# 遗传算法

## 遗传算法的简介

遗传算法（GeneticAlgorithm,GA）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，不需要确定的规则就能自动获取和指导优化的搜索空间，自适应地调整搜索方向。

遗传算法以一种群体中的所有个体为对象，并利用随机化技术指导对一个被编码的参数空间进行高效搜索。其中，选择、交叉和变异构成了遗传算法的遗传操作；参数编码、初始群体的设定、适应度函数的设计、遗传操作设计、控制参数设定五个要素组成了遗传算法的核心内容。

## 相关生物学术语

为了大家更好了解遗传算法，在此之前先简单介绍一下相关生物学术语。

* 基因型(genotype)：性状染色体的内部表现；
* 表现型(phenotype)：染色体决定的性状的外部表现，或者说，根据基因型形成的个体的外部表现；
* 进化(evolution)：种群逐渐适应生存环境，品质不断得到改良。生物的进化是以种群的形式进行的。
* 适应度(fitness)：度量某个物种对于生存环境的适应程度。
* 选择(selection)：以一定的概率从种群中选择若干个个体。一般，选择过程是一种基于适应度的优胜劣汰的过程。
* 复制(reproduction)：细胞分裂时，遗传物质DNA通过复制而转移到新产生的细胞中，新细胞就继承了旧细胞的基因。
* 交叉(crossover)：两个染色体的某一相同位置处DNA被切断，前后两串分别交叉组合形成两个新的染色体。也称基因重组或杂交；
* 变异(mutation)：复制时可能（很小的概率）产生某些复制差错，变异产生新的染色体，表现出新的性状。
* 编码(coding)：DNA中遗传信息在一个长链上按一定的模式排列。遗传编码可看作从表现型到基因型的映射。
* 解码(decoding)：基因型到表现型的映射。
* 个体（individual）：指染色体带有特征的实体；

## 遗传算法的实现

### 3.1遗传算法的基本原理

遗传算法是根据问题的目标函数构造一个适值函数(FitnessFuncton),对一个由多个解(每个解对应一个染色体)构成的种群进行评估、遗传运算、选择,经多代繁殖,获得适应值最好的个体作为问题的最优解。具体可以描述如下。

1.产生一个初始种群

遗传算法是一种基于群体寻优的方法,算法运行时是以一个种群在搜索空间进行搜索。一般是采用随机方法产生一个初始种群。也可以使用其他方法构造一个初始种群。

2.根据问题的目标函数构造适值函数(FitnessFunction)

在遗传算法中使用适值函数来表征种群中每个个体对其生存环境的适应能力,每个个体具有一个适应值(Fitnessvalue)。适应值是群体中个体生存机会的唯一确定性指标。适值函数的形式直接决定着群体的进化行为。适值函数基本上依据优化的目标函数来确定。为了能够直接将适值函数与群体中的个体优劣相联系,在遗传算法中适应值规定为非负,并且在任何情况下总是希望越大越好。

3.根据适应值的好坏不断选择和繁殖

在遗传算法中自然选择规律的体现就是以适应值的大小决定的概率分布来进行选择。个体的适应值越大,该个体被遗传到下一代的概率越大;反之,个体的适应值越小,该个体被遗传到下一代的概率也越小。被选择的个体两两进行繁殖。繁殖产生的个体组成新的种群。这样的选择和繁殖的过程不断重复。

4.若干代后得到适应值最好的个体即为最优解

在若干代后,得到的适应值最好的个体所对应的解即被认为是问题的最优解。

### 3.2构成要素

#### 3.2.1种群和种群大小

种群是由染色体构成的。每个个体就是一个是染色体,每个染色体对应着问题的一个解。种群中个体的数量称为种群大小或者种群规模(PopulationSize,Pop-size),后文中将用NP来表示。种群规模通常是采用一个不变的常数。一般来说,遗传算法中种群规模越大越好,但是种群规模的增大也将导致运算时间的增大,一般设为100~1000。在一些特殊情况下,群体规模也可能采用与遗传代数相关的变量,以获取更好的优化效果。

#### 3.2.2编码方法(EncodingScheme)

编码方法也称为基因表达方法(GeneRepresentation)。在遗传算法中,种群中的每个个体,即染色体是由基因构成的。所以染色体与要优化的问题的解如何进行对应,就需要通过基因来进行表示,即对染色体进行正确的编码。正确地对染色体进行编码来表示问题的解是遗传算法的基础工作,也是最重要的工作。

迄今为止人们已经提出了许多种不同的编码方法。总的来说，这些编码方法可以分为三大类：二进制编码法、浮点编码法、符号编码法。

#### 3.2.3遗传算子(GeneticOperator)

遗传算子包括交叉((Crossover)和变异(Mutation)遗传算子模拟了每一代中创造后代的繁殖过程,是遗传算法的精髓。

交叉是最重要的遗传算子,它同时对两个染色体进行操作,组合二者的特性产生新的后代。交叉的最简单方式是在双亲的染色体上随机地选择一个断点,将断点的右段互相交换,从而形成两个新的后代。这种方法对于二进制编码最合适。遗传算法的性能在很大程度上取决于采用的交叉运算的性能。双亲的染色体是否进行交叉由交叉率来进行控制。

适用于二进制编码个体或浮点数编码个体的交叉算子：

* 单点交叉（One-pointCrossover）：指在个体编码串中只随机设置一个交叉点，然后再该点相互交换两个配对个体的部分染色体。
* 两点交叉与多点交叉：
  1. 两点交叉（Two-pointCrossover）：在个体编码串中随机设置了两个交叉点，然后再进行部分基因交换。
  2. 多点交叉（Multi-pointCrossover）
* 均匀交叉（也称一致交叉，UniformCrossover）：两个配对个体的每个基因座上的基因都以相同的交叉概率进行交换，从而形成两个新个体。
* 算术交叉（ArithmeticCrossover）：由两个个体的线性组合而产生出两个新的个体。该操作对象一般是由浮点数编码表示的个体。

交叉率(记为P)定义为各代中交叉产生的后代数与种群中的个体数的比。显然,较高的交叉率将达到更大的解空间,从而减小停止在非最优解上的机会;但是交叉率太高,会因过多搜索不必要的解空间而耗费大量的计算时间。

变异是在染色体上自发地产生随机的变化。一种简单的变异方式是替换一个或者多个基因。在遗传算法中,变异可以提供初始种群中不含有的基因,或者找到选择过程中丢失的基因,为种群提供新的内容。染色体是否进行变异由变异率来进行控制。

适用于二进制编码个体或浮点数编码个体的：

* 基本位变异（SimpleMutation）：对个体编码串中以变异概率、随机指定的某一位或某几位仅因座上的值做变异运算。
* 均匀变异（UniformMutation）：分别用符合某一范围内均匀分布的随机数，以某一较小的概率来替换个体编码串中各个基因座上的原有基因值。（特别适用于在算法的初级运行阶段）
* 边界变异（BoundaryMutation）：随机的取基因座上的两个对应边界基因值之一去替代原有基因值。特别适用于最优点位于或接近于可行解的边界时的一类问题。
* 非均匀变异：对原有的基因值做一随机扰动，以扰动后的结果作为变异后的新基因值。对每个基因座都以相同的概率进行变异运算之后，相当于整个解向量在解空间中作了一次轻微的变动。
* 高斯近似变异：进行变异操作时用符号均值为Ｐ的平均值，方差为P\*\*2的正态分布的一个随机数来替换原有的基因值。

变异率(记为P)定义为种群中变异基因数在总基因数中的百分比。变异率控制着新基因导入种群的比例。若变异率太低,一些有用的基因就难以进入选择;若太高,即随机的变化太多,那么后代就可能失去从双亲继承下来的好特性,这样算法就会失去从过去的搜索中学习的能力。

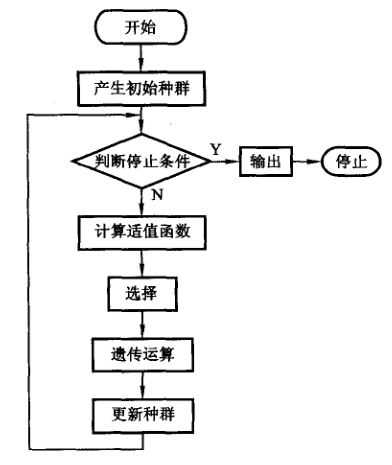
#### 3.2.4选择策略

选择策略是从当前种群中选择适应值高的个体以生成交配池的过程。使用最多的是正比选择策略。选择过程体现了生物进化过程中“适者生存,优胜劣汰”的思想,并保证优良基因遗传给下一代个体。

