进化优化算法

基于仿生和种群的计算机智能方法

第六课：粒子群优化算法

(Particle Swarm Optimization)

- 群体智能算法及特点
- 粒子群优化算法理论
- 粒子群优化算法流程
- 举例说明粒子群优化算法的工作机理
- 粒子群优化算法的参数设置及影响
- 粒子群优化算法与遗传算法的比较

群体智能算法

- 单个生物体在自然界是很难生存的，然而以群体协作方式形成的活动能力能够保证种群在恶劣的自然环境与物种间的生存竞争中繁衍生息

- 生物群体的一个有趣现象：虽然单个个体的行为表现非常简单，但是群体行为却表现出非常复杂的行为特征

- 群体智能指的是“无智能的主体通过合作表现出智能行为的特征”，群体智能算法是一种基于生物群体行为规律的计算技术

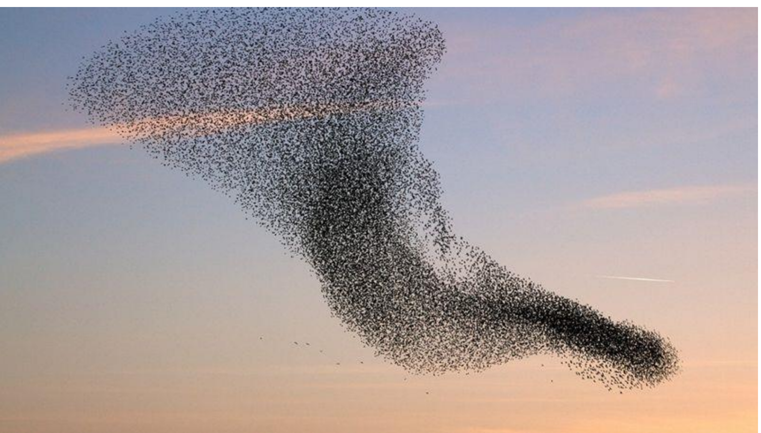
群体智能算法

- ➢群体行为的例子
  - ✓蜂巢附近的蜂群系统
    - ➢显得盲目的蜂群可以极高效率采蜜
    - ➢集体攻击消灭入侵的敌人

  - ✓蚂蚁群落 （单个能力有限，群体完成觅食、筑巢、迁徙、清扫蚁巢）

群体智能算法

- ➢群体行为的例子
  - ✓鸟群飞行与觅食（能在没有集中控制的情况下同步飞行或以某种方式进行队形重组，排列看起来是随机的却有着惊人的同步性，鸟群中蕴涵的“美学”引人注目）
  - ✓鱼群抵御捕食者（任何一只鱼发现异常都可带动整个鱼群逃避外来危险）



群体智能应该遵循的5条基本原则

➢ Millonas在1994年在开发应用于人工生命的模型时提出了群体智能应该遵循的5条基本原则

1. 相似性原则（Proximity Principle）：群内的个体具有对简单的空间或时间进行计算和评估的能力

2. 品质响应原则（Quality Principle）：群内的个体具有对环境以及群内其他个体的品质做出响应的能力

3. 多样性反应原则（Principle of Diverse Response）：群内的不同个体能够对环境中某些变化做出不同的多样反应，群体的行动范围不应该太窄

群体智能应该遵循的5条基本原则

4. 稳定性原则（Stability Principle）：群内个体的行为模式不会在每次环境发生变化时都发生改变

5. 适应性原则（Adaptability Principle）：群内的个体能够在所需代价不高的情况下，适当改变自身的行为模式

以上原则说明，实现群体智能的智能个体必须能够在环境中表现自主性、反应性、学习性和自适应性等智能特征

群体智能算法的特点

- ✓自组织性
  - ■ 群体表现出来的复杂行为是通过简单能力或智能的单个个体的交互过程突现出来的智能，具有自组织性
- ✓鲁棒性
  - ■控制是分布式的,不存在中心控制。具有较强的鲁棒性,即不会由于某一个或几个个体出现故障而影响群体对整个问题的求解

群体智能算法的特点

- ✓简单性
  - ■群体中每个个体的能力或遵循的行为规则非常简单,因而群体智能的实现比较方便,具有简单性的特点，算法中仅涉及各种基本的数学操作，其数据处理过程对CPU和内存的要求也不高
- ✓可扩充性
  - ■由于群体智能可以通过非直接通信的方式进行信息的传输与合作,因而随着个体数目的增加,通信开销的增幅较小,因此,它具有较好的可扩充性

- 群体智能算法及特点
- 粒子群优化算法理论
- 粒子群优化算法流程
- 举例说明粒子群优化算法的工作机理
- 粒子群优化算法的参数设置及影响
- 粒子群优化算法与遗传算法的比较

鸟群觅食带来的灵感

粒子群优化算法的生物学基础

- 雷诺兹（Reynolds）和赫普纳（Heppner）两位生物学家分别在1987年和1990年发表的论文中都关注了鸟群行动中的规律，并对其运动方式进行仿真建模

Computer Graphics, Volume 21, Number 4, July 1987

Flocks, Herds, and Schools: A Distributed Behavioral Model

Craig W. Reynolds
Symbolics Graphics Division

1401 Westwood Boulevard
Los Angeles, California 90024

(Electronic mail: cwr@Symbolics.COM)

Abstract

The aggregate motion of a flock of birds, a herd of land animals, or a school of fish is a beautiful and familiar part of the natural world. But this type of complex motion is rarely seen in computer animation. This paper explores an approach based on simulation as an alternative to scripting the paths of each bird individually. The simulated flock is an elaboration of a particle system, with the simulated birds being the particles. The aggregate motion of the simulated flock is created by a

it seems randomly arrayed and yet is magnificently synchronized. Perhaps most puzzling is the strong impression of intentional, centralized control. Yet all evidence indicates that flock motion must be merely the aggregate result of the actions of individual animals, each acting solely on the basis of its own local perception of the world.

One area of interest within computer animation is the description and control of all types of motion. Computer anima-

AND SCHOOLS: A DISTRIBUTED BEHAVIORAL MODEL

作者： CW Reynolds，H Flocks

摘要： The aggregate motion of a flock of birds, a herd of land animals, or a school of fish is a beautiful and familiar part of the natural world. But this type of complex motion is rarely seen in computer animation. This paper explores an approach based on simulation as an alternative to scripting the paths of each bird individually. The simulated flock is an elaboration of a particle systems, with the simulated birds being the particles. The aggregate motion of the simulated flock is created by a distributed behavioral model much like that at work in a natural flock; th 展开

关键词： actor aggregate motion behavioral animation bird constraints fish flight flock herd particle system

DOI： 10.1145/37401.37406

年份： 1987

引用走势

研究点分析

particle system

粒子群优化算法的生物学基础

- Reynolds设定了鸟群的行为应遵循的规则
  1. 碰撞的避免，即个体应避免和附近的同伴碰撞
  2. 速度的匹配，即个体必须同附近个体的速度保持一致
  3. 向中心聚集，即个体必须飞向邻域的中心

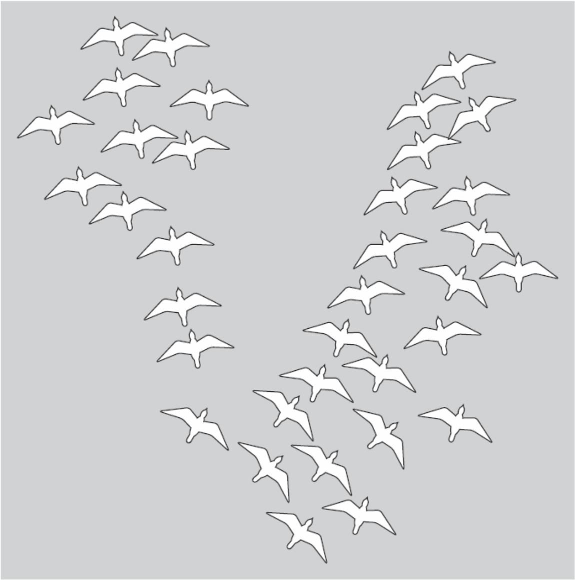


Figure 3.6 Reynolds showed that a realistic bird flock could be programmed by implementing three simple rules: match your neighbors' velocity, steer for the perceived center of the flock, and avoid collisions.

粒子群优化算法的生物学基础

- Heppner指出鸟儿在运动中遵循如下规则
  1. 鸟群中个体都遵循简单的共同行为规则。例如，共同寻找食物栖息地，飞行过程中倾向于向中心聚集但又都能够避免相互碰撞，从而形成一个看似有组织的协调群体
  2. 鸟群觅食时以漫无目的飞行搜寻为主，过程中可能会不断有食物被“眼尖”的个体鸟发现
  3. 一旦发现食物栖息地，个体鸟会主动脱离鸟群降落觅食，并吸引其他更多的鸟追随

Heppner F. Grenander U. A Stochastic nonlinear model for coordinated bird flocks[C]//Krasner. The ubiquity of chaos. Washington, USA: American Association for the Advancement of Science, 1990:233-238

粒子群优化算法的生物学基础

根据上述观察可以得出如下结论

1. 鸟群中总有突出的优秀个体具有更多的经验和能力
2. 群鸟往往通过群体觅食过程获得自身所需的食物，这一过程得益于群体中优秀个体成员的经验和偶然发现
3. 每个个体鸟除了自身不断积累经验之外，还乐意追随其中的优秀个体
4. 由于食物来源大多是以零散分布方式存在，具有一定的不可预知性，相对于个体之间争夺食物的竞争关系，同伴之间的信息共享协作关系更为重要

Heppner F. Grenander U. A Stochastic nonlinear model for coordinated bird flocks[C]//Krasner. The ubiquity of chaos. Washington, USA: American Association for the Advancement of Science, 1990:233-238

粒子群优化算法

- 肯尼迪（J. Kennedy）和埃伯哈特（R. Eberhart）于1995年在IEEE国际神经网络学术会议上发表了题为“Particle Swarm Optimization”的论文，标志着粒子群优化算法的诞生
- 粒子群优化算法（Particle Swarm Optimization----PSO）是一种基于种群的随机的优化算法

James Kennedy

Russell Eberhart

美国社会心理学家James Kennedy和Purdue大学Russell Eberhart教授

粒子群优化算法

- J. Kennedy and R. Eberhart的模型和仿真算法主要对Frank Heppner的模型进行了修正，以使粒子飞向解空间并在最优解处降落。由于它算法简单，容易实现，立刻引起了进化计算领域学者们的广泛关注，成为一个研究热点

Particle swarm optimization

Publisher: IEEE

J. Kennedy ; R. Eberhart **All Authors**

Abstract:
A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and applications, including nonlinear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms are described.

Published in: Proceedings of ICNN'95 - International Conference on Neural Networks

Date of Conference: 27 Nov.-1 Dec. 1995

Date Added to IEEE *Xplore*: 06 August 2002

Print ISBN: 0-7803-2768-3

INSPEC Accession Number: 5263228

DOI: 10.1109/ICNN.1995.488968

Publisher: IEEE

Conference Location: Perth, WA, Australia

Particle Swarm Optimization

James Kennedy[1] and Russell Eberhart[2]

[1]Washington, DC 20212
kennedy_jim@bls.gov

[2]Purdue School of Engineering and Technology
Indianapolis, IN 46202-5160
eberhart@engr.iupui.edu

ABSTRACT
A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and applications, including nonlinear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms are described.

