

基于随机微粒群算法的开放实验室规划研究

王建丽

(太原科技大学计算机科学与技术学院, 山西 太原 030024)

✉wangjianli@tyust.edu.cn



摘要: 近年来, 国家对高校实验室教学平台共享的要求逐步提高, 如何低能耗开放和规划实验室利用是非常有意义的研究。为充分利用随机微粒群算法的强局部搜索力和单纯形法的多样性, 提出了一种改进的基于单纯形法的随机微粒群算法, 通过数值实验证明算法的有效性, 并以实验室电量优化为目标进行仿真, 实验表明本文算法能有效减少电量消耗。

关键词: 实验室规划; 随机微粒群算法; 单纯形法

中图分类号: TP301.6 **文献标识码:** A

Research on Open Laboratory Planning based on Stochastic Particle Swarm Optimization

WANG Jianli

(School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

✉wangjianli@tyust.edu.cn

Abstract: In recent years, China has gradually increased the requirements for the sharing of university laboratory teaching platforms. How to open and plan laboratory utilization with low energy consumption is a very meaningful research. In order to make full use of the strong local search power of the stochastic particle swarm optimization and the diversity of the simplex algorithm, this paper proposes an improved stochastic particle swarm optimization based on the simplex algorithm. Numerical experiments show the effectiveness of the algorithm. Simulation is carried out with the goal of laboratory power optimization. Experiments show that the proposed algorithm can effectively reduce power consumption.

Keywords: laboratory planning; stochastic particle swarm optimization; simplex algorithm

1 引言(Introduction)

高校实验室是进行实验教学和科学研究的重要基地, 近年来国家要求各高等学校加强专业实验室、虚拟仿真实验室、创业实验室和训练中心建设, 促进实验教学平台共享^[1]。目前, 国内高校普遍存在校级公共实验室、专业院级实验室的重复建设, 为了更好地管理和规划开放实验室, 使资源配置和利用率均达到最优, 本文提出规划开放实验室管理的目标函数, 为合理开放实验室提供了算法参考。

微粒群算法(Particle Swarm Optimization, PSO)作为一种智能优化算法^[2-3], 是James Kennedy和Russell Eberhart于1995年提出的。该算法一经提出被广泛应用于很多实际优化

问题, 近几年被应用于群体动画行为自动控制^[4]、天然气管道的优化运行^[5]、城乡应急避难场所规划^[6]等。随机微粒群算法是2004年曾建潮等提出的一种改进微粒群算法^[7], 其基本思想是在基本微粒群算法的基础上, 将基本微粒群进化方程中先前速度项的系数设为0, 不保留先前的速度记忆, 从而取消原有速度对新微粒的影响, 提高局部搜索能力。

单纯形法是求解线性规划问题最常用、最有效的算法之一。单纯形法最早由George Dantzig于1947年提出^[8]。单纯形法的基本思路是: 先找出可行域的一个顶点, 根据一定规则判断其是否最优; 若不是最优值, 则切换到与之相邻的另一顶点, 并使目标函数值更优; 如此下去, 直到找到某最优解

为止。

2 开放实验室数据分析(Data analysis of open laboratory)

随着高校实验室的全面开放，学生会根据自己的需求提出实验申请，信息中心需为学生提供符合条件的实验机房。如何合理地规划机房选取，为学生提供优质实验条件并最大程度避免资源浪费是一项有益的探索。本文采用结合单纯形法改进的随机微粒群算法对开放实验室规模进行规划设计，针对基本实验学时要求、每种实验室的利用率、同一类型实验室内每台电脑的利用率等多目标约束条件提出规划函数，实验仿真求解规划函数，给出设计规划开放实验室的依据。

假定如下数学模型：将全校待安排的机房进行划分，令其变成M行、N列的二维表格，则有M×N个实验室单元。假设一共存在K种实验室类型，表示单元(i,j)，其中i=1,⋯,M; j=1,⋯,N; k=1,⋯,K。规划实验室总体目标为某一上机时间段内所有实验室能耗最低，同时还需满足以下条件：

- (1)学生单门实验课时数需满足本学期最小实验学时要求C_{min};
- (2)同一类型的实验室课时数尽量均衡。

设某一上课时段内单台电脑能耗函数为p(t)，某一种实验室某一上课时段学时数为c(x)，实验室能耗约束目标函数表示为：

$$\begin{cases} \min \sum_{k=1}^K \sum_{i=1}^M \sum_{j=1}^N p(t) \\ \text{s.t.} \sum_{k=1}^K \sum_{i=1}^M \sum_{j=1}^N c(x_{ijk}) \geq C_{\min} \\ \min \sum_{i=1}^M \sum_{j=1}^N p(t), \forall k \in K \end{cases}$$