常用的选择算子：

* 轮盘赌选择（RouletteWheelSelection）：是一种回放式随机采样方法。每个个体进入下一代的概率等于它的适应度值与整个种群中个体适应度值和的比例。选择误差较大。
* 随机竞争选择（StochasticTournament）：每次按轮盘赌选择一对个体，然后让这两个个体进行竞争，适应度高的被选中，如此反复，直到选满为止。
* 最佳保留选择：首先按轮盘赌选择方法执行遗传算法的选择操作，然后将当前群体中适应度最高的个体结构完整地复制到下一代群体中。
* 无回放随机选择（也叫期望值选择ExceptedValueSelection）：根据每个个体在下一代群体中的生存期望来进行随机选择运算。

适应度函数也称评价函数，是根据目标函数确定的用于区分群体中个体好坏的标准。适应度函数总是非负的，而目标函数可能有正有负，故需要在目标函数与适应度函数之间进行变换。评价个体适应度的一般过程为：

* 对个体编码串进行解码处理后，可得到个体的表现型。
* 由个体的表现型可计算出对应个体的目标函数值。
* 根据最优化问题的类型，由目标函数值按一定的转换规则求出个体的适应度。

#### 3.2.5停止准则(StoppingRule/Criterion)

一般使用最大迭代次数作为停止准则。

### 3.3算法流程

下面将以Holland的基本GA为例说明算法的具体实现。流程图如右图：

## 问题处理

### 4.1约束问题

约束优化（ConstrainedOptimization）是处理具有等式和（或）不等式约束的目标函数问题，是人们在实践中遇到最多的数学规划问题之一。

一般的约束优化问题的求解难度是很大的，由于其复杂性，无论在理论研究方面还是实际应用方面都有很大难度，因此吸引了很多研究者投入其中，寻求有效的求解方法。遗传算法是其中一种常用的方法。

用于操作染色体的遗传算法常会产生不可行的染色体的问题。因此处理约束对于遗传算法解决约束优化问题非常重要。一般来说，可以将遗传算法处理约束的方法分为以下几类。

* 拒绝策略：拒绝策略是抛弃进化过程中产生的所有不可行解。这是遗传算法处理约束问题的最简单也是效率最低的方法。当可行解不容易达到时，很难达到一个初始种群。
* 修复策略：修复策略是在进化过程中获得不可行解后，将其修复为可行解， 对于很多组合优化问题创建修复过程相对容易，但是可能导致失去种群多样性。
* 惩罚策略：惩罚策略是对约束进行处理的最一般的方式,是通过对不可行解的惩罚来将约束问题转化为无约束问题。任何对于约束的违反都要在目标函数中添加惩罚项。这就要设计适当的惩罚函数,但是惩罚函数设计不适当则容易掩盖目标函数的优化。
* 特殊的编码和遗传策略：也可以使用特殊的编码策略,在编码时就充分考虑约束问题,在编码时产生的都是符合约束的染色体。为了使染色体在遗传操作后仍然保持可行性,也要使用特殊的遗传策略,使遗传操作后染色体仍然保持可行。

### 4.2多目标的处理

现实的生产和生活中人们常常遇到存在的目标超过一个,并且需要同时处理的情况,而这些不同的目标又往往是相互冲突的,这就是多目标优化(Multiobjective Optimization )问题。

遗传算法正越来越多地被应用于解决多目标问题,遗传算法种群进化特征使其适合于这样的问题。处理多目标问题时,遗传算法遇到的一个主要问题是如何根据多个目标函数值来确定个体的适应值,即适应值分配机制。基本的处理方法包括以下几种。

1.向量评价方法:采用向量形式评价的适应值度量来产生下一代,而不是使用标量适应值度量方式来评价染色体。对于由q个目标的给定问题,每代中的选择过程是一个循环,它重复q次,每次循环依次使用一个目标,每次循环使用这个目标选出下一代中的一部分个体。

2.权重和方法:该方法为每个目标函数分配权重并将权重目标组合为单一目标函数。只需要合适的权重就可以实现该方法。权重的调整方法包括:固定权重方法;随机权重方法;适应性权重方法等。

3.基于Pareto 的方法：有两种基于Pareto 的方法:Pareto 排序和Pareto 竞争。Pareto 基于排序的适应值分配方法是希望对所有Pareto 个体分配相同的复制概率。它主要包括两个主要步骤:1.基于Pareto 排序对种群进行分类。2.根据排序对个体分配选择概率。

Pareto 竞争方法中采用了小生境Pareto 概念,而不是非支配分类和排序选择。小生境Pareto 指的是具有最小邻居数量的Pareto 解赢得竞争。

4.妥协方法：妥协方法是通过某种距离的度量来确定与理想解最近的解。为了克服找理想点的困难,也可以使用部分已经探索的解空间中的代理理想点来替代整个解空间的理想点。

5.目标规划方法：该方法是采用基于排序的适应值分配方法来判断个体的价值。具体过程为:根据第一优先目标进行种群排序,如果某些个体具有相同的目标值,根据第二优先目标进行排序,如此进行下去。如果各个目标上各个个体均相同,则随机对其进行排序。然后采用从最好到最差的指数插值进行个体的适应值分配。

# 禁忌搜索算法

## 禁忌搜索算法的简介

禁忌搜索算法（Tabu Search或Taboo Search，简称TS算法）是一种全局性邻域搜索算法，[模拟](https://wiki.mbalib.com/wiki/%E6%A8%A1%E6%8B%9F)人类具有[记忆](https://wiki.mbalib.com/wiki/%E8%AE%B0%E5%BF%86)功能的寻优特征。它通过局部邻域搜索机制和相应的禁忌准则来避免迂回搜索，并通过破禁水平来释放一些被禁忌的优良状态，进而保证多样化的有效探索，以最终实现全局优化。

禁忌搜索(Tabu Search 或Taboo Search 简称T)算法是继遗传算法之后出现的又一种元启发式(Meta-Heuristic ))优化算法,最早于1977年由Glover 提出。禁忌搜索算法模仿人类的记忆功能,使用禁忌表来封锁刚搜索过的区域来避免迂回搜索,同时赦免禁忌区域中的一些优良状态,进而保证搜索的多样性,从而达到全局优化。

## 禁忌搜索算法的实现

### 2.1禁忌搜索算法的基本原理

简单TS算法的基本思想是：给定一个当前解（初始解）和一种邻域，然后在当前解的邻域中确定若干候选解；若最佳候选解对应的[目标值](https://wiki.mbalib.com/wiki/%E7%9B%AE%E6%A0%87%E5%80%BC)优于“best so far”状态，则忽视其禁忌特性，用其替代当前解和“best so far”状态，并将相应的对象加入禁忌表，同时修改禁忌表中各对象的任期；若不存在上述候选解，则选择在候选解中选择非禁忌的最佳状态为新的当前解，而无视它与当前解的优劣，同时将相应的对象加入禁忌表，并修改禁忌表中各对象的任期；如此重复上述迭代搜索过程，直至满足停止准则。简单禁忌搜索的算法步骤可描述如下：

（1）给定算法参数，随机产生初始解x，置禁忌表为空。

（2）判断算法终止条件是否满足？若是，则结束算法并输出优化结果；否则，继续以下步骤。

（3）利用当前解的邻域函数产生其所有（或若干）邻域解，并从中确定若干候选解。

（4）对候选解判断藐视准则是否满足？若成立，则用满足藐视准则的最佳状态y替代x成为新的当前解，即x=y，并用与y对应的禁忌对象替换最早进入禁忌表的禁忌对象，同时用y替换“best so far”状态，然后转步骤6；否则，继续以下步骤。

（5）判断候选解对应的各对象的禁忌属性，选择候选解集中非禁忌对象对应的最佳状态为新的当前解，同时用与之对应的禁忌对象替换最早进入禁忌表的禁忌对象元素。

（6）转步骤（2）。

### 2.2构成要素

#### 2.2.1邻域移动:

邻域移动是从一个解产生另一个解的途径。它是保证产生好的解和算法搜索速度的最重要因素之一。邻域移动定义的方法很多，对于不同的问题应采用不同的定义方法。通过移动，目标函数值将产生变化，移动前后的目标函数值之差，称之为移动值。如果移动值是非负的，则称此移动为改进移动;否则称作非改进移动。最好的移动不一定是改进移动，也可能是非改进移动，这一点就保证搜索陷入局部最优时，禁忌搜索算法能自动把它跳出局部最优。