1 INTRODUCTION

This paper introduces a method for optimization of continuous nonlinear functions. The method was discovered through simulation of a simplified social model; thus the social metaphor is discussed, though the algorithm stands without metaphorical support. This paper describes the particle swarm optimization concept in terms of its precursors, briefly reviewing the stages of its development from social simulation to optimizer. Discussed next are a few paradigms that implement the concept. Finally, the implementation of one paradigm is discussed in more detail, followed by results obtained from applications and tests upon which the paradigm has been shown to perform successfully.

Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming. These relationships are briefly reviewed in the paper.

Particle swarm optimization as developed by the authors comprises a very simple concept, and paradigms can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. Early testing has found the implementation to be effective with several kinds of problems. This paper discusses application of the algorithm to the training of artificial neural network weights. Particle swarm optimization has also been demonstrated to perform well on genetic algorithm test functions. This paper discusses the performance on Schaffer's f6 function, as described in Davis [1].

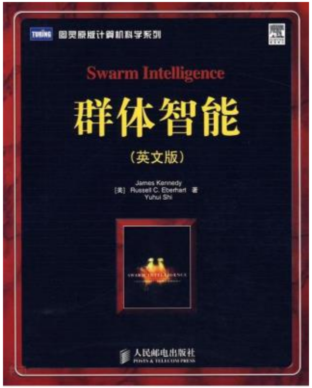
2 SIMULATING SOCIAL BEHAVIOR

A number of scientists have created computer simulations of various interpretations of the movement of organisms in a bird flock or fish school. Notably, Reynolds [8] and Heppner and Grenander [4] presented simulations of bird flocking. Reynolds was intrigued by the aesthetics of bird flocking choreography, and Heppner, a zoologist, was interested in discovering the underlying rules that enabled large numbers of birds to flock synchronously, often changing direction suddenly, scattering and regrouping, etc. Both of these scientists had the insight that local processes, such as those modeled by

粒子群优化算法

- 粒子群算法因具有很好的生物社会背景而易于理解，由于参数少而容易实现，对非线性、多峰问题均具有较强的全局搜索能力，在科学研究与工程实践中得到了广泛关注

- 2001年出版的J. Kennedy and R. Eberhart合著的《群体智能》将群体智能的影响进一步扩大，随后关于粒子群优化算法的研究报告和研究成果大量涌现，继而掀起了国内外研究热潮



粒子群优化算法

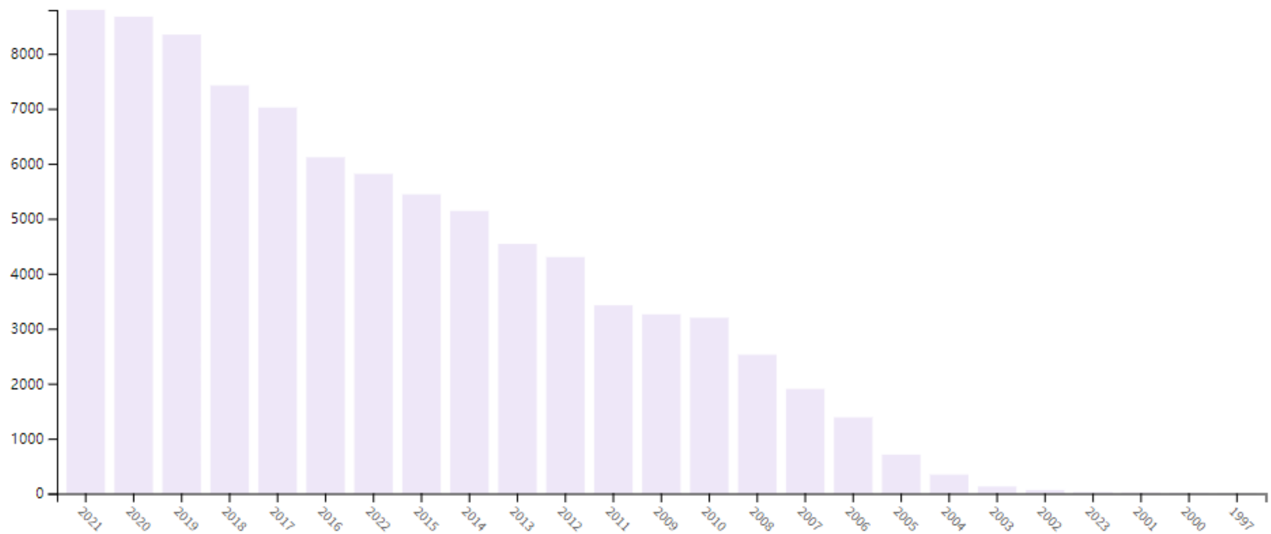
可视化数据:

柱状图

检索结果数:

25

下载



粒子群优化算法

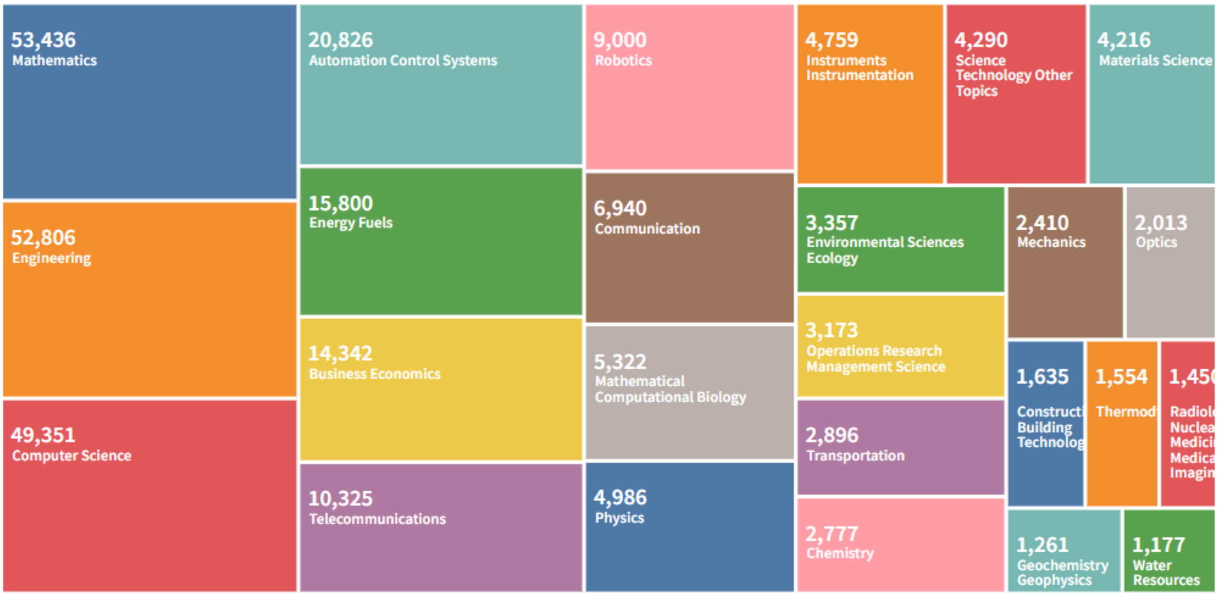
可视化数据:

树状图

检索结果数:

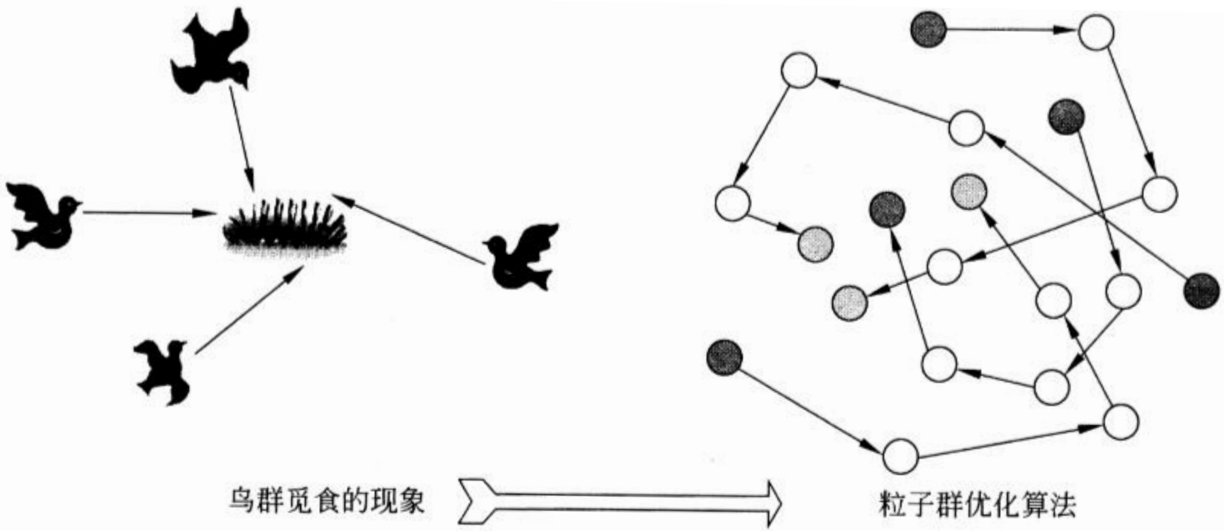
25

下载



图表上的区域并不严格与每个条目的值成正比

从鸟群觅食到粒子群优化算法的关系示意图



一群分散的鸟在随机地飞行觅食，它们不知道食物所在的具体位置，但是有一个间接的机制会让小鸟知道它当前位置离食物的距离(例如食物的香味的浓淡等)。于是各个小鸟就会在飞行过程中不断地记录和更新它曾经到达的离食物最近位置，同时，它们通过信息交流的方法比较大家找到的最好位置，得到一个当前整个群体已经找到的最佳位置。这样，每个小鸟在飞行的时候就有了一个指导的方向，它们会结合自身的经验和整个群体的经验，调整自己的飞行速度和所在位置，不断地寻找更加接近食物的位置，最终使得群体聚集到食物位置。

在粒子群优化算法中，鸟群中的每个小鸟被称为一个“粒子”，通过随机产生一定规模的粒子作为问题搜索空间的有效解。和小鸟一样，每个粒子都具有速度和位置，可以由问题定义的适应度函数确定粒子的适应值，然后不断进行迭代，由粒子本身的历史最优解和群体的全局最优解来影响粒子的速度和位置，让粒子在搜索空间中探索和开发，最终找到全局最优解。

什么是粒子群优化算法?

- 粒子群优化算法是来源于鸟类群体活动的规律性，进而利用群体智能建立一个简化的模型

- 它模拟鸟类的觅食行为，将求解问题的搜索空间比作鸟类的飞行空间，将每只鸟抽象成一个粒子，用它来表征问题的一个可能解，将寻找问题最优解的过程看成鸟类寻找食物的过程，进而求解复杂的优化问题

- 每一个粒子/鸟有一个对应的位置和速度

什么是粒子群优化算法?

