优化算法之粒子群算法（PSO）

lx青萍之末 2018-08-03 10:26:45 124282 收藏 882

分类专栏： # 经典算法及分析 文章标签： 粒子群算法 PSO

粒子群算法的概念

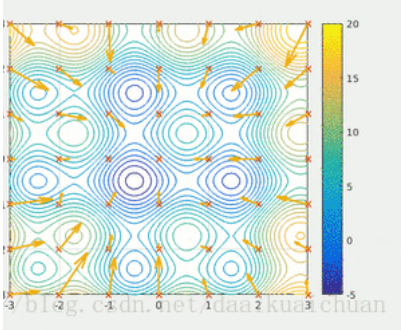
粒子群优化算法(PSO：Particle swarm optimization) 是一种进化计算技术（evolutionary putation）。源于对鸟群捕食的行为研究。粒子群优化算法的基本思想：是通过**群体中个体的协作和信息共享**来寻找最优解.

PSO的优势：在于简单容易实现并且没有许多参数的调节。目前已被广泛应用于函数优化、网络训练、模糊系统控制以及其他遗传算法的应用领域。

粒子群算法分析

基本思想

粒子群算法通过设计一种无质量的粒子来模拟鸟群中的鸟，粒子仅具有两个属性：**速度和位**速度代表移动的快慢，位置代表移动的方向。每个粒子在搜索空间中单独的搜寻最优解，并记为当前个体极值，并将个体极值与整个粒子群里的其他粒子共享，找到最优的那个个体极为整个粒子群的当前全局最优解，粒子群中的所有粒子根据自己找到的当前个体极值和整个群共享的当前全局最优解来调整自己的速度和位置。下面的动图很形象地展示了PSO算法的：



更新规则

PSO初始化为一群随机粒子(随机解)。然后通过迭代找到最优解。在每一次的迭代中，粒子跟踪两个“极值”(pbest，gbest)来更新自己。在找到这两个最优值后，粒子通过下面的公式来自己的速度和位置。

式（1）：

$$v_i = v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i)$$

式（2）：

$$x_i = x_i + v_i$$

公式（1）、（2）中， i=1, 2, . . . , N，N是此群中粒子的总数。
是粒子的速度
id(): 介于(0, 1)之间的随机数
粒子的当前位置
1c_2: 是学习因子，通常$c_1 = c_2 = 2$
]最大值为V_{max}(大于0)，如果v_i大于V_{max}，则$v_i = V_{max}$
式（1）、（2）为*PSO*的标准形式。

;(1)的第一部分称为【记忆项】，表示上次速度大小和方向的影响；公式(1)的第二部分称为身认知项】，是从当前点指向粒子自身最好点的一个矢量，表示粒子的动作来源于自己经验分；公式(1)的第三部分称为【群体认知项】，是一个从当前点指向种群最好点的矢量，反映子间的协同合作和知识共享。粒子就是通过自己的经验和同伴中最好的经验来决定下一步的。以上面两个公式为基础，形成了**PSO的标准形式**。