**2.2.1禁忌表:**

不允许恢复即被禁止的性质称作Tabu。禁忌表的主要目的是阻止搜索过程中出现循环和避免陷入局部最优，它通常记录前若干次的移动，禁止这些移动在近期内返回。在迭代固定次数后，禁忌表释放这些移动，重新参加运算，因此它是一个循环表，每迭代一次，将最近的一次移动放在禁忌表的末端，而它的最早的一个移动就从禁忌表中释放出来。为了节省记忆时间，禁忌表并不记录所有的移动，只记录那些有特殊性质的移动，如记载能引起目标函数发生变化的移动。禁忌表记载移动的方式主要有三种:\*记录目标值；\*移动前的状态；\*移禁忌搜索算法在冷藏供应链配送网络中的应用研究动本身。

禁忌表是禁忌搜索算法的核心，禁忌表的大小在很大程度上影响着搜素速度和解的质量。如果选择的好，可有助于识别出曾搜索过的区域。实验表明，如果禁忌表长度过小，那么搜索过程就可能进入死循环，整个搜索将围绕着相同的几个解徘徊;相反，如果禁忌表长度过大，那它将在相当大的程度上限制了搜索区域，好的解就有可能被跳过，同时，不会改进解的效果而增加算法运算时间。因此一个好的禁忌表长度应该是尽可能小却又能避免算法进入循环。禁忌表的这种特性非常类似于“短期记忆”，因而人们把禁忌表称作短期记忆函数。

禁忌表另一个作用是通过调整禁忌表的大小使搜索发散或收敛。初始搜索时，为提高解的分散性，扩大搜索区域，使搜索路径多样化，经常希望禁忌表长度小。

相反当搜索过程接近最优解时，为提离解的集中性，减少分散，缩小搜索区域，这时通常希望禁忌表长度大。为达到这样的目的，最近越来越多的人们允许禁忌表的大小和结构随搜索过程发生改变，即使用动态禁忌表，实验结果表明了动态禁忌表往往比固定禁忌表获得更好的解。

**2.2.2选择策略:**

选择策略即择优规则，是对当前的邻域移动选择一个移动而采用的准则。择优规则可以采用多种策略，不同的策略对算法的性能影响不同。一个好的选择策略应该是既保证解的质量又保证计算速度。当前采用最广泛的两类策略是最好解优先策略(Bestlmprovedstrategy)和第一个改进解优先策略(FirstImProvedstrategy)。最好改进解优先策略就是对当前邻域中选择移动值最好的移动产生的解，作为下一次迭代的开始。而第一个改进解优先策略是搜索邻域移动时选择第一改进当前解的邻域移动产生的解作为下一次迭代的开始。最好改进解优先策略相当于寻找最陡的下降，这种择优规则效果比较好，但是它需要更多的计算时间;而最快的下降对应寻找第一个改进解的移动，由于它无需搜索整个一次邻域移动，所以它所花计算时间较少，对于比较大的邻域，往往比较适合。

**2.2.3破禁策略:**

破禁策略通常指渴望水平(Aspiration)函数选择，当一个禁忌移动在随后}T}次的迭代内再度出现时，如果它能把搜索带到一个从未搜索过的区域，则应该接受该移动即破禁，不受禁忌表的限制。衡量标准就是定义一个渴望水平函数，渴望水平函数通常选取当前迭代之前所获得的最好解的目标值或此移动禁忌时的目标值作为渴望水平函数。

**2.2.4停止规则:**

在禁忌搜索中停止规则通常有两种，一种是把最大迭代数作为停止算法的标准，而不以局优为停止规则;另一种是在给定数目的迭代内所发现的最好解无法改进或无法离开它时，算法停止。

**2.2.5长期表:**

短期记忆用来避免最近所作的一些移动被重复，但是在很多的情况下短期记忆并不足以把算法搜索带到能够改进解的区域。因此在实际应用中常常短期记忆与长期记忆相结合使用，以保持局部的强化和全局多样化之间的平衡，即在加强与好解有关性质的同时还能把搜索带到未搜索过的区域。

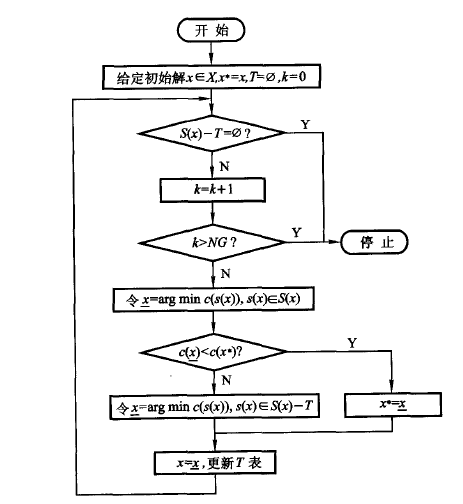
**2.2.1长期记忆:**

在长期记忆中，频率起着非常重要的作用，使用频率的目的就是通过了解同样的选择在过去做了多少次来重新指导局部选择。当在非禁忌移动中找不到可以改进的解时用长期记忆更有效。

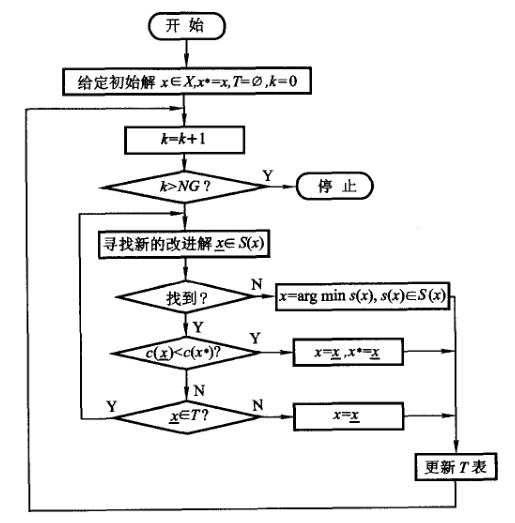
**2.2.1长期记忆函数:**

目前长期记忆函数主要有两种形式，一种通过惩罚的形式，即用一些评价函数来惩罚在过去的搜索中用得最多或最少的那些选择，并用一些启发方法来产生新的初始点。用这种方式获得的多样性可以通过保持惩罚一段时间来得到加强，然后取消惩罚，禁忌搜索继续按照正常的评价规则进行。另一种形式采用频率矩阵，使用两种长期记忆，一种是基于最小频率的长期记忆，另一种是基于最大频率的长期记忆。通过使用基于最小频率的长期记忆，可以在未搜索的区域产生新的序列;而使用基于最大频率的长期记忆，可以在过去的搜索中认为是好的可行区域内产生不同的序列。在整个搜索过程中频率矩阵被不断的修改。

### 3.3算法流程



如果以整个邻域为候选解集以目标函数作为适值函数,以给定最大代数NG为停止准则,以优于历史最优解为渴望水平,则算法的流程如左图所示。



如果以整个邻域为候选解集以目标函数作为适值函数,以给定最大代数NG为停止准则,以优于历史最优解为渴望水平,则算法的流程如左图所示。

可见,如果禁忌搜索算法中的各环节采取的策略不同,得到的程序流程图也是不同的,究竟采取哪种策略,应该根据问题的具体情况确定。

同传统的优化算法相比,禁忌搜索算法具有如下优点:

①能接受劣解,具有很好的爬山能力。

②区域集中搜索与全局分散搜索能较好平衡。

但是,以上介绍的基本禁忌搜索也有明显的不足:

迭代到最优解,一个较差的初始解可能会极大地降低搜索质量。

②搜索过程是串行的,不像遗传算法那样具有并行的搜索机制。

①对初始解和邻域结构有较大的依赖性,一个好的初始解可能很快

## 算法性能的改进

禁忌搜索算法具有全局寻优能力,而且比较容易实现,自从20世纪90年代就引起了广泛的重视。但是应用中也发现,以上基本的禁忌搜索算法有一些缺点,对于给定的实际工程问题,可能需要大量的调试工作才能得到较好的效果,于是提出一些改进做法。下面介绍几种比较主要的改进,包括并行禁忌搜索算法、主动禁忌搜索算法以及禁忌搜索算法和其他算法的混合策略等。