- ➢粒子通过改变速度改变粒子所在的位置
  - ✓寻找食物
  - ✓避免被捕食的风险
  - ✓发现最优的环境参数
- ➢粒子的位置和速度受到每一个粒子的最佳位置所影响，在每次循环迭代中改变
  - ✓个体的飞行经验：它记下了在过去的最好位置，为回到那个位置它会改变速度
  - ✓群体的飞行经验：个体知道在当前这一代它的邻居的最好位置。这需要定义邻居的规模，还需要所有邻居针对优化问题的性能相互沟通

什么是粒子群优化算法?

- 每一个粒子是一个候选解，通过在搜索空间中移动来进化，在搜索空间中的移动是粒子群优化算法的精髓所在

- 粒子群算法也是基于“种群”和“进化”的概念，通过个体间的协作与竞争，实现复杂空间最优解的搜索

鸟群觅食和粒子群优化算法的基本定义对照表

鸟群觅食	粒子群优化算法
鸟群	搜索空间的一组有效解（表现为种群规模N）
觅食空间	问题的搜索空间（表现为维数D）
飞行速度	解的速度向量 $v_i=[v_i^1,v_i^2,\ldots v_i^D]$
所在位置	解的位置向量 $x_i=[x_i^1,x_i^2,\ldots x_i^D]$
个体认知与群体协作	每个粒子i根据自身历史最优位置和群体的全局最优位置更新速度和位置
找到食物	算法结束，输出全局最优解

粒子群优化算法中的专门术语

- 粒子（Particle）：每个粒子代表解空间中的一个点，每个粒子可以在解空间中移动（飞行）来搜索优质解
- 群（swarm）：粒子种群或者粒子组。PSO是基于种群的算法，利用一个种群的粒子同时搜索优质解
- 循环（ Cycle ）：迭代。每个循环代表所有的粒子改变位置一次
- 速度（Velocity)：代表每个粒子在一次循环中能够在解空间中移动（飞行）的距离

粒子群优化算法中的专门术语

- 惯性因子（Inertia Weight）：通常用w表示，惯性因子控制给定粒子先前的速度对当前速度的影响
- 个体最佳（p_{best}）：给定粒子到目前为止找到的最佳位置
- 群体最佳（g_{best}）：整个粒子群找到的全局最佳位置
- 自身认知因子（Cognition Weight）：通常用c_1表示，自身认知因子代表一个粒子个体最佳位置对它的吸引程度
- 社会认知因子（Social Weight）：通常用c_2表示，社会认知因子代表全局最佳位置对一个粒子的吸引程度

- 群体智能算法及特点
- 粒子群优化算法理论
- 粒子群优化算法流程
- 举例说明粒子群优化算法的工作机理
- 粒子群优化算法的参数设置及影响
- 粒子群优化算法与遗传算法的比较

粒子群优化算法的三个鲜明特征

- 粒子群优化算法的每个解被称为一个粒子

粒子群优化算法的三个鲜明的特征

每个粒子的最优解

整个粒子群的最优解

更新每一个粒子的速度和位置

- $p_{best,i}$：粒子i的历史最优解
- g_{best}：整个粒子群体的历史最优解

粒子群优化算法

粒子的初始位置和速度在搜索空间内随机生成

粒子速度v由下面公式确定

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i) \qquad (1)$$

粒子的位置通过下式更新

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)} \qquad (2)$$

粒子群优化算法

粒子的初始位置和速度在搜索空间内随机生成

粒子速度v由下面公式确定 $$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i) \quad (1)$$

粒子的位置通过下式更新 $$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)} \quad (2)$$

其中： $i = 1, \ 2, \ldots, N$

$v_i^{(t)}$ 第t次迭代中第i个粒子的速度向量 $v_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$

w 粒子的惯性权重

c_1和c_2 学习因子通常处于（0， 2）区间，c_1为粒子“自身认识因子”，

c_2为“社会认识因子”，通常取$c_1 = c_2 = 2$

r_1和r_2 均匀分布在（0， 1）区间的随机数，（$1 \times D$）

$p_{best,i}^{(t)}$ 第i个粒子本身找到的最优解位置 $p_{best} = [p_{i1}, p_{i2}, \ldots, p_{iD}]$

$g_{best}^{(t)}$ 整个种群目前找到的最优解位置（不断更新，包括部分$t+1$次迭代的信息）

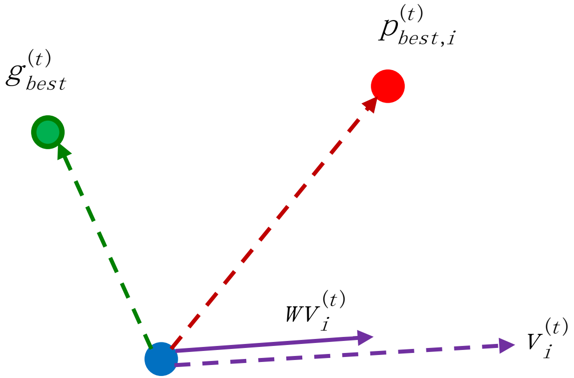
$g_{best} = [g_1, g_2, \ldots, g_D]$

X_i 第i个粒子的位置 $X_i = [X_{i1}, X_{i2}, \ldots, X_{iD}]$

粒子的速度：动量部分

粒子速度 $$v_i^{(t+1)} = \boxed{wv_i^{(t)}} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$wv_i^{(t)}$表示动量部分：
- 粒子先前的速度向量，依据自身的速度进行惯性运动
- 表示粒子对当前自身状态的信任
- 避免速度方向的剧烈改变



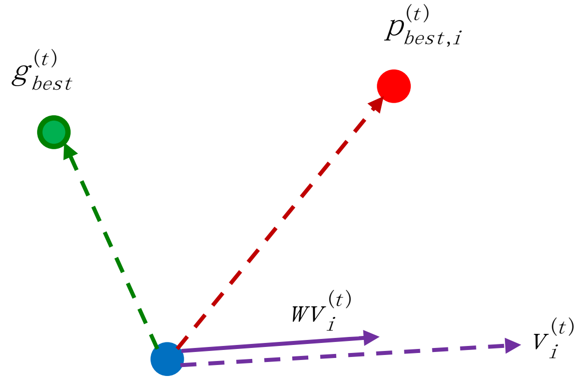
动量部分：$wv_i^{(t)}$

粒子的速度：动量部分

粒子寻优过程贯穿探索和开发两种模式：

* 探索是指粒子在一定程度上从当前的寻优轨迹转到新的方向进行搜索，体现了一种向未知区域开拓的能力，类似于全局搜索；
* 开发则是粒子在一定程度上继续在当前的搜索轨迹上进行更细致的搜索，主要指对探索过程中所搜索到的区域进行更进一步的搜索。
* 影响这两种模式的重要因素是惯性权重w。*PSO*算法的运行是否成功，取决于如何平衡粒子的探索和开发能力。

动量部分：$wV_i^{(t)}$



粒子群优化算法的参数设置

- 惯性权重：控制前一代中粒子的速度对新一代粒子速度的影响

- 平衡探索（exploration）与开发(exploitation)

- 惯性权重值越大越侧重于探索，全局寻优能力较强，局部寻优能力较弱；惯性权重值越小越侧重于开发，全局寻优能力较弱，局部寻优能力较强

粒子群优化算法的参数设置

➢惯性权重值的选取

✓可以设置为常数，通常为0.4~1.4

✓或者在每次迭代中乘以一个衰减系数

✓或者在w_{max}和w_{min}区间范围内线性减小

✓固定的惯性权重可以使粒子保持相同的探索和开发能力，而时变权重可以使粒子在进化的不同阶段拥有不同的探索和开发能力

$$w = \alpha w$$

α：用户定义的衰减系数

$$w = w_{max} - \frac{(w_{max} - w_{min})t}{T}$$

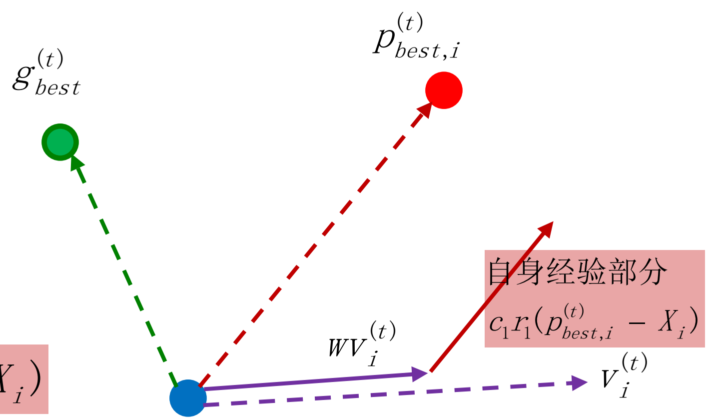
t：当前的迭代次数　　T：最大迭代次数

w_{max}和w_{min}为用户定义的w值的最大取值和最小取值

粒子的速度：自身经验部分

粒子速度 $$v_i^{(t+1)} = wv_i^{(t)} + \boxed{c_1 r_1 (p_{best,i}^{(t)} - X_i)} + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$c_1 r_1 (p_{best,i}^{(t)} - X_i)\begin{cases}\text{自身的经验部分，具有向自己曾找到过的最好点靠近的趋势} \\ \text{以自身经验为指导进行下一步行为的决策} \\ \text{“粒子的怀旧情节”}\end{cases}$$



自身经验部分 $c_1 r_1 (p_{best,i}^{(t)} - X_i)$

粒子的速度：自身经验部分

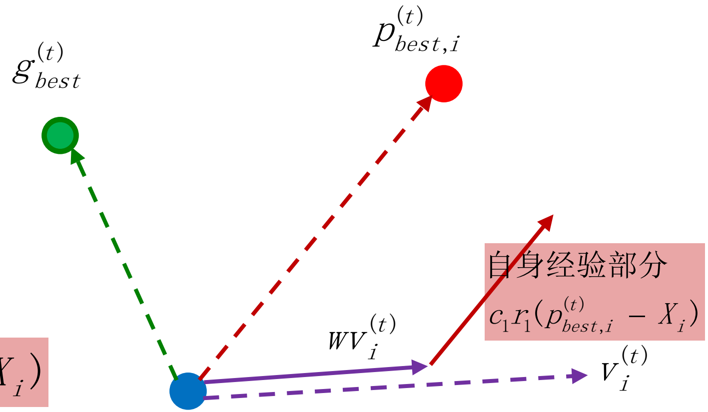
粒子速度 $v_i^{(t+1)} = wv_i^{(t)} + c_1r_1(p_{best,i}^{(t)} - X_i) + c_2r_2(g_{best}^{(t)} - X_i)$

c_1通常取值为2，也可以取介于0和2之间的其他值。

如果$c_1 = 0$，则粒子只有“社会学习”能力，缺少自身的认知能力

如果$c_1 \neq 0$，在粒子的相互作用下，群体有能力在新的搜索空间内搜索，并飞向它本身最好位置$p_{best,i}^{(t)}$和全局最好位置$g_{best}^{(t)}$的加权中心。

在这种情况下，c_1的取值决定了粒子向自身运动的趋势程度



自身经验部分 $c_1r_1(p_{best,i}^{(t)} - X_i)$

粒子的速度：群体经验部分

粒子速度 $v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + \boxed{c_2 r_2 (g_{best}^{(t)} - X_i)}$

