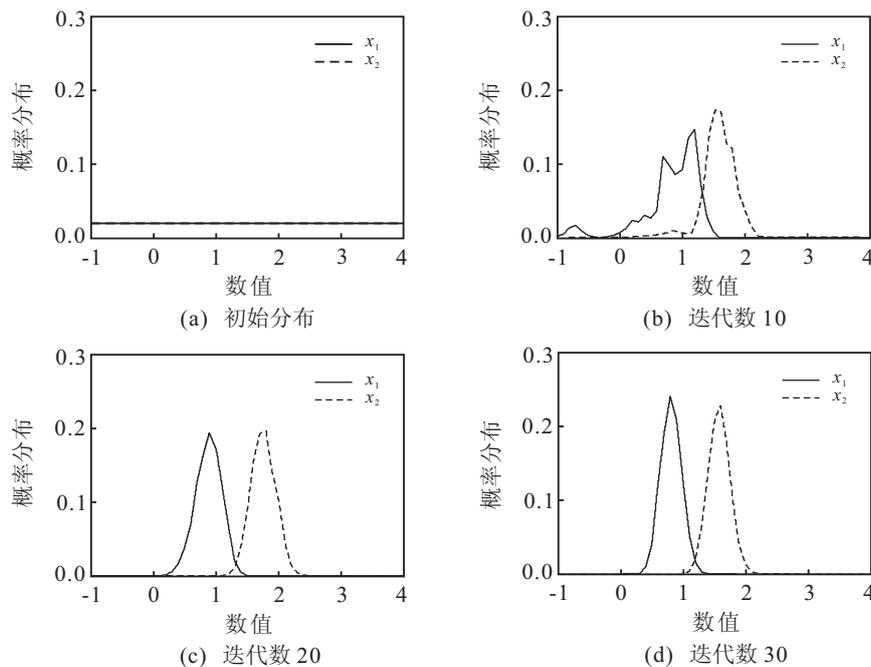


图7 迭代优化过程 $x_1, x_2$ 的概率分布演化

从实验结果可以看出, 标准PC算法在采样数较小的情况下优化性能受到了较大影响, 而改进型PC则有效地克服了上述缺点, 在求解约束及无约束优化问题的测试中均能以较高的效率找到全局最优解, 表现出了良好的性能和稳定性.

## 6 结 论

概率集群优化算法作为一种跨学科、跨领域的新颖分布式优化算法, 具有很强的发展潜力. 本文通过在标准概率集群算法中引入补偿采样机制和概率更新平滑算子, 使智能体的采样策略进一步覆盖了潜在最优区域, 同时保持了概率集群算法的优点. 对约束及无约束Benchmark问题的测试结果表明, 改进型PC算法与原算法相比能以更高的效率和精度找到问题的全局最优解. 同时, 与遗传算法等随机优化方法相比, 改进型PC算法也显示出了良好的性能

和优势. 下一步工作拟结合实际优化问题开展改进型PC算法的应用研究, 并构建相应的分布式优化环境, 以充分发挥该算法的性能优势.

## 参考文献(References)

- [1] Wolpert D H, Tumer K. An introduction to collective intelligence[R]. Moffett Field California: NASA Ames Research Center, 1999.
- [2] Bieniawski S R, Kroo I M, Wolpert D H. Discrete, continuous, and constrained optimization using collectives[C]. The 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf. Albany, 2004.
- [3] Huang C F, Bieniawski S R. A comparative study of probability collectives based multi-agent systems and genetic algorithms[C]. GECCO 2005. Washington, 2005.

(下转第530页)