# 模拟退火算法

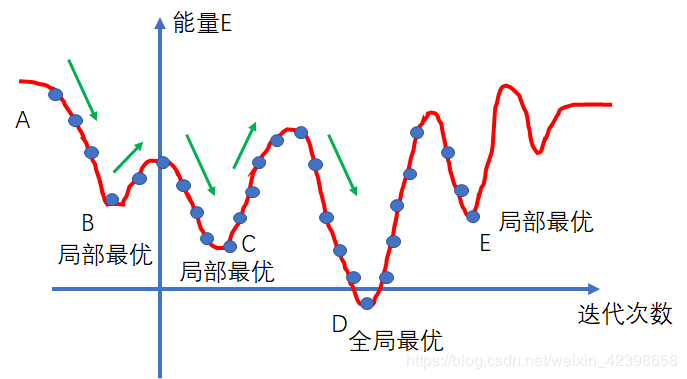
## 模拟退火算法的简介

为了解决局部最优解问题， 1983年，Kirkpatrick等提出了模拟退火算法（SA）能有效的解决局部最优解问题。我们知道在分子和原子的世界中，能量越大，意味着分子和原子越不稳定，当能量越低时，原子越稳定。‘退火’是物理学术语，指对物体加温在冷却的过程。模拟退火算法来源于晶体冷却的过程，如果固体不处于最低能量状态，给固体加热再冷却，随着温度缓慢下降，固体中的原子按照一定形状排列，形成高密度、低能量的有规则晶体，对应于算法中的全局最优解。而如果温度下降过快，可能导致原子缺少足够的时间排列成晶体的结构，结果产生了具有较高能量的非晶体，这就是局部最优解。因此就可以根据退火的过程，给其在增加一点能量，然后在冷却，如果增加能量，跳出了局部最优解，这本次退火就是成功的。

## 模拟退火算法的实现

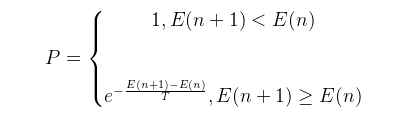
### 2.1模拟退火算法的基本原理

模拟退火算法包含两个部分即Metropolis算法和退火过程。Metropolis算法就是如何在局部最优解的情况下让其跳出来，是退火的基础。1953年Metropolis提出重要性采样方法，即以概率来接受新状态，而不是使用完全确定的规则，称为Metropolis准则，计算量较低。下面先形象的说一下，然后在解释数学公式：



假设开始状态在A，随着迭代次数更新到B的局部最优解，这时发现更新到B时，能力比A要低，则说明接近最优解了，因此百分百转移，状态到达B后，发现下一步能量上升了，如果是梯度下降则是不允许继续向前的，而这里会以一定的概率跳出这个坑，这各概率和当前的状态、能量等都有关系，下面会详细说，如果B最终跳出来了到达C，又会继续以一定的概率跳出来，可能有人会迷惑会不会跳回之前的B呢？下面会解释，直到到达D后，就会稳定下来。所以说这个概率的设计是很重要的，下面从数学方面进行解释。

假设前一个状态为x(n),系统根据某一指标（梯度下降，上节的能量），状态变为x(n+1),相应的，系统的能量由E(n)变为E(n+1),定义系统由x(n)变为x(n+1)的接受概率P为：



从上式我们可以看到，如果能量减小了，那么这种转移就被接受（概率为1），如果能量增大了，就说明系统偏离从上式我们可以看到，如果能量减小了，那么这种转移就被接受（概率为1），如果能量增大了，就说明系统偏离全局最优值位置更远了，此时算法不会立刻将其抛弃，而是进行概率操作：首先在区间[0,1]产生一个均匀分布的随机数，如果P，则此种转移接受，否则拒绝转移，进入下一步，往复循环。其中P以能量的变化量和T进行决定概率P的大小，所以这个值是动态的。全局最优值位置更远了，此时算法不会立刻将其抛弃，而是进行概率操作：首先在区间[0,1]产生一个均匀分布的随机数，如果P，则此种转移接受，否则拒绝转移，进入下一步，往复循环。其中P以能量的变化量和T进行决定概率P的大小，所以这个值是动态的。

### 2.2退火算法的参数控制

Metropolis算法是模拟退火算法的基础，但是直接使用Metropolis算法 可能会导致寻优速度太慢，以至于无法实际使用，为了确保在有限的时间收敛，必须设定控制算法收敛的参数，在上面的公式中，可以调节的参数就是T，T如果过大，就会导致退火太快，达到局部最优值就会结束迭代，如果取值较小，则计算时间会增加，实际应用中采用退火温度表，在退火初期采用较大的T值，随着退火的进行，逐步降低，具体如下：

（1）初始的温度T(0)应选的足够高，使的所有转移状态都被接受。初始温度越高，获得高质量的解的概率越大，耗费的时间越长。

（2） 退火速率。 最简单的下降方式是指数式下降：



其中是小于1的正数，一般取值为0.8到0.99之间。使的对每一温度，有足够的转移尝试，指数式下降的收敛速度比较慢，其他下降方式如下：

T(n) = \frac{T(0)}{log(1+t)} T(n) = \frac{T(0)}{1+t}

（3）终止温度

如果在若干次迭代的情况下每有可以更新的新状态或者达到用户设定的阈值，则退火完成。

### 2.3模拟退火的步骤

1.模拟退火算法可以分解为解空间、目标函数和初始解三部分。

2.模拟退火的基本思想:

(1) 初始化：初始温度T(充分大)，初始解状态S(是算法迭代的起点)，每个T值的迭代次数L

(2) 对k=1, …, L做第(3)至第6步：

(3) 产生新解S′

(4) 计算增量ΔT=C(S′)-C(S)，其中C(S)为代价函数

(5) 若ΔT<0则接受S′作为新的当前解，否则以概率exp(-ΔT/T)接受S′作为新的当前解.

(6) 如果满足终止条件则输出当前解作为最优解，结束程序。

终止条件通常取为连续若干个新解都没有被接受时终止算法。

(7) T逐渐减少，且T->0，然后转第2步。

模拟退火算法新解的产生和接受可分为如下四个步骤：

第一步是由一个产生函数从当前解产生一个位于解空间的新解；为便于后续的计算和接受，减少算法耗时，通常选择由当前新解经过简单地变换即可产生新解的方法，如对构成新解的全部或部分元素进行置换、互换等，注意到产生新解的变换方法决定了当前新解的邻域结构，因而对冷却进度表的选取有一定的影响。

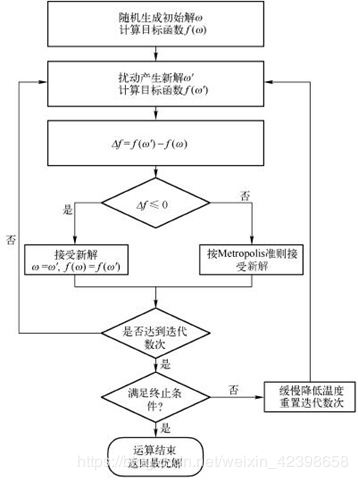
第二步是计算与新解所对应的目标函数差。因为目标函数差仅由变换部分产生，所以目标函数差的计算最好按增量计算。事实表明，对大多数应用而言，这是计算目标函数差的最快方法。

第三步是判断新解是否被接受,判断的依据是一个接受准则，最常用的接受准则是Metropolis准则: 若ΔT<0则接受S′作为新的当前解S，否则以概率exp(-ΔT/T)接受S′作为新的当前解S。

第四步是当新解被确定接受时，用新解代替当前解，这只需将当前解中对应于产生新解时的变换部分予以实现，同时修正目标函数值即可。此时，当前解实现了一次迭代。可在此基础上开始下一轮试验。而当新解被判定为舍弃时，则在原当前解的基础上继续下一轮试验。

模拟退火算法与初始值无关，算法求得的解与初始解状态S(是算法迭代的起点)无关；模拟退火算法具有渐近收敛性，已在理论上被证明是一种以概率l 收敛于全局最优解的全局优化算法；模拟退火算法具有并行性。

### 2.4退火算法程序流程图



### 2.5模拟退火算法的优点

迭代搜索效率高，并且可以并行化；

算法中有一定概率接受比当前解较差的解，因此一定程度上可以跳出局部最优；

算法求得的解与初始解状态S无关，因此有一定的鲁棒性；

具有渐近收敛性，已在理论上被证明是一种以概率l 收敛于全局最优解的全局优化算法。

# 蚁群算法

## 蚁群算法的简介

蚁群算法又称蚂蚁算法，是一种用来在图中寻找优化路径的机率型算法。它由Marco Dorigo于1992年在他的博士论文中提出，其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。蚁群算法是一种模拟进化算法，初步的研究表明该算法具有许多优良的性质。针对PID控制器参数优化设计问题，将蚁群算法设计的结果与遗传算法设计的结果进行了比较，数值仿真结果表明，蚁群算法具有一种新的模拟进化优化方法的有效性和应用价值。