采用基于随机微粒群算法规划理论将规划方案视为优化目标，将学生上机申请看作一个粒子群，对原始粒子群按照目标函数及其约束条件进行适当的评价，并经过不断地更新，直到使其适应度达到最合适或者满足进化代数为止。

3 算法(Algorithm)

3.1 改进的随机微粒群算法

随机微粒群算法(Stochastic Particle Swarm Optimization, SPSO)进化过程中每个粒子的更新方程为下式：

$$x_i(t+1) = x_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (p_g - x_i(t))$$

这种进化方式保证了全局收敛性。为充分利用随机微粒群算法的强局部搜索力和单纯形法的多样性，同时提高随机微粒群算法的进化速度和质量，本文提出了一种改进的基于单纯形法的随机微粒群算法(Simplex Method-Stochastic Particle Swarm Optimization, SM-SPSO)，该算法的流程描述如下：

Step1: 初始化算法基本参数，给随机微粒群每个粒子

的位置和速度赋初值。

Step2: 奇数种群使用随机微粒群算法进行进化，对于每代进化的微粒，将其当前适应值与之前的最优值进行比较，取其优者保留，如此进化R代。

Step3: 偶数种群使用单纯形法进化R代。

Step4: 混合全部微粒，将最优值作为全局最优值。

Step5: 判读是否满足精度要求，若是，退出程序；否则，其余较好微粒随机分配给两个子种群，重复Step2—Step5。

3.2 测试函数

为了说明SM-SPSO算法的有效性，本文选择了两个典型的多峰测试函数进行测试。

(1)F1: Rastrigin函数

$$f_1(x) = \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i) + A) \quad -5.12 \leq x_i \leq 5.12$$

在 $x_i = 0 (i=1, \dots, n)$ 时达到全局极小值，在 $S = \{x_i \in (-5.12, 5.12), i=1, 2, \dots, n\}$ 范围内大约存在10n个局部极小值。

(2)F2: Griewank函数

$$f_2(x) = \sum_{i=1}^n x_i^2 / 4000 - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1 \quad -10 \leq x_i \leq 10$$

在 $x_i = 0 (i=1, \dots, n)$ 时达到全局极小值，局部极小值范围为：

$$x_i \approx \pm k \cdot \pi \sqrt{i}, \quad i=1, 2, \dots, n; \quad k=0, 1, 2, \dots, n$$

3.3 实验方法

仿真实验中，Rastrigin函数和Griewank函数的维数先取5维实验，再升至10维。设置初始种群数为30个粒子，迭代精度要求取值为0.01，进化3,000代，c₁、c₂各取1.8，都分别进行50次运算。统计收敛率和平均收敛代数，并分析结果。

3.3.1 实验数据分析

Rastrigin函数更新周期、达优次数、进化代数的关系如图1(a)、图1(b)所示，Griewank函数更新周期、达优次数、进化代数的关系如图1(c)、图1(d)所示。

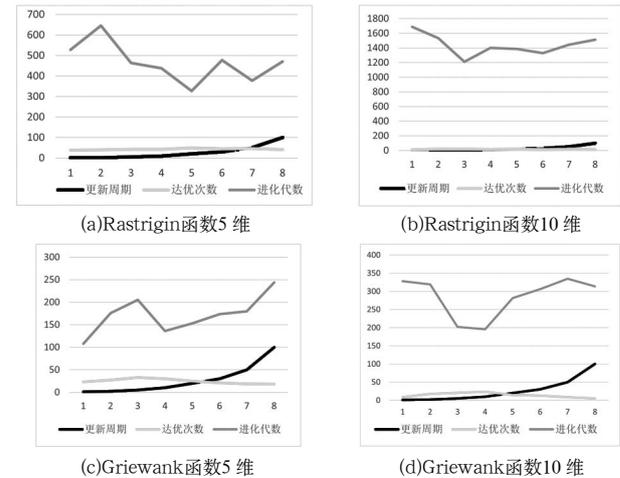


图1 更新周期、达优次数、进化代数的关系

Fig.1 The relationship of renewal period, optimal degree and evolutionary algebra

更新周期如果设置得太大，算法的全局搜索性能会严重下降；更新周期如果设置得太小，算法的通信代价会大大增加，所以综合考虑设置算法更新周期为20。这样在 $c1$ 、 $c2$ 和 R 取值相同的情况下，两个函数平均收敛代数对比如图2(a)、图2(b)所示，平均收敛率对比如图2(c)、图2(d)所示。