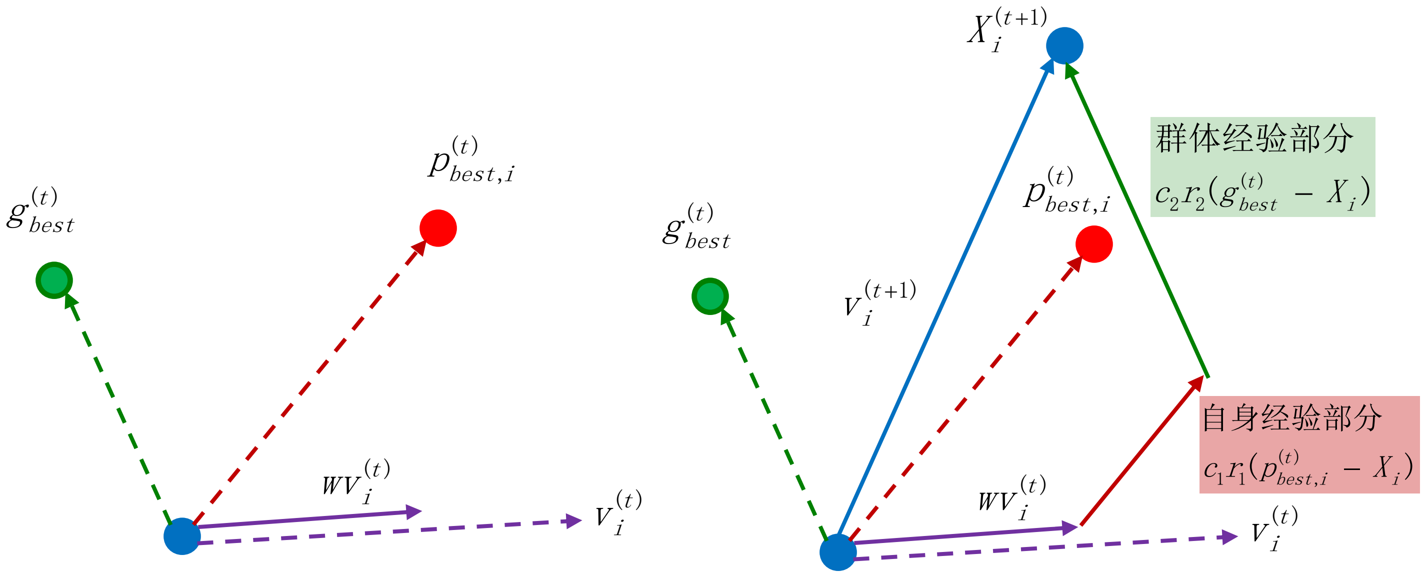
$c_2 r_2 (g_{best}^{(t)} - X_i)$
- 群体的经验部分，向群体或邻域中其他粒子学习
- 以整个群体的知识为指导进行下一步行为的决策
- “每个粒子寻求达到的群体规范”

c_2通常取值为2，也可以取介于0和2之间的其他值。

如果$c_2 = 0$，则粒子之间没有群体信息共享机制，只有“自身认知”部分。在这种模型下，由于不同粒子间缺乏有效的信息交流，一个规模为M的群体就等价于N个粒子的单独运行，因而得到解的概率非常小

群体经验部分 $c_2 r_2 (g_{best}^{(t)} - X_i)$

粒子的速度



动量部分：$wV_i^{(t)}$

自身经验部分 $c_1r_1(p_{best,i}^{(t)} - X_i)$

群体经验部分 $c_2r_2(g_{best}^{(t)} - X_i)$

学习因子对粒子群优化算法的影响

粒子速度v由下面公式确定

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t+1)} - X_i)$$

粒子的位置通过下式更新

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

$c_1 = c_2 = 0$　　粒子按照初始方向移动直至碰到搜索空间的边界为止，粒子仅能搜索有限的区域，所以难以找到最优解

$c_1 > 0 \ \& \ c_2 = 0$　　为“认知”模型，没有社会的共享信息，个体之间没有信息的交互，所以找到最优解的概率较小，一个规模为M的群体等价于N个各行其是的粒子在进行独立爬山算法，在局部空间内搜索局部最优解

$c_1 = 0 \ \& \ c_2 > 0$　　为“社会”模型，粒子缺乏认知能力，只有群体经验，它的收敛速度较快，但容易陷入局部最优

学习因子对粒子群优化算法的影响

粒子速度v由下面公式确定

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t+1)} - X_i)$$

粒子的位置通过下式更新

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

$c_1 = c_2$ 粒子向$p_{best,i}$和g_{best}的平均值靠近

$c_1 >> c_2$ 粒子被吸引到$p_{best,i}$，造成过度游荡的结果

$c_1 << c_2$ 粒子被吸引到g_{best}，造成过早收敛

c_1和c_2取值较小 粒子的运动轨迹更加平滑

c_1和c_2取值很大 粒子的运动轨迹变化剧烈

粒子群优化算法的参数设置

➢学习因子c_1和c_2：代表了粒子向自身极值pBest和全局极值gBest推进的加速权值

1. 根据经验，研究人员建议$c_1+c_2 \le 4$，c_1和c_2通常都等于2.0，代表着对两个引导方向的同等重视[8]
2. 也存在一些c_1和c_2不相等的设置[2][9]，但其范围都在0~4之间
3. 将c_1线性减少，c_2线性增大的设置能动态平衡算法的多样性和收敛性[10]，其中c_1随着种群演化过程的进行从2.5线性递减至0.5，而c_2则从0.5线性递增至2.5。这种调整策略的目的是根据当前优化过程的进行状况，来自适应地调整两个学习因子c_1和c_2，期望得到更好的收敛性能
4. 研究对c_1和c_2的自适应调整方案对算法性能的增强有重要意义[11-13]

粒子群优化算法的参数设置

- 最大速度V_{max}：表示粒子速度在每一个维度上不能超过V_{max}，决定粒子每一次的最大移动距离，制约着算法的探索和开发能力
  1. 粒子的速度在空间上的每一维上都有一个最大速度限制值V_{max}，用来对粒子的速度进行钳制，使速度控制在[- V_{max}, V_{max}]内,这决定问题空间搜索的力度
  2. V_{max}的每一维一般可以取相应维搜索空间的10%~20%，甚至100%
  3. 也有研究将V_{max}设定为变量值定义域的2倍左右
  4. 在文献[3]中，就提出了一种将V_{max}按照进化代数从大到小递减的设置方案
  5. V_{max}过小强调局部搜索，粒子们可能无法对局部最优区域以外的区域进行充分的探测，极易陷入局部最优解，会极大地增加全局搜索的时间，导致搜索的失败
  6. 而较大的V_{max}虽然增加了全局搜索能力，但同时也容易使粒子飞过目标区域，导致粒子局部搜索能力降低

粒子群优化算法

计算目标函数值f_i, 根据适应度函数值更新整个种群信息

更新$p_{best,i}$和g_{best}（假设针对最小化问题）

$$\left.\begin{aligned} p_{best,i} &= X_i \\ f_{p_{best,i}} &= f_i \end{aligned}\right\} \text{如果} f_i < f_{p_{best,i}}$$

$$\left.\begin{aligned} g_{best} &= p_{best,i} \\ f_{g_{best}} &= f_{p_{best,i}} \end{aligned}\right\} \text{如果} f_{p_{best,i}} < f_{g_{best}}$$

可能发生的情况

	比个体最佳要好	比全局最佳好	
1	×	×	不对p_{best}和g_{best}进行更新
2	√	×	更新p_{best}，不更新g_{best}
3	√	√	更新p_{best}和g_{best}

$$\min \quad f(x) = \sum_{i=1}^{2} x_i^2 \qquad 0 \le x_i \le 10 \qquad i = 1, \ 2$$

Case1 :

$$X = [5 \quad 6], \qquad f = 61$$
$$p_{best} = [4 \quad 5], \qquad f_{p_{best}} = 41$$
$$g_{best} = [2 \quad 3], \qquad f_{g_{best}} = 13$$
$$f > f_{p_{best}}$$
$$f > f_{g_{best}}$$

Case2 :

$$X = [4 \quad 3], \qquad f = 25$$
$$p_{best} = [4 \quad 5], \qquad f_{p_{best}} = 41$$
$$g_{best} = [2 \quad 3], \qquad f_{g_{best}} = 13$$
$$f < f_{p_{best}}$$
$$f_{p_{best}} = 25 \qquad \& \qquad p_{best} = [4 \quad 3]$$
$$f > f_{g_{best}}$$

Case3 :

$$X = [1 \quad 3], \qquad f = 10$$
$$p_{best} = [4 \quad 5], \qquad f_{p_{best}} = 41$$
$$g_{best} = [2 \quad 3], \qquad f_{g_{best}} = 13$$
$$f < f_{p_{best}}$$
$$f_{p_{best}} = 10 \qquad \& \qquad p_{best} = [1 \quad 3]$$
$$f < f_{g_{best}}$$
$$f_{g_{best}} = 10 \qquad \& \qquad g_{best} = [1 \quad 3]$$