各个蚂蚁在没有事先告诉他们食物在什么地方的前提下开始寻找食物。当一只找到食物以后，它会向环境释放一种挥发性分泌物pheromone (称为信息素,该物质随着时间的推移会逐渐挥发消失，信息素浓度的大小表征路径的远近)来实现的，吸引其他的蚂蚁过来，这样越来越多的蚂蚁会找到食物。有些蚂蚁并没有像其它蚂蚁一样总重复同样的路，他们会另辟蹊径，如果另开辟的道路比原来的其他道路更短，那么，渐渐地，更多的蚂蚁被吸引到这条较短的路上来。最后，经过一段时间运行，可能会出现一条最短的路径被大多数蚂蚁重复着。

蚁群算法是一种仿生学算法，是由自然界中蚂蚁觅食的行为而启发的。在自然界中，蚂蚁觅食过程中，蚁群总能够按照寻找到一条从蚁巢和食物源的最优路径。

## 蚁群算法的特点

### 2.1蚁群算法是一种自组织的算法。

在系统论中，自组织和它组织是组织的两个基本分类，其区别在于组织力或组织指令是来自于系统的内部还是来自于系统的外部，来自于系统内部的是自组织，来自于系统外部的是他组织。如果系统在获得空间的、时间的或者功能结构的过程中，没有外界的特定干预，我们便说系统是自组织的。在抽象意义上讲，自组织就是在没有外界作用下使得系统熵减小的过程(即是系统从无序到有序的变化过程)。蚁群算法充分体现了这个过程，以蚂蚁群体优化为例子说明。当算法开始的初期，单个的人工蚂蚁无序的寻找解，算法经过一段时间的演化，人工蚂蚁间通过信息激素的作用，自发的越来越趋向于寻找到接近最优解的一些解，这就是一个无序到有序的过程。

### 2.2蚁群算法是一种本质上并行的算法。

每只蚂蚁搜索的过程彼此独立，仅通过信息激素进行通信。所以蚁群算法则可以看作是一个分布式的多agent系统，它在问题空间的多点同时开始进行独立的解搜索，不仅增加了算法的可靠性，也使得算法具有较强的全局搜索能力。

### 2.3蚁群算法是一种正反馈的算法。

从真实蚂蚁的觅食过程中我们不难看出，蚂蚁能够最终找到最短路径，直接依赖于最短路径上信息激素的堆积，而信息激素的堆积却是一个正反馈的过程。对蚁群算法来说，初始时刻在环境中存在完全相同的信息激素，给予系统一个微小扰动，使得各个边上的轨迹浓度不相同，蚂蚁构造的解就存在了优劣，算法采用的反馈方式是在较优的解经过的路径留下更多的信息激素，而更多的信息激素又吸引了更多的蚂蚁，这个正反馈的过程使得初始的不同得到不断的扩大，同时又引导整个系统向最优解的方向进化。因此，正反馈是蚂蚁算法的重要特征，它使得算法演化过程得以进行。

### 2.4蚁群算法具有较强的鲁棒性。

相对于其它算法，蚁群算法对初始路线要求不高，即蚁群算法的求解结果不依赖于初始路线的选择，而且在搜索过程中不需要进行人工的调整。其次，蚁群算法的参数数目少，设置简单，易于蚁群算法应用到其它组合优化问题的求解。

### 2.4蚁群算法为什么有智能行为？

蚂蚁之所以具有智能行为，完全归功于它的简单行为规则，而这些规则综合起来具有下面两个方面的特点：多样性、正反馈。

多样性保证了蚂蚁在觅食的时候不至走进死胡同而无限循环，正反馈机制则保证了相对优良的信息能够被保存下来。我们可以把多样性看成是一种创造能力，而正反馈是一种学习强化能力。正反馈的力量也可以比喻成权威的意见，而多样性是打破权威体现的创造性，正是这两点小心翼翼的巧妙结合才使得智能行为涌现出来了。

引申来讲，大自然的进化，社会的进步、人类的创新实际上都离不开这两样东西，多样性保证了系统的创新能力，正反馈保证了优良特性能够得到强化，两者要恰到好处的结合。如果多样性过剩，也就是系统过于活跃，这相当于蚂蚁会过多的随机运动，它就会陷入混沌状态；而相反，多样性不够，正反馈机制过强，那么系统就好比一潭死水。这在蚁群中来讲就表现为，蚂蚁的行为过于僵硬，当环境变化了，蚂蚁群仍然不能适当的调整。

既然复杂性、智能行为是根据底层规则涌现的，既然底层规则具有多样性和正反馈特点，那么也许你会问这些规则是哪里来的？多样性和正反馈又是哪里来的？我本人的意见：规则来源于大自然的进化。而大自然的进化根据刚才讲的也体现为多样性和正反馈的巧妙结合。而这样的巧妙结合又是为什么呢？为什么在你眼前呈现的世界是如此栩栩如生呢？答案在于环境造就了这一切，之所以你看到栩栩如生的世界，是因为那些不能够适应环境的多样性与正反馈的结合都已经死掉了，被环境淘汰了！

## 构成要素

### 3.1范围

蚂蚁观察到的范围是一个方格世界，蚂蚁有一个参数为速度半径（一般是3），那么它能观察到的范围就是3\*3个方格世界，并且能移动的距离也在这个范围之内。

### 3.2环境

蚂蚁所在的环境是一个虚拟的世界，其中有障碍物，有别的蚂蚁，还有信息素，信息素有两种，一种是找到食物的蚂蚁洒下的食物信息素，一种是找到窝的蚂蚁洒下的窝的信息素。每个蚂蚁都仅仅能感知它范围内的环境信息。环境以一定的速率让信息素消失。

### 3.3觅食规则

在每只蚂蚁能感知的范围内寻找是否有食物，如果有就直接过去。否则看是否有信息素，并且比较在能感知的范围内哪一点的信息素最多，这样，它就朝信息素多的地方走，并且每只蚂蚁都会以小概率犯错误，从而并不是往信息素最多的点移动。蚂蚁找窝的规则和上面一样，只不过它对窝的信息素做出反应，而对食物信息素没反应。

### 3.4移动规则

每只蚂蚁都朝向信息素最多的方向移，并且，当周围没有信息素指引的时候，蚂蚁会按照自己原来运动的方向惯性的运动下去，并且，在运动的方向有一个随机的小的扰动。为了防止蚂蚁原地转圈，它会记住刚才走过了哪些点，如果发现要走的下一点已经在之前走过了，它就会尽量避开。

### 3.5避障规则

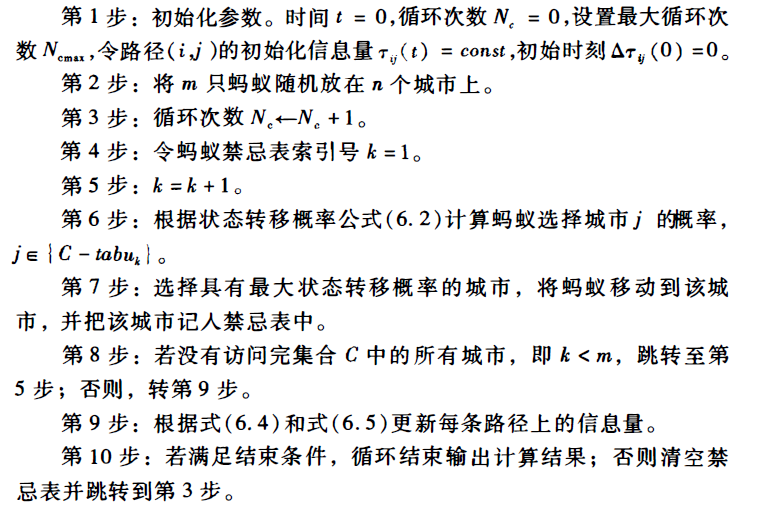
如果蚂蚁要移动的方向有障碍物挡住，它会随机的选择另一个方向，并且有信息素指引的话，它会按照觅食的规则行为。