式（3）：

$$v_i = \omega \times v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i)$$

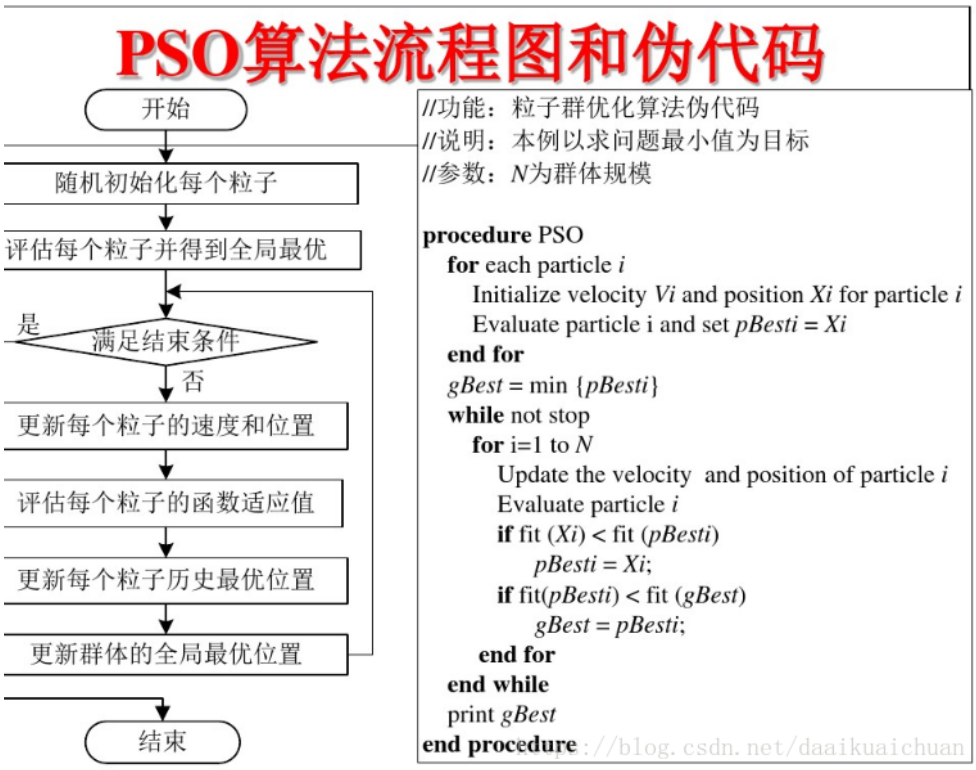
叫做惯性因子，其值为非负。
其值较大，全局寻优能力强，局部寻优能力弱；
其值较小，全局寻优能力弱，局部寻优能力强。
动态ω能获得比固定值更好的寻优结果。动态ω可在PSO搜索过程中
变化，也可以根据PSO性能的某个测度函数动态改变。
目前采用较多的是线性递减权值(Linearly Decreasing Weight, LDW)策略。

$$= (\omega_{ini} - \omega_{end})(G_k - g) / G_k + \omega_{end}$$

最大迭代次数
初始惯性权值
：迭代至最大进化代数时的惯性权值
权值：
ω_{ini}=0.9， ω_{end}=0.4
引入，使用*PSO*算法性能有了很大的提高，针对不同的搜索问题，
调整全局和局部搜索能力，也使PSO算法有成功地应用于很多实际问题。