粒子群优化算法流程图

*Step*1.根据给定的变量初始范围，对粒子群进行随机初始化，包括随机位置和速度

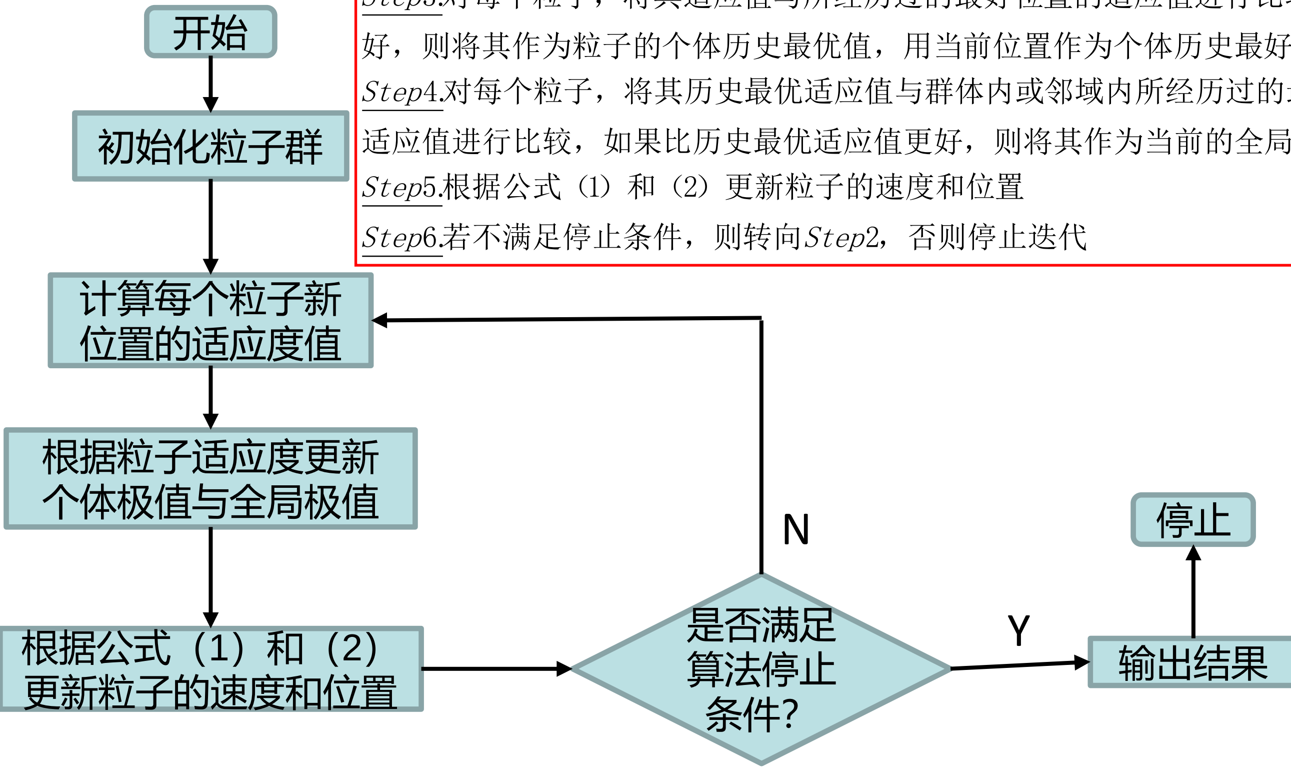
*Step*2.计算每个粒子位置的适应度值

*Step*3.对每个粒子，将其适应值与所经历过的最好位置的适应值进行比较，如果更好，则将其作为粒子的个体历史最优值，用当前位置作为个体历史最好位置

*Step*4.对每个粒子，将其历史最优适应值与群体内或邻域内所经历过的最好位置的适应值进行比较，如果比历史最优适应值更好，则将其作为当前的全局最好位置

*Step*5.根据公式（1）和（2）更新粒子的速度和位置

*Step*6.若不满足停止条件，则转向*Step*2，否则停止迭代



全局模式的粒子群优化算法

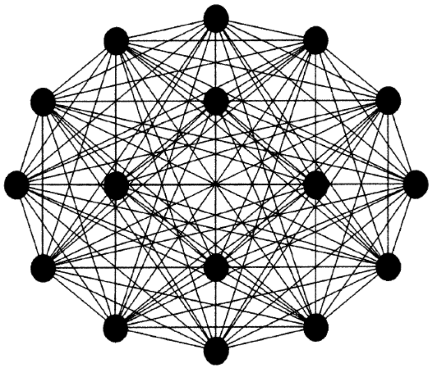
*Gbest*模型（全局最好模型）

*Gbest*模型以牺牲算法的鲁棒性作为代价提高算法的收敛速度，基本粒子群算法就是该模型的具体实现。在该模型中，粒子跟踪两个极值，即自身极值和群全局极值。整个算法以某一粒子为吸引子，将所有粒子拉向它，使所有粒子将最终收敛于该位置，其拓扑结果如下图所示。这样，如果在进化过程中，该最好解得不到有效地更新，则该粒子群将出现类似于遗传算法的早熟现象。为讨论方便，将粒子群算法的进化方程重新表述如下：

$$P_g(t) \in \{P_0(t), P_1(t), \ldots, P_N(t)\} \mid f(P_g(t)) = \min\{f(P_0(t)), f(P_1(t)), \ldots, f(P_N(t))\}$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

- 粒子群优化算法全局版本（整个群体最好位置设为g_{best}）：优化速度快，但有时会陷入局部最优



(a) 全局模型

基本粒子群优化算法伪代码

```
//功能：粒子群优化算法伪代码
//说明：本例以求问题最小值为目标
//参数：N为群体规模

procedure PSO
  for each particle i
     Initialize velocity Vi and position Xi
     for particle i
    Evaluate particle i and set pBesti=Xi
  end for
  gBest=min{pBesti}
  while not stop
    for i=1 to N
       Update the velocity and position of
       particle i
      Evaluate particle i
      if fit (Xi)<fit(pBesti)
        pBesti=Xi;
      if fit(pBesti)<fit (gBest)
        gBest= pBesti;
      end for
  end while
  print gBest
end procedure
```

全局版本

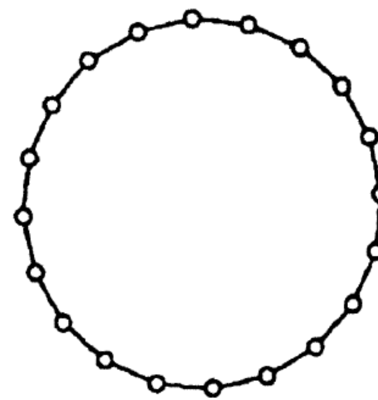
局部模式的粒子群优化算法

$Lbest$模型（局部最好模型）

为防止$Gbest$模型可能出现的早熟现象，$Lbest$模型采用多吸引子代替$Gbest$模型中的单一吸引子。粒子除了追随自身经历的极值之外，不跟踪全局极值$P_g(t)$，而是跟随拓扑近邻粒子当中的局部极值$P_l(t)$。在该模型中，每个粒子需记录自己和它所在局部的最优值，而不需要记录整个群体的最优值。例如，邻居大小为2，则第i粒子只需比较自己适应值和第$i-1$和第$i+1$粒子的适应值大小，如下图所示

局部粒子群算法的进化方程重新表述如下：

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (l_{best}^{(t)} - X_i)$$



（b）局部模型

- 粒子群优化算法局部版本（粒子邻域内的最好位置设为l_{best}）：优化速度慢，但不易陷入局部极值

基本粒子群优化算法伪代码

算法 11.1 最小化 n 元函数 $f(\boldsymbol{x})$ 的基本粒子群优化算法, 其中 $\boldsymbol{x}_i$ 是第 i 个候选解, $\boldsymbol{v}_i$ 是速度向量. 记号 $\boldsymbol{a} \circ \boldsymbol{b}$ 是指向量 $\boldsymbol{a}$ 与向量 $\boldsymbol{b}$ 的元素与元素相乘所成的向量.

初始化一个随机个体的种群 $\{\boldsymbol{x}_i\}$, $i \in [1, N]$
初始化每一个个体的 n 元速度向量 $\boldsymbol{v}_i, i \in [1, N]$
初始化每一个个体到目前为止最好的位置: $\boldsymbol{b}_i \leftarrow \boldsymbol{x}_i$, $i \in [1, N]$
定义邻域规模 $\sigma < N$
定义最大影响值 $\phi_{1,\max}$ 和 $\phi_{2,\max}$
定义最大速度 $\boldsymbol{v}_{\max}$
While not (终止准则)
　　For 每一个个体 $\boldsymbol{x}_i$, $i \in [1, N]$
　　　　$H_i \leftarrow \{\boldsymbol{x}_i$ 最近的 σ 个邻居 $\}$
　　　　$\boldsymbol{h}_i \leftarrow \underset{\boldsymbol{x}}{\operatorname{argmin}}\{f(\boldsymbol{x}) : \boldsymbol{x} \in H_i\}$
　　　　生成一个随机向量 $\boldsymbol{\phi}_1$, $\phi_1(k) \sim U[0, \phi_{1,\max}], k \in [1, n]$
　　　　生成一个随机向量 $\boldsymbol{\phi}_2$, $\phi_2(k) \sim U[0, \phi_{2,\max}], k \in [1, n]$
　　　　$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{\phi}_1 \circ (\boldsymbol{b}_i - \boldsymbol{x}_i) + \boldsymbol{\phi}_2 \circ (\boldsymbol{h}_i - \boldsymbol{x}_i)$
　　　　If $|\boldsymbol{v}_i| > \boldsymbol{v}_{\max}$ then[1]
　　　　　　$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i v_{\max}/|\boldsymbol{v}_i|$
　　　　End if
　　　　$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i$
　　　　$\boldsymbol{b}_i \leftarrow \arg\min\{f(\boldsymbol{x}_i), f(\boldsymbol{b}_i)\}$
　　下一个个体
下一代

局部版本

粒子群优化算法的特点

- 粒子群算法本质是一种随机搜索算法，它是一种新兴的智能优化技术
- 该算法能以较大概率收敛于全局最优解
- 实践证明，它适合在动态、多目标优化环境中寻优，与传统优化算法相比，具有较快的计算速度和更好的全局搜索能力

粒子群优化算法的特点

1. 粒子群算法是基于群智能理论的优化算法，通过群体中粒子间的合作与竞争产生的群体智能指导优化搜索。与其他算法相比，粒子群算法是一种高效的并行搜索算法

2. 粒子群算法与遗传算法都是随机初始化种群，使用适应值来评价个体的优劣程度和进行一定的随机搜索。但粒子群算法根据自己的速度来决定搜索，没有遗传算法的交叉与变异。与进化算法相比，粒子群算法保留了基于种群的全局搜索策略，但是其采用的速度-位移模型操作简单，避免了复杂的遗传操作

粒子群优化算法的特点

3. 由于每个粒子在算法结束时仍保持其个体极值，即粒子群算法除了可以找到问题的最优解外，还会得到若干较好的次优解，因此将粒子群算法用于调度和决策问题可以给出多种有意义的方案

4. 粒子群算法特有的记忆使其可以动态地跟踪当前搜索情况并调整其搜索策略。另外，粒子群算法对种群的大小不敏感，即使种群数目下降时，性能下降也不是很大

- 群体智能算法及特点
- 粒子群优化算法理论
- 粒子群优化算法流程
- 举例说明粒子群优化算法的工作机理
- 粒子群优化算法的参数设置及影响
- 粒子群优化算法与遗传算法的比较

例题：Sphere function

$$\min \quad f(x) = \sum_{i=1}^{4} x_i^2 \qquad 0 \le x_i \le 10 \qquad i = 1,\ 2,\ 3,\ 4$$

决策变量：x_1, x_2, x_3, x_4 $\qquad f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2$

第一步：确定种群规模, 惯性因子，学习因子，最大循环次数

$N_p = 5$, $w = 0.7$, $c_1 = 1.5$, $c_2 = 1.5$, $T = 10$

例题：Sphere function

第二步：在决策变量的取值范围内生成随机解，计算适应度函数值

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix}$$

确定个体最佳及全局最佳解

第三步：在决策变量的取值范围内生成随机速度

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix} \qquad v = \begin{bmatrix} 9 & 6 & 1 & 8 \\ 5 & 1 & 3 & 0 \\ 7 & 4 & 1 & 4 \\ 3 & 0 & 2 & 1 \\ 1 & 6 & 8 & 7 \end{bmatrix}$$

确定个体最佳及全局最佳解

第四步：确定所有解中的个体最佳解和全局最佳解

$$P_{best} = P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{p_{best}} = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix}$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{g_{best}} = 35$$

第一个解：产生新解

第五步：生成两组随机向量

$$r_1 = \begin{bmatrix} 0.4 & 0.3 & 0.9 & 0.5 \end{bmatrix}$$

$$r_2 = \begin{bmatrix} 0.8 & 0.2 & 0.7 & 0.4 \end{bmatrix}$$

$w = 0.7,\quad c_1 = 1.5,\quad c_2 = 1.5,\quad T = 10$

$$v_1 = \begin{bmatrix} 9 & 6 & 1 & 8 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \qquad f = 80$$

$$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \qquad f_{p_{best}} = 80$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{g_{best}} = 35$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第一个解：产生新解