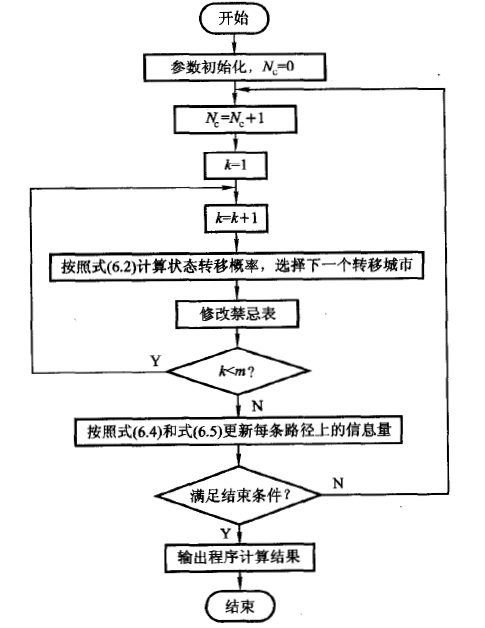
### 3.6信息素规则

每只蚂蚁在刚找到食物或者窝的时候撒发的信息素最多，并随着它走远的距离，播撒的信息素越来越少。

根据这几条规则，蚂蚁之间并没有直接的关系，但是每只蚂蚁都和环境发生交互，而通过信息素这个纽带，实际上把各个蚂蚁之间关联起来了。比如，当一只蚂蚁找到了食物，它并没有直接告诉其它蚂蚁这儿有食物，而是向环境播撒信息素，当其它的蚂蚁经过它附近的时候，就会感觉到信息素的存在，进而根据信息素的指引找到了食物。

## 算法实现





# 粒子群优化算法

## 粒子群优化算法的简介

粒子群算法（ Particle Swarm Optimization, PSO）最早是由Eberhart和Kennedy于1995年提出，它的基本概念源于对鸟群觅食行为的研究。设想这样一个场景：一群鸟在随机搜寻食物，在这个区域里只有一块食物，所有的鸟都不知道食物在哪里，但是它们知道当前的位置离食物还有多远。最简单有效的策略？寻找鸟群中离食物最近的个体来进行搜素。PSO算法就从这种生物种群行为特性中得到启发并用于求解优化问题。

粒子群算法的发展过程。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能解，通过粒子个体的简单行为，群体内的信息交互实现问题求解的智能性。由于PSO操作简单、收敛速度快，因此在函数优化、 图像处理、大地测量等众多领域都得到了广泛的应用。 随着应用范围的扩大，PSO算法存在早熟收敛、维数灾难、易于陷入局部极值等问题需要解决，主要有以下几种发展方向。

（1）调整PSO的参数来平衡算法的全局探测和局部开采能力。如Shi和Eberhart对PSO算法的速度项引入了惯性权重，并依据迭代进程及粒子飞行情况对惯性权重进行线性（或非线性）的动态调整，以平衡搜索的全局性和收敛速度。2009年张玮等在对标准粒子群算法位置期望及方差进行稳定性分析的基础上，研究了加速因子对位置期望及方差的影响，得出了一组较好的加速因子取值。

（2）设计不同类型的拓扑结构，改变粒子学习模式，从而提高种群的多样性，Kennedy等人研究了不同的拓扑结构对SPSO性能的影响。针对SPSO存在易早熟收敛，寻优精度不高的缺点，于2003年提出了一种更为明晰的粒子群算法的形式：骨干粒子群算法（Bare Bones PSO，BBPSO）。

（3）将PSO和其他优化算法（或策略）相结合，形成混合PSO算法。如曾毅等将模式搜索算法嵌入到PSO算法中，实现了模式搜索算法的局部搜索能力与PSO算法的全局寻优能力的优势互补。

（4）采用小生境技术。小生境是模拟生态平衡的一种仿生技术，适用于多峰函数和多目标函数的优化问题。例如，在PSO算法中，通过构造小生境拓扑，将种群分成若干个子种群，动态地形成相对独立的搜索空间，实现对多个极值区域的同步搜索，从而可以避免算法在求解多峰函数优化问题时出现早熟收敛现象。 Parsopoulos提出一种基于“分而治之”思想的多种群PSO算法，其核心思想是将高维的目标函数分解成多个低维函数，然后每个低维的子函数由一个子粒子群进行优化，该算法对高维问题的求解提供了一个较好的思路。

不同的发展方向代表不同的应用领域，有的需要不断进行全局探测，有的需要提高寻优精度，有的需要全局搜索和局部搜索相互之间的平衡，还有的需要对高维问题进行求解。这些方向没有谁好谁坏的可比性，只有针对不同领域的不同问题求解时选择最合适的方法的区别。

## 粒子群优化算法的实现

### 2.1粒子群优化算法的基本原理

模拟系统利用局部信息从而可以产生不可预测的群行为。我们经常能够看到成群的鸟、鱼或者浮游生物。这些生物的聚集行为有利于它们觅食和逃避捕食者。它们的群落动辄以十、百、千甚至万计，并且经常不存在一个统一的指挥者。它们是如何完成聚集、移动这些功能呢？Millonas在开发人工生命算法时(1994年)，提出群体智能概念并提出五点原则：

1、接近性原则：群体应能够实现简单的时空计算；

2、优质性原则：群体能够响应环境要素；

3、变化相应原则：群体不应把自己的活动限制在一狭小范围；

4、稳定性原则：群体不应每次随环境改变自己的模式；

5、适应性原则：群体的模式应在计算代价值得的时候改变。

粒子群优化算法的基本思想是通过群体中个体之间的协作和信息共享来寻找最优解。用一种粒子来模拟个体，每个粒子可视为N维搜索空间中的一个搜索个体，粒子的当前位置即为对应优化问题的一个候选解，粒子的飞行过程即为该个体的搜索过程．粒子的飞行速度可根据粒子历史最优位置和种群历史最优位置进行动态调整．粒子仅具有两个属性：速度和位置，速度代表移动的快慢，位置代表移动的方向。每个粒子单独搜寻的最优解叫做个体极值，粒子群中最优的个体极值作为当前全局最优解。不断迭代，更新速度和位置。最终得到满足终止条件的最优解。

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、神经网络训练、模糊系统控制以及其他遗传算法的应用领域。

### 2.1算法描述：

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，粒子通过跟踪两个“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更新自己的速度和位置。

i＝1，2，…，M，M是该群体中粒子的总数；Vi 是粒子的速度； pbest和gbest如前定义； rand()是介于（0、1）之间的随机数； Xi 是粒子的当前位置。 和是学习因子，通常取==2 在每一维，粒子都有一个最大限制速度，如果 某一维的速度超过设定的 ，那么这一维的速度 就被限定为 。（ >0） 以上面两个公式为基础，形成了后来PSO 的标准形式。

### 2.3 算法优化

1998年shi等人在进化计算的国际会议上 发表了一篇论文《A modified particle swarm optimizer》对前面的公式进行了修正。引入 惯性权重因子。值较大，全局寻优能力强，局部寻优能力弱； 值较小反之。

初始时，shi将 取为常数，后来实验发现，动 态 能够获得比固定值更好的寻优结果。动态 可以在PSO搜索过程中线性变化，也可根据PSO 性能的某个测度函数动态改变。 目前，采用较多的是shi建议的线性递减权值 (linearly decreasing weight, LDW)策略。

### 2.4 标准PSO算法流程

Step1:初始化一群微粒(群体规模为m)，包括随机位置和 速度；

Step2:评价每个微粒的适应度；

Step3:对每个微粒，将其适应值与其经过的最好位置 pbest作比较，如果较好，则将其作为当前的 最好位置pbest;

Step4:对每个微粒，将其适应值与其经过的最好位置 gbest作比较，如果较好，则将其作为当前的 最好位置gbest;

Step5:根据(2)、(3)式调整微粒速度和位置；

Step6：未达到结束条件则转Step2。

迭代终止条件根据具体问题一般选为最大迭代次数Gk或(和)微粒群迄今为止搜索到的最优位置满足预定最小适应阈值。

3.5 参数分析

　　方程中pbest和gbest分别表示微粒群的局部和 全局最优位置，当＝0时，则粒子没有了认知能力， 变为只有社会的模型(social-only):