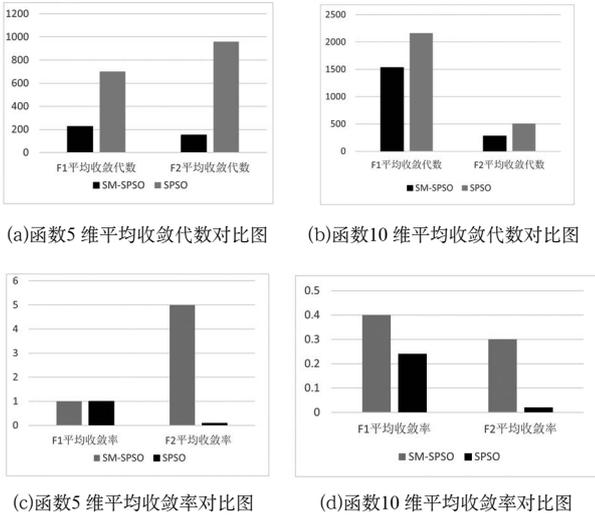


图2 平均收敛代数、平均收敛率对比图

Fig.2 Comparison chart of average convergence algebra and average convergence rate rate

3.3.2 收敛性及结果分析

本文提出的改进的随机微粒群算法是随机微粒群算法的分种群双运行，只是进化过程中对较好微粒进行了混合、筛选，再重新分配，粒子更新仍然按照随机微粒群算法的进化方程来更新，更新过程保存的全局极值点是比原先的极值点下降的，所以改进算法是一种保证全局收敛的算法。

分析图1数据可以得出：对于Rastrigin函数和Griewank函数，本文提出的算法更新周期增加时，两个函数的达优次数均比较稳定。更新周期对进化代数的影响包括：两个函数5维时随着更新周期的增大，进化代数易随之波动；但10维时随着更新周期增大，进化代数趋于稳定。

分析图2数据，对于Rastrigin函数和Griewank函数，本文提出的改进的随机微粒群算法在选取5维、10维时，两个函数的平均收敛代数、平均收敛率均优于基本随机微粒群算法，此数值实验结果表明了算法的有效性。

4 改进随机微粒群算法实验室规划分析(Laboratory planning analysis for improved stochastic particle swarm optimization)

4.1 实验室规划基础数据分析

设单个学生的申请为随机微粒群算法的微粒，每个微粒设置为6维微粒，指标分别为学号、课程名、申请时间、申请实验室类型、实验机器号、能耗值，即： $stu=(stu0,stu1,stu2,$

$stu3,stu4,stu5)^T$ 。

初始微粒随机分配，假设实验学校第 k 种实验室，拥有 i 个实验房间，每个实验房间有 j 台实验仪器，某一门课程要求的最少课时数量为 C_{min} ；某一学期需要进行该项实验学生数目为 num ，取种群规模为取整 $INT(num/(i \times j \times k))$ 。实验室规划要求能耗函数 $p(x)$ 值越低越好，课程课时数函数 $c(x)$ 需大于规定课时数，选择的实验函数 $z(x)$ 需是学校 $K(K \geq 0)$ 种实验室之一，由以上分析得到该实验室规划的多目标优化函数定义：

$$\begin{cases} \min \sum_{stu=1}^{num} p(stu) \\ \text{s.t. } c(stu) \geq C_{min} \\ z(stu) \geq 0 \end{cases}$$

4.2 实验方法和实验结果

参数选取： $c1=c2=1.8$ ，种群数 num 初始值设定为500，进化代数设定为3,000代， k 值取5。

使用SM-SPSO算法求解函数最优值按照如下流程运行：

- Step1: 将粒子按照同一实验室种类 k 值分类。
- Step2: 对于每一类粒子，按照初始种群的 num 倍数分组。
- Step3: 选取一组随机初始化粒子的 $stu4$ 、 $stu5$ 属性参数。
- Step4: 按照SM-SPSO算法开始进化。
- Step5: 对其他组粒子重复Step2—Step4，完成所有粒子寻优。
- Step6: 判断 k 值是否为0，若为0则退出程序；若不为0则 k 值减1，重复Step2—Step6。

选取24个计算机实验机房为例，每个机房有45台电脑可供使用，设定单日申请机房用量500人，初始微粒随机分布如图3(a)所示，优化运算后微粒分布如图3(b)所示。从图中可以看出，以实验室机房能耗最低化为目标函数进行微粒进化后，可以大大提高机房利用率。以单台电脑开机12小时耗电2.76度电计算，每天至少节约1,117.8度电。

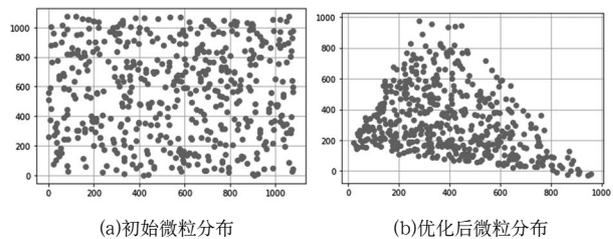


图3 优化前后微粒分布图

Fig.3 Particle distribution diagram before and after optimization

5 结论(Conclusion)

(1)通过求解Rastrigin函数和Griewank函数表明：改进
(下转第22页)