第六步：确定速度

$$v_1 = 0.7\times[9\quad 6\quad 1\quad 8] + 1.5\times[0.4\quad 0.3\quad 0.9\quad 0.5]\times([4\quad 0\quad 0\quad 8]-[4\quad 0\quad 0\quad 8])$$
$$+1.5\times[0.8\quad 0.2\quad 0.7\quad 0.4]\times([0\quad 3\quad 1\quad 5]-[4\quad 0\quad 0\quad 8])$$

$$v_1 = [1.5\quad 5.1\quad 1.75\quad 3.8]$$

$$w = 0.7,\quad c_1 = 1.5,\quad c_2 = 1.5,\quad T = 10$$
$$v_1 = [9\quad 6\quad 1\quad 8]$$
$$X_1 = [4\quad 0\quad 0\quad 8] \qquad f = 80$$
$$p_{best,1} = [4\quad 0\quad 0\quad 8] \qquad f_{p_{best}} = 80$$
$$g_{best} = [0\quad 3\quad 1\quad 5] \qquad f_{g_{best}} = 35$$

第七步：确定位置

$$X_1 = [4\quad 0\quad 0\quad 8] + [1.5\quad 5.1\quad 1.75\quad 3.8] = [5.5\quad 5.1\quad 1.75\quad 11.8]$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1r_1(p_{best,i}^{(t)} - X_i) + c_2r_2(g_{best}^{(t)} - X_i)$$

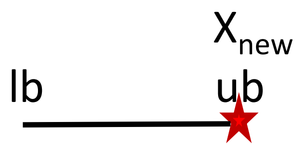
$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

保证新解在取值范围内

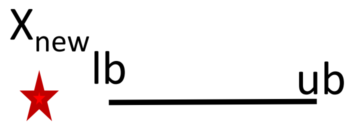
X_{new}在取值范围内

不需要修改任何值

X_{new}违反取值范围的上限



将X_{new}移到取值范围的上限



X_{new}违反取值范围的下限

将X_{new}移到取值范围的下限

第一个解：检查取值范围，更新

$$X_1 = \begin{array}{c} \begin{array}{cccc} \checkmark & \checkmark & \checkmark & \times \end{array} \\ \begin{bmatrix} 5.5 & 5.1 & 1.75 & 11.8 \end{bmatrix} \end{array}$$

第八步：检查变量取值范围，x_4违反了取值范围的上限

$$X_1 = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \end{bmatrix}$$

第九步：计算适应度函数值

$$f_1 = 5.5^2 + 5.1^2 + 1.75^2 + 10^2 = 159.32$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_1 = \begin{bmatrix} 1.5 & 5.1 & 1.75 & 3.8 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \qquad f = 80$$

$$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \qquad f_{p_{best},1} = 80$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{g_{best}} = 35$$

第一个解：检查取值范围，更新

第十步：更新种群

$$P=\begin{bmatrix}5.5 & 5.1 & 1.75 & 10\\ 3 & 1 & 9 & 7\\ 0 & 3 & 1 & 5\\ 2 & 1 & 4 & 9\\ 6 & 2 & 8 & 3\end{bmatrix} \qquad f=\begin{bmatrix}159.32\\ 140\\ 35\\ 102\\ 113\end{bmatrix}$$

第十一步：由于当前解优于新解，因此个体最佳解$p_{best,1}$不做更新

$$p_{best,1}=\begin{bmatrix}4 & 0 & 0 & 8\end{bmatrix} \qquad f_{p_{best},1}=80$$

第十二步：全局最佳解g_{best}不做更新

$$g_{best}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}}=35$$

$$w=0.7,\quad c_1=1.5,\quad c_2=1.5,\quad T=10$$

$$v_1=\begin{bmatrix}1.5 & 5.1 & 1.75 & 3.8\end{bmatrix}$$

$$X_1=\begin{bmatrix}5.5 & 5.1 & 1.75 & 10\end{bmatrix} \qquad f=159.32$$

$$p_{best,1}=\begin{bmatrix}4 & 0 & 0 & 8\end{bmatrix} \qquad f_{p_{best},1}=80$$

$$g_{best}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}}=35$$

第二个解：产生新解

第一步：生成两组随机向量

$$r_1 = [0.1 \quad 0.4 \quad 0.6 \quad 0.3]$$

$$r_2 = [0.7 \quad 0.5 \quad 0.8 \quad 0.2]$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_2 = [5 \quad 1 \quad 3 \quad 0]$$

$$X_2 = [3 \quad 1 \quad 9 \quad 7] \qquad f = 140$$

$$p_{best,2} = [3 \quad 1 \quad 9 \quad 7] \qquad f_{p_{best},2} = 140$$

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{g_{best}} = 35$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第二个解：产生新解

第二步：确定速度

$$v_2 = 0.7 \times [5 \quad 1 \quad 3 \quad 0] + 1.5 \times [0.1 \quad 0.4 \quad 0.6 \quad 0.3] \times ([3 \quad 1 \quad 9 \quad 7] - [3 \quad 1 \quad 9 \quad 7])$$
$$+ 1.5 \times [0.7 \quad 0.5 \quad 0.8 \quad 0.2] \times ([0 \quad 3 \quad 1 \quad 5] - [3 \quad 1 \quad 9 \quad 7])$$

$$v_2 = [0.35 \quad 2.2 \quad -7.5 \quad -0.6]$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$
$$v_2 = [5 \quad 1 \quad 3 \quad 0]$$
$$X_2 = [3 \quad 1 \quad 9 \quad 7] \qquad f = 140$$
$$p_{best,2} = [3 \quad 1 \quad 9 \quad 7] \qquad f_{p_{best},2} = 140$$
$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{g_{best}} = 35$$

第三步：确定位置

$$X_2 = [3 \quad 1 \quad 9 \quad 7] + [0.35 \quad 2.2 \quad -7.5 \quad -0.6] = [3.35 \quad 3.2 \quad 1.5 \quad 6.4]$$

$$v_i^{(t+1)} = w v_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第二个解：检查取值范围，更新

第四步：检查变量取值范围，没有违反取值范围

$$X_2 = [3.35 \quad 3.2 \quad 1.5 \quad 6.4]$$

第五步：计算适应度函数值

$$f_2 = 3.35^2 + 3.2^2 + 1.5^2 + 6.4^2 = 64.67$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_2 = [0.35 \quad 2.2 \quad -7.5 \quad -0.6]$$

$$X_2 = [3 \quad 1 \quad 9 \quad 7] \qquad f = 140$$

$$p_{best,2} = [3 \quad 1 \quad 9 \quad 7] \qquad f_{p_{best},2} = 140$$

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{g_{best}} = 35$$

第二个解：检查取值范围，更新

第六步：更新种群

$$P=\begin{bmatrix}5.5 & 5.1 & 1.75 & 10\\ 3.35 & 3.2 & 1.5 & 6.4\\ 0 & 3 & 1 & 5\\ 2 & 1 & 4 & 9\\ 6 & 2 & 8 & 3\end{bmatrix} \qquad f=\begin{bmatrix}159.32\\ 64.67\\ 35\\ 102\\ 113\end{bmatrix}$$

第七步：由于新解优于当前解，因此个体最佳解$p_{best,2}$需要进行更新

$$p_{best,2}=\begin{bmatrix}3.35 & 3.2 & 1.5 & 6.4\end{bmatrix} \qquad f_{p_{best},2}=64.67$$

第八步：全局最佳解g_{best}不做更新

$$g_{best}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}}=35$$

$w=0.7,\quad c_1=1.5,\quad c_2=1.5,\quad T=10$

$v_2=\begin{bmatrix}0.35 & 2.2 & -7.5 & -0.6\end{bmatrix}$

$X_2=\begin{bmatrix}3.35 & 3.2 & 1.5 & 6.4\end{bmatrix} \qquad f=64.67$

$p_{best,2}=\begin{bmatrix}3.35 & 3.2 & 1.5 & 6.4\end{bmatrix} \qquad f_{p_{best},2}=64.67$

$g_{best}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}}=35$

第三个解：产生新解

第一步：生成两组随机向量

$$r_1 = \begin{bmatrix}0.2 & 0.7 & 0.4 & 0.9\end{bmatrix}$$

$$r_2 = \begin{bmatrix}0.9 & 0.2 & 0.1 & 0.4\end{bmatrix}$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_3 = \begin{bmatrix}7 & 4 & 1 & 4\end{bmatrix}$$

$$X_3 = \begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f = 35$$

$$p_{best,3} = \begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{p_{best},3} = 35$$

$$g_{best} = \begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}} = 35$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第三个解：产生新解

第二步：确定速度

$$v_3 = 0.7 \times [7 \quad 4 \quad 1 \quad 4] + 1.5 \times [0.2 \quad 0.7 \quad 0.4 \quad 0.9] \times ([0 \quad 3 \quad 1 \quad 5] - [0 \quad 3 \quad 1 \quad 5])$$
$$+ 1.5 \times [0.9 \quad 0.2 \quad 0.1 \quad 0.4] \times ([0 \quad 3 \quad 1 \quad 5] - [0 \quad 3 \quad 1 \quad 5])$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$
$$v_3 = [7 \quad 4 \quad 1 \quad 4]$$
$$X_3 = [0 \quad 3 \quad 1 \quad 5] \qquad f = 35$$
$$p_{best,3} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{p_{best},3} = 35$$
$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{g_{best}} = 35$$

$$v_3 = [4.9 \quad 2.8 \quad 0.7 \quad 2.8]$$

第三步：确定位置

$$X_3 = [0 \quad 3 \quad 1 \quad 5] + [4.9 \quad 2.8 \quad 0.7 \quad 2.8] = [4.9 \quad 5.8 \quad 1.7 \quad 7.8]$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第三个解：检查取值范围，更新

第四步：检查变量取值范围，没有违反取值范围

$$X_3 = \begin{bmatrix} 4.9 & 5.8 & 1.7 & 7.8 \end{bmatrix}$$

第五步：计算适应度函数值

$$f_3 = 4.9^2 + 5.8^2 + 1.7^2 + 7.8^2 = 121.38$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_3 = \begin{bmatrix} 4.9 & 2.8 & 0.7 & 2.8 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f = 35$$

$$p_{best,3} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{p_{best},3} = 35$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{g_{best}} = 35$$

第三个解：检查取值范围，更新

第六步：更新种群

$$P=\begin{bmatrix}5.5 & 5.1 & 1.75 & 10\\ 3.35 & 3.2 & 1.5 & 6.4\\ 4.9 & 5.8 & 1.7 & 7.8\\ 2 & 1 & 4 & 9\\ 6 & 2 & 8 & 3\end{bmatrix} \qquad f=\begin{bmatrix}159.32\\ 64.67\\ 121.38\\ 102\\ 113\end{bmatrix}$$

第七步：由于当前解优于新解，因此个体最佳解$p_{best,3}$不进行更新

$$p_{best,3}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{p_{best},3}=35$$