被称为全局PSO算法.粒子有扩展搜索空间的能力，具有 较快的收敛速度，但由于缺少局部搜索，对于复杂问题 比标准PSO 更易陷入局部最优。

当＝0时，则粒子之间没有社会信息，模型变为 只有认知(cognition-only)模型：

被称为局部PSO算法。由于个体之间没有信息的 交流，整个群体相当于多个粒子进行盲目的随机 搜索，收敛速度慢，因而得到最优解的可能性小。

群体规模m 一般取20～40，对较难或特定类别的问题 可以取到100～200。

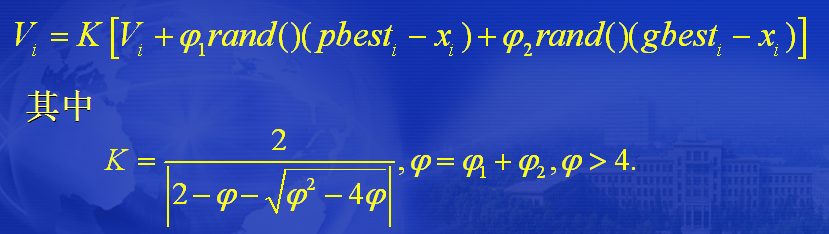
最大速度Vmax决定当前位置与最好位置之间的区域的 分辨率(或精度)。如果太快，则粒子有可能越过极小 点;如果太慢，则粒子不能在局部极小点之外进行足 够的探索，会陷入到局部极值区域内。这种限制可以 达到防止计算溢出、决定问题空间搜索的粒度的目的。

权重因子 包括惯性因子 和学习因子和。 使粒子 保持着运动惯性，使其具有扩展搜索空间的趋势，有 能力探索新的区域。和代表将每个粒子推向Pbest 和gbest位置的统计加速项的权值。较低的值允许粒子 在被拉回之前可以在目标区域外徘徊，较高的值导致粒 子突然地冲向或越过目标区域。

### 2.4.优化PSO

#### 2.4.1 引入收敛因子，不要惯性权重

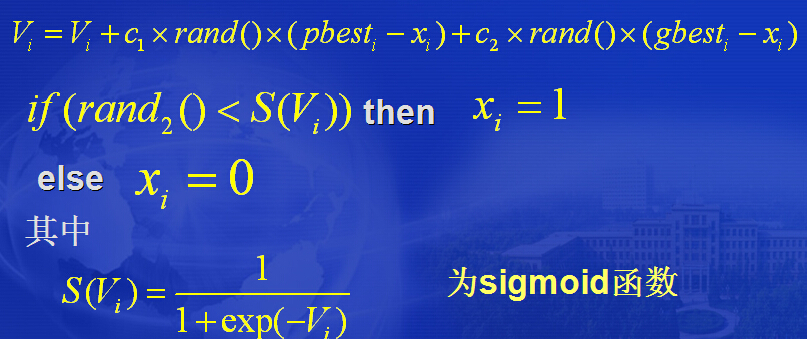
通常设=＝2。Suganthan的实验表明：和为常数时可以得到较好的解，但不一定必须等于2。 Clerc引入收敛因子(constriction factor) K来保证 收敛性。



通常取 为4.1,则K＝0.729.实验表明，与使 用惯性权重的PSO算法相比，使用收敛因子的 PSO有更快的收敛速度。其实只要恰当的选取 和、，两种算法是一样的。因此使用收 敛因子的PSO可以看作使用惯性权重PSO的特例。 恰当的选取算法的参数值可以改善算法的性能。

#### 2.4.2 离散二进制粒子群

基本PSO是用于实值连续空间，然而许多实际问题是组合 优化问题，因而提出离散形式的PSO。 速度和位置更新式为：



#### 2.4.3 PSO和GA比较

共性: （1）都属于仿生算法。 (2) 都属于全局优化方法。 (3) 都属于随机搜索算法。 (4) 都隐含并行性。 (5) 根据个体的适配信息进行搜索，因此不受函数 约束条件的限制，如连续性、可导性等。 (6) 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，都无法保证收敛到最优点。

差异： (1) PSO有记忆，好的解的知识所有粒子都保 存，而GA，以前的知识随着种群的改变被改变。 (2) PSO中的粒子仅仅通过当前搜索到最优点进行共享信息，所以很大程度上这是一种单共享项信息机制。而GA中，染色体之间相互共享信息，使得整个种群都向最优区域移动。 (3) GA的编码技术和遗传操作比较简单，而PSO 相对于GA，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。

GA可以用来研究NN的三个方面：网络连接权重、网络 结构、学习算法。优势在于可处理传统方法不能处理的 问题，例如不可导的节点传递函数或没有梯度信息。 缺点:在某些问题上性能不是特别好；网络权重的编码和 遗传算子的选择有时较麻烦。 已有利用PSO来进行神经网络训练。研究表明PSO是一 种很有潜力的神经网络算法。速度较快且有较好的结果。 且没有遗传算法碰到的问题。

# 基于改进遗传算法的路径规划问题研究

## 摘要

本课题研究基于改进遗传算法的机器人路径规划问题。对个体的编码方案和基本遗传操作等进行改进，并引入插入算法将离散路径点完善成一条连续无碰的路径。在总体流程上，仍然采用种群初始化、计算适应值、选择、交叉、变异等基本动作。仿真实验结果表明，本方法可以获得最优或较优解，达到了预期的效果。

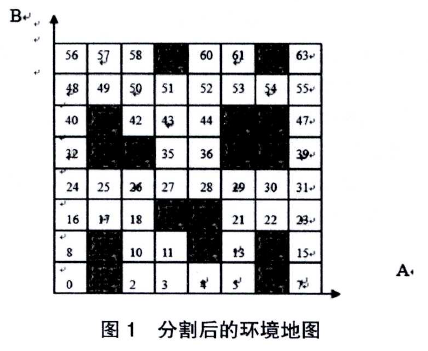
## 路径规划算法

### 2.1问题

(1)机器人如何利用传感器获取周围的障碍物信息和其他的相关信息来构建环境地图

(2)机器人如何根据自己目前所处的位置,来决定自己下下一步的行动策略

### 2.2构建基于栅格法的环境地图



将机器人运动的环境简化为二维空间。用栅格法将环境地图分成大小相等的格子,格数取决于环境和机器人本身的大小,要保证在机器人运动时不会因为机器人体积过大而碰到障碍物。分好格子后,按照从左到右、从下到上的顺序对其进行编号,使其分布在一个直角坐标系中。如左图所示,其中黑色格子表示障碍物栅格。

### 2.3基于遗传算法的路径规划思路描述

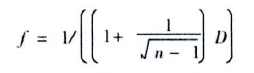
第一步是初始化种群,然后进行遗传基本操作:选择、交叉、变异和插入。经过若干次进化后,过程停止,输出得到的最优或较优个体。

#### 2.3.1种群初始化

在路径规划问题中,一个个体代表一条从起点到终点的路径。一定数量个体的集合就是种群。初始确定了起点和终点;首先随机生成离散点,将由离散点组成的个体变成一条连续的无碰路径;然后删去路径中的回路,使个体变成一条连续无碰且没有回路的路径;最后删去种群中相同的个体。

#### 2.3.2计算适应值

适应值用来衡量某个物种对于生存环境的适应程度。在路径规划问题中,就是度量一个个体是否优越。路径越长,有可能表示这条路径越来越被完善,通过所有相邻点的距离之和表示。因此选取适应值函数为:



其中n代表个体路过的栅格个数，D表示个体中所有相邻点的距离之和。

#### 2.3.3选择操作

选择操作是指以一定的概率从种群中选择若干个体,它是一种基于适应度的优胜劣汰的过程。最常用的是轮盘赌算法,转动轮盘,获得每次轮盘停止时指针的位置,指针停在某一扇区,该扇区代表的个体即被选中。适应值越高的个体被选中的几率越大。

#### 2.3.4交叉操作

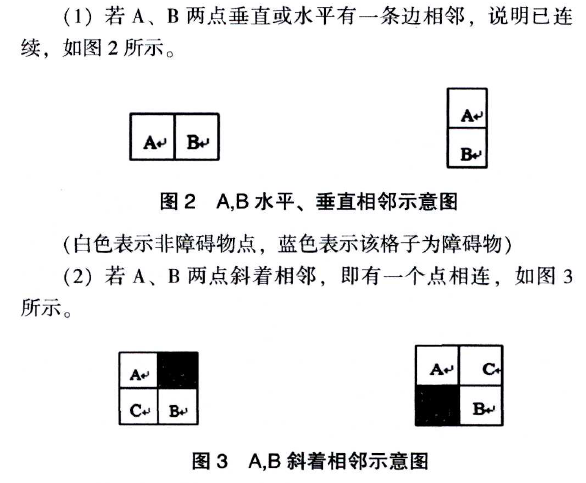
交叉操作是指生物在繁殖下一代时两个染色体之间通过交叉重组。也就是以一定的概率,在两个染色体的某一相同位置处切断DNA,前后两串分别交叉组合形成两个新的染色体,并替换原父代个体。