;(2)和 公式(3)被视为**标准PSO算法**。

SO算法的流程和伪代码



PSO算法举例

6.1 已知函数 $y = f(x_1, x_2) = x_1^2 + x_2^2$

中 $-10 \le x_1, x_2 \le 10$ **，用粒子群优化算法求解y的最小值。**

PSO算法的demo

```
#include <iostream>
#include <vector>
#include <cmath>
#include <map>
#include <algorithm>
#include <random>
#include <ctime>
#include <Eigen/Dense>
using namespace Eigen;
using namespace std;

const int dim = 1;//维数
const int p_num = 10;//粒子数量
const int iters = 100;//迭代次数
const int inf = 999999;//极大值
const double pi = 3.1415;
//定义粒子的位置和速度的范围
const double v_max = 4;
const double v_min = -2;
const double pos_max = 2;
const double pos_min = -1;
//定义位置向量和速度向量
vector<double> pos;
vector<double> spd;
```

```
//定义粒子的历史最优位置和全局最优位置
vector<double> p_best;
double g_best;
//使用eigen库定义函数值矩阵和位置矩阵
Matrix<double, iters, p_num> f_test;
Matrix<double, iters, p_num> pos_mat;

//定义适应度函数
double fun_test(double x)
{
    double res = x * x + 1;
    return res;
}

//初始化粒子群的位置和速度
void init()
{
    //矩阵中所有元素初始化为极大值
    f_test.fill(inf);
    pos_mat.fill(inf);
    //生成范围随机数
    static std::mt19937 rng;
    static std::uniform_real_distribution<double> distribution1(-1, 2);
    static std::uniform_real_distribution<double> distribution2(-2, 4);
    for (int i = 0; i < p_num; ++i)
    {
        pos.push_back(distribution1(rng));
        spd.push_back(distribution2(rng));
    }
    vector<double> vec;
    for (int i = 0; i < p_num; ++i)
    {
        auto temp = fun_test(pos[i]);//计算函数值
        //初始化函数值矩阵和位置矩阵
        f_test(0, i) = temp;
        pos_mat(0, i) = pos[i];
        p_best.push_back(pos[i]);//初始化粒子的历史最优位置
    }
    std::ptrdiff_t minRow, minCol;
    f_test.row(0).minCoeff(&minRow, &minCol);//返回函数值矩阵第一行中极小值对应的位置
    g_best = pos_mat(minRow, minCol);//初始化全局最优位置
}

void PSO()
{
    static std::mt19937 rng;
    static std::uniform_real_distribution<double> distribution(0, 1);
    for (int step = 1; step < iters; ++step)
    {
        for (int i = 0; i < p_num; ++i)
        {
            //更新速度向量和位置向量
            spd[i] = 0.5 * spd[i] + 2 * distribution(rng) * (p_best[i] - pos[i]) +
                2 * distribution(rng) * (g_best - pos[i]);
            pos[i] = pos[i] + spd[i];
            //如果越界则取边界值
            if (spd[i] < -2 || spd[i] > 4)
                spd[i] = 4;
            if (pos[i] < -1 || pos[i] > 2)
                pos[i] = -1;
            //更新位置矩阵
            pos_mat(step, i) = pos[i];
        }
        //更新函数值矩阵
        for (int i = 0; i < p_num; ++i)
        {
```

```
            auto temp = fun_test(pos[i]);
            f_test(step, i) = temp;
        }
        for (int i = 0; i < p_num; ++i)
        {
            MatrixXd temp_test;
            temp_test = f_test.col(i);//取函数值矩阵的每一列
            std::ptrdiff_t minRow, minCol;
            temp_test.minCoeff(&minRow, &minCol);//获取每一列的极小值对应的位置
            p_best[i] = pos_mat(minRow, i);//获取每一列的极小值，即每个粒子的历史最优位置
        }
        g_best = *min_element(p_best.begin(), p_best.end());//获取全局最优位置
    }
    cout << fun_test(g_best);
}

int main()
{
    init();
    PSO();
    system("pause");
    return 0;
}
```

：https://blog.csdn.net/myarrow/article/details/51507671
://blog.csdn.net/google19890102/article/details/30044945
://wenku.baidu.com/view/65c600b9294ac850ad02de80d4d8d15abe230048.html
://blog.csdn.net/darin1997/article/details/80675933

群算法及应用 09-30
详算法及应用 主要讲解蚁群粒子群算法的原理和若干应用场景

群优化算法（PSO）简介及MATLAB实现 Yancy的博客 4万+
粒子群优化算法概述 PSO算法步骤 PSO（粒子群优化算法）与GA（遗传算法）对比 PSO的MATLAB实现 粒子群优…

优质评论可以帮助作者获得更高权重 评论

大威天龙世尊地藏： 讲解的真清楚，不过在实例步骤2中的第三个粒子P3写错了，P3的位置应该是x3=x3+v3 感谢博主！ 2年前 回复 8

weixin_47477263： 所有的教材都不如这个例子讲得透彻。赞！ 8月前 回复 6

guofei9987： 这里有整理好的粒子群算法Python实现 https://github.com/guofei9987/scikit-opt 2年前 回复 4

HigCol： 博主您好，看完您的博客后，写的真是太好了。速度这个向量其实就是跟梯度下降算法中的微分是一样的道理，就是控制鸟往接近全局最优解的地方去飞；然后PSO比梯度下降好的地方是，大家一起来确定怎么飞，梯度下降只是一只鸟往哪儿飞，所以梯度下降很容易飞到局部最优解，PSO可以飞到大家都认为的全局最优解；初始化的速度向量，初始化的量应该不会影响PSO算法的最终结果，但是会影响计算的过程对吧？ 2年前 回复 2