第八步：全局最佳解g_{best}不做更新

$$g_{best}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}}=35$$

$w=0.7,\quad c_1=1.5,\quad c_2=1.5,\quad T=10$

$v_3=\begin{bmatrix}4.9 & 2.8 & 0.7 & 2.8\end{bmatrix}$

$X_3=\begin{bmatrix}4.9 & 5.8 & 1.7 & 7.8\end{bmatrix} \qquad f=121.38$

$p_{best,3}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{p_{best},3}=35$

$g_{best}=\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}}=35$

第四个解：产生新解

第一步：生成两组随机向量

$$r_1 = [0.7 \quad 0.5 \quad 0.8 \quad 0.1]$$

$$r_2 = [0.8 \quad 0.1 \quad 0.7 \quad 0.9]$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_4 = [3 \quad 0 \quad 2 \quad 1]$$

$$X_4 = [2 \quad 1 \quad 4 \quad 9] \qquad f = 102$$

$$p_{best,4} = [2 \quad 1 \quad 4 \quad 9] \qquad f_{p_{best},4} = 102$$

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{g_{best}} = 35$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第四个解：产生新解

第二步：确定速度

$$v_4 = 0.7 \times \begin{bmatrix}3 & 0 & 2 & 1\end{bmatrix} + 1.5 \times \begin{bmatrix}0.7 & 0.5 & 0.8 & 0.1\end{bmatrix} \times \left(\begin{bmatrix}2 & 1 & 4 & 9\end{bmatrix} - \begin{bmatrix}2 & 1 & 4 & 9\end{bmatrix}\right)$$
$$+ 1.5 \times \begin{bmatrix}0.8 & 0.1 & 0.7 & 0.9\end{bmatrix} \times \left(\begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} - \begin{bmatrix}2 & 1 & 4 & 9\end{bmatrix}\right)$$

$$v_4 = \begin{bmatrix}-0.3 & 0.3 & -1.75 & -4.7\end{bmatrix}$$

$w = 0.7,\quad c_1 = 1.5,\quad c_2 = 1.5,\quad T = 10$

$v_4 = \begin{bmatrix}3 & 0 & 2 & 1\end{bmatrix}$

$X_4 = \begin{bmatrix}2 & 1 & 4 & 9\end{bmatrix} \qquad f = 102$

$p_{best,4} = \begin{bmatrix}2 & 1 & 4 & 9\end{bmatrix} \qquad f_{p_{best},4} = 102$

$g_{best} = \begin{bmatrix}0 & 3 & 1 & 5\end{bmatrix} \qquad f_{g_{best}} = 35$

第三步：确定位置

$$X_4 = \begin{bmatrix}2 & 1 & 4 & 9\end{bmatrix} + \begin{bmatrix}-0.3 & 0.3 & -1.75 & -4.7\end{bmatrix} = \begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix}$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第四个解：检查取值范围，更新

第四步：检查变量取值范围，没有违反取值范围

$$X_4 = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix}$$

第五步：计算适应度函数值

$$f_3 = 1.7^2 + 1.3^2 + 2.25^2 + 4.3^2 = 28.13$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$V_4 = \begin{bmatrix} -0.3 & 0.3 & -1.75 & -4.7 \end{bmatrix}$$

$$X_4 = \begin{bmatrix} 2 & 1 & 4 & 9 \end{bmatrix} \qquad f = 102$$

$$p_{best,4} = \begin{bmatrix} 2 & 1 & 4 & 9 \end{bmatrix} \qquad f_{p_{best},4} = 102$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{g_{best}} = 35$$

第四个解：检查取值范围，更新

第六步：更新种群

$$P=\begin{bmatrix}5.5 & 5.1 & 1.75 & 10\\ 3.35 & 3.2 & 1.5 & 6.4\\ 4.9 & 5.8 & 1.7 & 7.8\\ 1.7 & 1.3 & 2.25 & 4.3\\ 6 & 2 & 8 & 3\end{bmatrix} \qquad f=\begin{bmatrix}159.32\\ 64.67\\ 121.38\\ 28.13\\ 113\end{bmatrix}$$

第七步：由于新解优于当前解，因此个体最佳解$p_{best,4}$需要进行更新

$$p_{best,4}=\begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} \qquad f_{p_{best},4}=28.13$$

第八步：$f_{p_{best},4}=28.13<f_{g_{best}}=35$, 全局最佳解$g_{best}$进行更新

$$g_{best}=\begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} \qquad f_{g_{best}}=28.13$$

$w=0.7,\quad c_1=1.5,\quad c_2=1.5,\quad T=10$

$v_4=\begin{bmatrix}-0.3 & 0.3 & -1.75 & -4.7\end{bmatrix}$

$X_4=\begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} \qquad f=28.13$

$p_{best,4}=\begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} \qquad f_{p_{best},4}=28.13$

$g_{best}=\begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} \qquad f_{g_{best}}=28.13$

第五个解：产生新解

第一步：生成两组随机向量

$$r_1 = \begin{bmatrix} 0.3 & 0.8 & 0.2 & 0.1 \end{bmatrix}$$

$$r_2 = \begin{bmatrix} 0.5 & 0.1 & 0.2 & 0.7 \end{bmatrix}$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_5 = \begin{bmatrix} 1 & 6 & 8 & 7 \end{bmatrix}$$

$$X_5 = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = 113$$

$$p_{best,5} = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{p_{best},5} = 113$$

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \qquad f_{g_{best}} = 28.13$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第五个解：产生新解

第二步：确定速度

$$v_5 = 0.7 \times \begin{bmatrix}1 & 6 & 8 & 7\end{bmatrix} + 1.5 \times \begin{bmatrix}0.3 & 0.8 & 0.2 & 0.1\end{bmatrix} \times \left(\begin{bmatrix}6 & 2 & 8 & 3\end{bmatrix} - \begin{bmatrix}6 & 2 & 8 & 3\end{bmatrix}\right)$$
$$+ 1.5 \times \begin{bmatrix}0.5 & 0.1 & 0.2 & 0.7\end{bmatrix} \times \left(\begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} - \begin{bmatrix}6 & 2 & 8 & 3\end{bmatrix}\right)$$

$$v_5 = \begin{bmatrix}-2.52 & 4.09 & 3.87 & 6.26\end{bmatrix}$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$
$$v_5 = \begin{bmatrix}1 & 6 & 8 & 7\end{bmatrix}$$
$$X_5 = \begin{bmatrix}6 & 2 & 8 & 3\end{bmatrix} \qquad f = 113$$
$$p_{best,5} = \begin{bmatrix}6 & 2 & 8 & 3\end{bmatrix} \qquad f_{p_{best},5} = 113$$
$$g_{best} = \begin{bmatrix}1.7 & 1.3 & 2.25 & 4.3\end{bmatrix} \qquad f_{g_{best}} = 28.13$$

第三步：确定位置

$$X_5 = \begin{bmatrix}6 & 2 & 8 & 3\end{bmatrix} + \begin{bmatrix}-2.52 & 4.09 & 3.87 & 6.26\end{bmatrix} = \begin{bmatrix}3.48 & 6.09 & 11.87 & 9.26\end{bmatrix}$$

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1 (p_{best,i}^{(t)} - X_i) + c_2 r_2 (g_{best}^{(t)} - X_i)$$

$$X_i^{(t+1)} = X_i^{(t)} + v_i^{(t+1)}$$

第五个解：检查取值范围，更新

第四步：检查变量取值范围，第3个变量违反取值范围

$$X_5 = [3.48 \quad 6.09 \quad 11.87 \quad 9.26]$$

$$X_5 = [3.48 \quad 6.09 \quad 10 \quad 9.26]$$

第五步：计算适应度函数值

$$f_3 = 3.48^2 + 6.09^2 + 10^2 + 9.26^2 = 234.95$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_5 = [-2.52 \quad 4.09 \quad 3.87 \quad 6.26]$$

$$X_5 = [6 \quad 2 \quad 8 \quad 3] \qquad f = 113$$

$$p_{best,5} = [6 \quad 2 \quad 8 \quad 3] \qquad f_{p_{best},5} = 113$$

$$g_{best} = [1.7 \quad 1.3 \quad 2.25 \quad 4.3] \qquad f_{g_{best}} = 28.13$$

第五个解：检查取值范围，更新

第六步：更新种群

$$P = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 1.7 & 1.3 & 2.25 & 4.3 \\ 3.48 & 6.09 & 10 & 9.26 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 64.67 \\ 121.38 \\ 28.13 \\ 234.95 \end{bmatrix}$$

第七步：由于当前解优于新解，因此个体最佳解$p_{best,5}$不进行更新

$$p_{best,5} = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{p_{best},5} = 113$$

第八步：全局最佳解g_{best}不进行更新

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \qquad f_{g_{best}} = 28.13$$

$$w = 0.7, \quad c_1 = 1.5, \quad c_2 = 1.5, \quad T = 10$$

$$v_5 = \begin{bmatrix} -2.52 & 4.09 & 3.87 & 6.26 \end{bmatrix}$$

$$X_5 = \begin{bmatrix} 3.48 & 6.09 & 10 & 9.26 \end{bmatrix} \qquad f = 234.95$$

$$p_{best,5} = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{p_{best},5} = 113$$

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \qquad f_{g_{best}} = 28.13$$

初始种群

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix}$$

$$P_{best} = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{p_{best}} = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix} \qquad v = \begin{bmatrix} 9 & 6 & 1 & 8 \\ 5 & 1 & 3 & 0 \\ 7 & 4 & 1 & 4 \\ 3 & 0 & 2 & 1 \\ 1 & 6 & 8 & 7 \end{bmatrix}$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{g_{best}} = 35$$

完成第一次迭代之后

经过一次迭代之后

$$P = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 1.7 & 1.3 & 2.25 & 4.3 \\ 3.48 & 6.09 & 10 & 9.26 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 64.67 \\ 121.38 \\ 28.13 \\ 234.95 \end{bmatrix}$$

$$P_{best} = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 0 & 3 & 1 & 5 \\ 1.7 & 1.3 & 2.25 & 4.3 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{p_{best}} = \begin{bmatrix} 80 \\ 64.67 \\ 35 \\ 28.13 \\ 113 \end{bmatrix} \qquad v = \begin{bmatrix} 1.5 & 5.1 & 1.75 & 3.8 \\ 0.35 & 2.2 & -7.5 & -0.6 \\ 4.9 & 2.8 & 0.7 & 2.8 \\ -0.3 & 0.3 & -1.75 & -4.7 \\ -2.52 & 4.09 & 3.87 & 6.26 \end{bmatrix}$$

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \qquad f_{g_{best}} = 28.13$$

伪代码

$\text{Input}: Fitness \quad function, \quad lb, \quad ub, \quad N_p, \quad T, \quad w, \quad c_1, \quad c_2$

1. $Initialize \quad a \quad random \quad population \quad (P) \quad and \quad velocity(v) \quad within \quad the \quad bounds$
2. $Evaluate \quad the \quad objective \quad function \quad value(f) \quad of \quad P$
3. $Assigan \quad p_{best} \quad as \quad P \quad and \quad f_{p_{best}} \quad as \quad f$
4. $Identify \quad the \quad solution \quad with \quad best \quad fitness \quad and \quad assign \quad that \quad solution \quad as \quad g_{best} \quad and \quad fitness \quad as \quad f_{gbest}$