#### 2.3.5变异操作

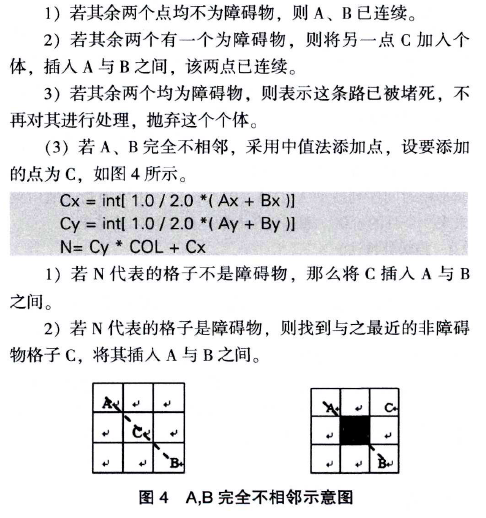
为避免造成近亲繁殖,可以在进化过程中加入有新遗传基因的个体,这种方法就是变异。根据自然界规律,变异的发生是小概率的。变异操作是将所有个体的每一位(除了第一位和最后一位)按照一定的概率p进行删除,代替父个体。

#### 2.3.6插入算法

在上述遗传算法的基本操作的基础上,引入插入算法的目的是将本来离散的路径点逐渐完善成一条连续的无障碍的路径。对于个体中任意两相邻点,做如下操作:



对于个体中每两个相邻的格子做如上操作，重复进行，直至该个体已经可以表示一条连续无碰撞的路径，成功；或者无法找到新的点．将该个体舍弃。



#### 2.3.7删除算法

经过插入箅法能够将本来离散的一组路径点完善成一条连续的路径．但有可能产生回路．删除算法就是将路径中的回路删去，使其进化成一条连续无碰且无同路的路径。

#### 2.3.8结束条件

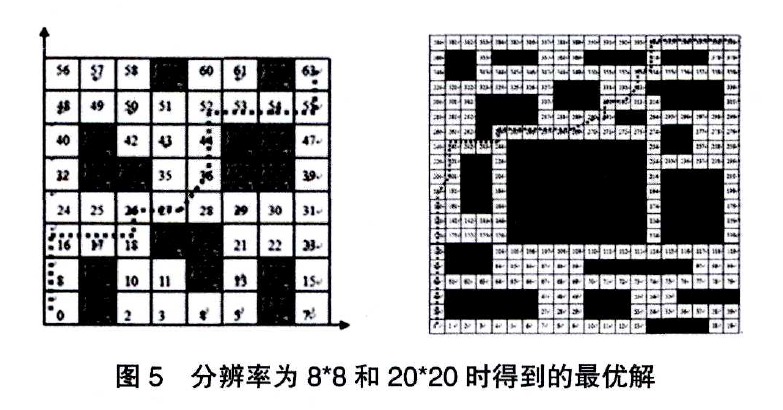
遗传算法是一种随机算法，虽然在宏观上有一定的方向性，仍然要人为设置一个明确的结束条件．这里用设定最大进化代数的方法，同时若种群内只剩一个个体，也强行终止。

## 仿真结果

本算法在VS2010上实现,交叉概率为60%,变异概率为1%。如下所示几个仿真实例,当环境地图的分辨率分别为8\*8、20\*20时,具体仿真结果如下:

1)当分辨率为8\*8时,设定初始种群规模为6,最大进化代数为100,进过三代进化得到的最优路径如图5所示。

2)当分辨率为20\*20时,设定初始种群规模为200,障碍物栅格174个,最大进化代数为100,得到的最优路径如图5所示。



## 结语

提出了基于改进遗传算法的路径规划方法,详细论述了构建环境地图和遗传操作,并引入插入算法将离散路径点完善成一条连续无碰的路径。最后分别在分辨率为8\*8、20\*20的栅格环境下进行了仿真。

# 船舶电力系统无功优化过程的自适应粒子群算法应用

## 摘要

随着现代船舶工业的需要,船载用电设备的数量越来越多,船舶电力系统的容量也越来越大。船载用电设备在提高船舶自动化水平的同时,也对船舶电力系统的稳定性、安全性、节能性等提出更高要求。其中,船舶电力系统网络的无功损耗问题一直是相关领域的研究重点。电力系统的无功优化是指当系统的结构参数、负载等保持不变的情况下,通过改善船舶电力系统电源的电压等参数,调节电网的潮流计算,无功优化对提高船舶电力系统的供电效率,降低事故发生率等有重要的作用。本文系统的介绍了自适应粒子群算法,并将该粒子群寻优算法应用到船舶电力系统的无功优化过程中,对于改善电力系统的无功优化过程有重要的意义。

## 自适应粒子群算法的电力系统无功优化

### 2.1无功优化数学模型的建立

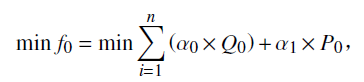
本文主要研究船舶电力系统的无功优化问题，建立船舶的无功优化模型如下：



式中:x为状态变量;f1为目标函数;f2为约束函数。

在船舶电力系统无功优化时,主要针对下面几个方面

1)系统的无功补偿量最小



式中:α0为某节点的无功补偿系数; Q0为补偿节点的容量; α1为有功补偿系数; P0为电力系统的有功损耗。

2)电力系统发电费用最低

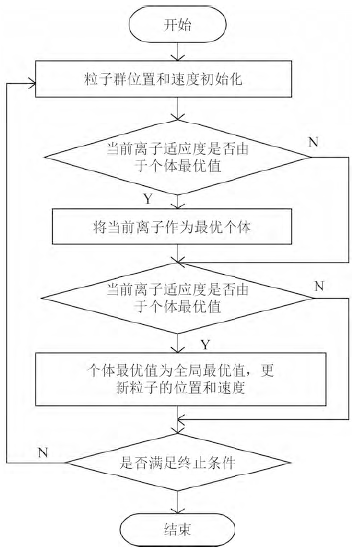
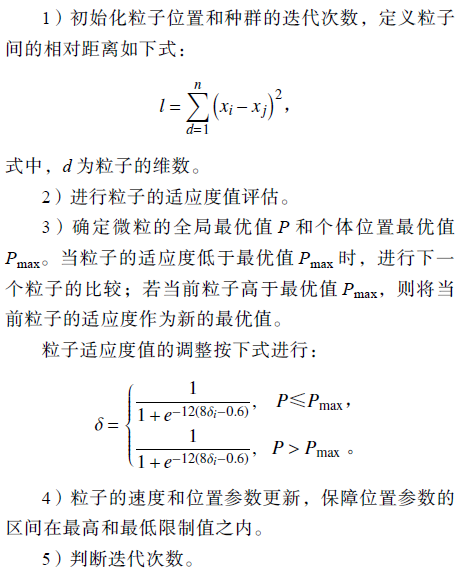


式中:G1为电力系统第i台发电机组的发电效率;P1为第i台发电机组的输出功率。

### 2.2自适应粒子群算法研究

随着计算机技术和控制算法的发展,自适应粒子群优化算法(PSO)逐渐发展起来。自适应粒子群优化算法的核心是群体中的信息提取和智能优化。

自适应粒群优化算法的流程如图2所示。

#### 2.3基于粒子群优化算法的电力系统无功优化

本文在电力系统无功优化数学模型的基础上,利用自适应粒子群算法对无功损耗进行了优化。

为了验证该优化算法的效果,本文采用了Matlab 软件平台对优化结果进行仿真,并定义仿真过程的惯性权重函数为Sphere 函数。该函数如下:



设定惯性权重的初始值ω=0.6,粒子的种群规模为N=4000,维数D=26。在Matlab 软件平台的优化前后电网的损耗仿真结果如图3所示。

图中曲线1为未优化的电网损耗,曲线2为基于自适应粒子群优化的电网损耗。可见,基于粒子群优化的电网无功损耗相对较小,且随着迭代次数的增加,两者的无功损耗均呈现减小的趋势。

## 结语

随着现代船舶工业发展,导航、照明、装卸等自动化设备的装机量越来越多,对船舶电力系统的要求也不断提高。电力系统的无功优化是通过改善船舶电力系统电源的电压等参数,提高船舶电力系统的供电效率,减低供电成本。本文系统的介绍了自适应粒子群算法的原理和步骤,并利用该算法对船舶电力系统进行了无功优化,并对优化结果做了基于Matlab 的仿真。