lx青萍之末 博主 回复： 初始值可以随机取嘛，只是迭代的过程不同，最终的趋势是一样的 2年前 回复 2

码哥萧逸凡： 感谢博主，阅读后对PSO的理解更深刻了，算例中是不是有错误？ 5天前 回复

爱码士小周同学的博客： 博主不光能写的一手好代码，还能写的一手好文章。 18天前 回复

weixin_43558200： 请问粒子群算法求多个最优解怎么得呀，比如说选址问题，我要得到多个最佳位置 1月前 回复

qq_36545870 回复： 按照上面的做法之后，做个排序？记录一下位置 1月前 回复

My Alter： 感谢 2月前 回复

点赞196 评论30 分享 收藏882 打赏 举报 关注 一键三连

fuzimango: 动图很帅 3月前 回复

码哥hkcooll: 这个算法容易陷入局部最优，死循环 6月前 回复

qq_45385524 回复： 粒子数取大点试试 6月前 回复

< 1 2 3 >

能算法】粒子群寻优算法 weixin_30752699的博客 2476
基础 粒子群算法（particle swarm optimization，PSO）是计算智能领域中的一种生物启发式方法，属于群体智能优…

群算法 yy2050645的博客 1万+
篇博客，为大家简单介绍粒子群算法（PSO）。粒子群算法同遗传算法相似，也是根据生物界中的

化算法之粒子群算法(PSO) - weixin_44244371的博客 - CSDN博客 10-7
LAB] 经典智能算法1:粒子群优化算法PSO 阅读数 245 经典智能算法文章集:粒子群优化算法PSO:一、粒子群优化算…

化算法---粒子群算法(PSO)_BIRD_CHARLES 12-11
属于进化算法的一种,和模拟退火相似,也是从随机解出发,通过迭代寻找最优解,也是通过适应度来评价解的品质,比遗…

pso（粒子群优化）算法代码合集 07-21
码，共13类

群算法详解 ZCC的专栏 9万+
生背景 *粒子群算法(particleswarm optimization，PSO)由Kennedy和Eberhart在1995年提出，该算法对于Hepper…

群优化算法(PSO)_森的博客_粒子群算法 12-18
算法的发展过程。粒子群优化算法(Partical Swarm Optimization PSO),粒子群中的每一个粒子都代表一个问题的可…

群优化算法(PSO)_DragonBallSuper的博客 12-5
优化算法(PSO:Particle swarm optimization)是一种进化计算技术(evolutionary computation)。 源于对鸟群捕食的行…

解决TSP问题（粒子群算法解决旅行商问题）--python实现 DomicZhong 1万+
感谢这位github上的Marcos Castro de Souza大神 本文利用代码链接如下：https://github.com/marcoscastro/tsp_pso…

群优化算法(PSO) MyArrow的专栏 9万+
念 粒子群优化算法(PSO：Particle swarm optimization) 是一种进化计算技术（evolutionary computation）。 源…

化算法 之 PSO算法_YoYoDelphine的博客 12-22
SO算法简要介绍 PSO算法是一种最优化算法。它预定义一组粒子,和计算粒子适应值(fitness value)的函数。然后进…

群算法(PSO)示例的C++实现_larry233的博客 12-25
算法的介绍和原理见最优化算法之粒子群算法(PSO) 优化目标函数如下: 该目标函数可改为其他函数,设定的粒子数…

thon3实现粒子群优化算法（PSO） FREEMAN 4万+
优化算法（Particle Swarm Optimization，PSO）属于进化算法的一种，是通过模拟鸟群捕食行为设计的。从随机…

群算法原理及Matlab实现（PSO — Particle Swarm Optimization） Tan_HandSome的博客 2万+
在计算机科学中，粒子群优化（PSO）是一种计算方法，可以通过迭代来改进候选方案的优化问题。它通过粒子的…

群算法(PSO)详解以及使用_cxy_hust的博客 12-30
原理(particle swarm optimazition) 1.1 算法使用范围 根据名字: 粒子群优化,其实就界定了该算法是作为一个优化算…

lx青萍之末
码龄6年 暂无认证

点赞196 评论30 分享 收藏882 打赏 举报 关注 一键三连

2016年 18篇