$for \quad t = 1 \quad to \quad T$

$for \quad i = 1 \quad to \quad N_p$

$Determine \quad the \quad velocity(v_i) \quad of \quad i^{th} \quad particle$

$Determine \quad the \quad new \quad position(X_i) \quad of \quad i^{th} \quad particle$

$Bound \quad X_i$

$Evaluate \quad the \quad objective \quad function \quad value(f_i) \quad of \quad i^{th} \quad particle$

$Update \quad the \quad population \quad by \quad indicating \quad X_i \quad and \quad f_i$

$Update \quad p_{best,i} \quad and \quad f_{p_{best}}$

$Update \quad g_{best} \quad and \quad f_{g_{best}}$

end

end

- 群体智能算法及特点
- 粒子群优化算法理论
- 粒子群优化算法流程
- 举例说明粒子群优化算法的工作机理
- 粒子群优化算法的参数设置及影响
- 粒子群优化算法与遗传算法的比较

粒子群优化算法的参数设置

- 粒子的长度D：由优化问题本身决定，就是问题解的长度
- 粒子的范围R：由优化问题本身决定，每一维可以设定不同的范围

粒子群优化算法的参数设置

➢ 种群规模N：影响着算法的搜索能力和计算量

1. PSO对种群规模要求不高，一般取20~40就可以达到很好的求解效果[1]
2. 也有研究建议N取变量个数的2~3倍
3. Carlisle和Dozier[2]的研究中建议将N设为30
4. 不过对于比较难的问题或者特定类别的问题，粒子数可以取到100或200
5. 粒子数目N越大，算法搜索的空间范围就越大，也就更容易发现全局最优解
6. 粒子数目N过小，陷入局部最优的可能性很大
7. 然而，群体过大将导致计算时间的大幅度增加，并且粒子个体达到一定数目后，单纯增加N对PSO的搜索效率不会有显著的改善

粒子群优化算法的参数设置

- 终止条件：决定算法运行的结束，由具体的应用和问题本身确定，终止条件的设定需同时兼顾算法的求解时间、优化质量和搜索效率等多方面性能
    1. 将最大循环数设定为500、1000、5000，或者最大的函数评估次数等
    2. 也可以使用算法求解得到一个可接受的解作为终止条件
    3. 或者是当算法在很长一段迭代中没有得到任何改善，则可以终止算法

粒子群优化算法的参数设置

- 全局和局部PSO：决定算法如何选择两种版本的粒子群优化算法----全局版PSO和局部版PSO
  1. 全局版本PSO将整个群体作为粒子的邻域，具有收敛速度快的优点，不过有时会陷入局部最优
  2. 局部版本PSO将位置相近的个体作为粒子的邻域，收敛速度慢一点，不过不容易陷入局部最优
  3. 在实际应用中，可以根据具体问题选择具体的算法版本，可先采用全局粒子群算法寻找最优解的方向，即得到大致的结果，然后采用局部粒子群算法在最优点附近进行精细搜索
  4. 国内外研究人员已经提出不少粒子拓扑结构的模型，根据现有的成果，可将其划分为3种：空间邻域、性能空间邻域和社会关系邻域

粒子群优化算法的参数设置

- 同步和异步更新：两种更新方式的区别在于对全局gBest或者局部lBest的更新方式
  1. 在同步更新方式中，在每一代中，当所有粒子都采用当前的gBest进行速度和位置的更新之后才对粒子进行评估，更新各自的pBest，再选最好的pBest作为新的gBest
  2. 在异步更新方式中，在每一代中，粒子采用当前的gBest进行速度和位置的更新，然后马上评估，更新自己的pBest，而且如果其pBest要优于当前的gBest，则立刻更新gBest，迅速将更好的gBest用于后面的粒子的更新过程中
  3. 一般而言，异步更新的PSO具有高效的信息传播能力，具有更快的收敛速度

粒子群优化算法的参数设置

边界条件处理:

- 当某一维或若干维的位置或速度超过设定值时，采用边界条件处理策略可将粒子的位置限制在可行搜索空间内，这样能避免种群的膨胀与发散，也能避免粒子大范围地盲目搜索，从而提高了搜索效率。

- 具体的方法有很多种，比如通过设置最大位置限制X_{max}和最大速度限制V_{max}，当超过最大值或最大速度时，在范围内随机产生一个数值代替，或者将其设置为最大值，即边界吸收

参考文献

1. Shi Y H , Eberhart R C . Empirical study of particle swarm optimization[C]// Congress on Evolutionary Computation. IEEE, 2002.
2. Carlisle A , Dozier G . An off-the-shelf PSO[C]// Proceedings of the Workshop on Particle Swarm Optimization. 2001.
3. Fan B H , Shi Y . Study on Vmax of particle swarm optimization[C]// Workshop on Particle Swarm Optimization Indianapolis. 2001.
4. Shi Y . A Modified Particle Swarm Optimizer[C]// Proc of IEEE Icec Conference. 1998.
5. Shi Y , Eberhart R C . Fuzzy adaptive particle swarm optimization[C]// Congress on Evolutionary Computation. IEEE Xplore, 2001.
6. Eberhart R C , Shi Y . Tracking and optimizing dynamic systems with particle swarms[C]// Evolutionary Computation, 2001. Proceedings of the 2001 Congress on. IEEE, 2001.
7. Clerc M , Kennedy J . Kennedy, J.: The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space. IEEE Trans. on Evolutionary Computation 6, 58-73[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1):58-73.
8. Kennedy J , Eberhart R . Particle Swarm Optimization[C]// Icnn95-international Conference on Neural Networks. IEEE, 1995.

参考文献

9. P N Suganthan. Particle Swarm Optimizer with Neighborhood Optimizer. In Proceedings of the Congress on Evolutionary Computation, pp. 1958-1962, 1999
10. Ratnaweera A , Halgamuge S K , Watson H C . Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3):240-255.
11. Ratnaweera A , Halgamuge S K , Watson H C . Particle Swarm Optimization with Self-Adaptive Acceleration Coefficients.[C]// International Conference on Fsdk. DBLP, 2002.
12. Yamaguchi T , Yasuda K . Adaptive Particle Swarm Optimization; Self-coordinating Mechanism with Updating Information[C]// Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on. IEEE, 2006.
13. Adaptive mufti-objective particle swarm optimization algorithm[C]// IEEE Congress on Evolutionary Computation. IEEE, 2007.

- 群体智能算法及特点
- 粒子群优化算法理论
- 粒子群优化算法流程
- 举例说明粒子群优化算法的工作机理
- 粒子群优化算法的参数设置及影响
- 粒子群优化算法与遗传算法的比较

粒子群优化算法与遗传算法的比较

PSO

- 基于种群的搜索
- 没有交叉或变异操作/算子
- 粒子具有记忆功能
- 同时进行全局搜索和局部搜索
- 速度更快

GA

- 基于种群的搜索
- 交叉或变异是重要的操作/算子
- 解没有记忆功能
- 全局搜索的强大工具
- 速度慢

粒子群优化算法与遗传算法的比较

- ✓ Angeline经过大量的实验研究发现，PSO算法保留了基于种群的全局搜索策略，PSO算法所采用的速度与位移模型操作简单，可避免复杂的遗传操作，在解决一些典型的复杂函数优化问题时能够取得比遗传算法更好的求解效果

- ✓ 与遗传算法相比，PSO算法仅需调整少数参数即可用于函数优化，且其对函数没有任何特别的要求（如可微分、时间连续等），因而其通用性更强，在优化多变量、高度非线性、不连续且不可微的函数时具有一定的优势

Note: 对于大多数优化问题，PSO通常可以得到比GA更好的结果

粒子群优化算法与遗传算法的比较

➢粒子群优化算法和遗传算法的相同点

1. 都属于仿生计算。粒子群优化算法主要模拟鸟类觅食、人类认知等社会行为而提出；遗传算法主要模拟生物进化中“适者生存”的规律
2. 都具有很强的随机性。粒子群优化算法公式均带有随机数，而遗传算法的遗传操作也都属于随机操作
3. 都属于全局优化方法。两种算法都是在解空间随机产生初始种群，进而在全局的解空间进行搜索，且将搜索重点集中在寻优性能高的部分
4. 都隐含并行性。它们的搜索过程是从问题解的一个集合开始的，而不是从单个个体开始，具有隐含并行搜索特性，从而减小了陷入局部极小的可能性；并且由于这种并行性，易在计算机上实现，以提高算法的性能和效率
5. 都不受函数约束条件的限制，如连续性、可导性等，根据个体的适应度信息进行搜索，但对高维复杂问题，往往会遇到早熟收敛和收敛性能差的缺点，都无法保证收敛到全局最优点

粒子群优化算法与遗传算法的比较

- 粒子群优化算法和遗传算法的不同点

1. 粒子群优化算法具有记忆特性，粒子获得的较优解都保存，而遗传算法没有记忆，因而以前的某些优良个体的信息会随着种群的演化过程而被改变
2. 在遗传算法中，染色体之间相互共享信息，所以整个种群比较均匀地向最优区域移动。粒子群优化算法中的粒子仅仅通过当前搜索到最优位置进行信息共享，所以在很大程度上这是一种单向信息共享机制，整个搜索更新过程是跟随当前最优解的过程。在大多数情况下，所有粒子可能比遗传算法中的进化个体以更快的速度收敛于最优解
3. 粒子群优化算法相对于遗传算法，不需要编码，没有交叉和变异操作，粒子只是通过内部速度进行更新，因此原理更简单，参数更少，实现更容易

PSO的研究内容和研究方向

算法理论研究	指的是对粒子群优化算法进行理论分析，从数学推导上证明算法具有收敛性，能够保证算法求解到全局最优解。这种理论研究从早期的静态分析，已经逐步发展到动态分析，对保证算法的性能提供了理论依据。	希望从理论上证明算法的搜索能力和收敛能力。
算法参数研究	PSO具有惯量权重 ω，加速系数 c_1、c_2，最大速度，种群规模等参数。参数分析试图通过研究这些参数对算法全局搜索能力和局部搜索能力的影响，找到更好的参数配置或者自适应调参方案，提高算法性能。	希望从实验分析的角度理解算法运行机理，提高算法性能。
拓扑结构研究	PSO有全局版本和局部版本之分，它们的不同之处在于更新速度的时候采用整个种群最优的粒子作为样本还是局部邻居中最优的粒子作为样本。不同的拓扑结构会影响算法局部搜索和全局搜索的能力。	希望通过研究不同的拓扑结构，平衡算法的全局搜索和局部搜索能力。
混合算法研究	PSO自身收敛速度快，但是容易落入局部最优。混合算法引入了进化算法中的相关算子，例如选择、交叉、变异等算子，或者其他的一些增加算法多样性的技术，提高算法性能。	希望通过加入一些附加的算子，提高种群的多样性以提高算法的性能。
算法应用研究	PSO简单易用、性能高效，已经得到了广泛的应用。传统的应用以连续领域的优化问题居多，近年来，将PSO离散化，应用到离散组合优化问题中的研究也频频出现，算法应用范围进一步扩展。	希望通过将算法应用到各种不同的领域，在实践中检验算法的有效性。