

# 计算智能——禁忌搜索

作者 柯良军

西安交通大学 电信学院

October 15, 2015

# 目录

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 1 基本思想
- 2 算法要素
- 3 算法步骤及流程图
- 4 例子
- 5 禁忌搜索的改进

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 局部搜索的局限

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 局部搜索的局限

## 禁忌的作用

# 基本思想

计算智能——禁忌搜索

柯良军

- 禁忌表

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 禁忌表

将算法搜索过程中最近的若干次移动加入禁忌表中，禁止在之后的迭代中进行移动，避免了重复搜索已经搜索过的邻域；

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 禁忌表

将算法搜索过程中最近的若干次移动加入禁忌表中，禁止在之后的迭代中进行移动，避免了重复搜索已经搜索过的邻域；

- 接受劣解

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- **禁忌表**

将算法搜索过程中最近的若干次移动加入禁忌表中，禁止在之后的迭代中进行移动，避免了重复搜索已经搜索过的邻域；

- **接受劣解**

将算法带入新的区域进行搜索

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 禁忌表

将算法搜索过程中最近的若干次移动加入禁忌表中，禁止在之后的迭代中进行移动，避免了重复搜索已经搜索过的邻域；

- 接受劣解

将算法带入新的区域进行搜索

- “解禁”与“破禁”

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- **禁忌表**

将算法搜索过程中最近的若干次移动加入禁忌表中，禁止在之后的迭代中进行移动，避免了重复搜索已经搜索过的邻域；

- **接受劣解**

将算法带入新的区域进行搜索

- **“解禁”与“破禁”**

“解禁”：算法循环过程中会不断更新禁忌表，一定次数的循环后，最早进入禁忌表的移动会从禁忌表中删除

# 基本思想

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- **禁忌表**

将算法搜索过程中最近的若干次移动加入禁忌表中，禁止在之后的迭代中进行移动，避免了重复搜索已经搜索过的邻域；

- **接受劣解**

将算法带入新的区域进行搜索

- **“解禁”与“破禁”**

“解禁”：算法循环过程中会不断更新禁忌表，一定次数的循环后，最早进入禁忌表的移动会从禁忌表中删除

“破禁”：当移动达到了渴望水平，不论它是否在禁忌表中，都被接受

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## ● 编码方法

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解
- 搜索空间和邻域结构

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解
- 搜索空间和邻域结构
- 适值函数

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解
- 搜索空间和邻域结构
- 适应值函数
- 禁忌表

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解
- 搜索空间和邻域结构
- 适值函数
- 禁忌表
- 选择策略

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解
- 搜索空间和邻域结构
- 适值函数
- 禁忌表
- 选择策略
- 渴望水平

# 算法要素

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

- 编码方法
- 初始解
- 搜索空间和邻域结构
- 适值函数
- 禁忌表
- 选择策略
- 渴望水平
- 停止准则

# 编码方法

禁忌搜索实现的过程中，需要选择合适的编码方法。编码就是将实际问题通过一种便于以代码表达的形式来描述。

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

# 编码方法

禁忌搜索实现的过程中，需要选择合适的编码方法。编码就是将实际问题通过一种便于以代码表达的形式来描述。

卫星测控资源调度问题，假设有10个任务需要调度，其中3号、5号、7号任务有冲突，1号、9号任务有冲突，可以有以下两种方法：（1）对于每个任务分别设置一个标志位，标志位的值为0或1，为0时表示不调度该任务，为1时表示调度该任务，0-1方式的编码如下图1：

1	2	3	4	5	6	7	8	9	10
1	1	0	1	1	1	0	1	0	1

图：求解卫星资源调度问题是的0-1编码。

由图1可见，上面的编码即为1101110101，表示3号、7号、9号任务没有调度，其他都调度了

计算智

能——禁忌搜索

索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

# 编码方法

计算智能——禁忌搜索

柯良军

目录

基本思想

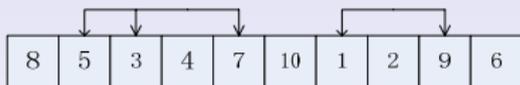
算法要素

算法步骤及流程图

例子

禁忌搜索的改进

(2) 由于任务之间的冲突情况是固定的，故可以根据任务号的不同排序表示不同的解，产生解时依照任务号的顺序依次调度，如果当前任务号与之前已调度的任务不冲突，则调度；否则，不调度。



图：求解卫星资源调度问题的顺序编码。

此编码方法中（见图2），在冲突的任务号之间会调度5号、1号任务，3号、7号、9号任务不会被调度。

# 初始解

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索算法是一个单点法，其性能依赖于初始解质量。

# 初始解

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索算法是一个单点法，其性能依赖于初始解质量。  
当初始解质量较好时，算法能够较快的收敛，且结果质量较好；

# 初始解

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索算法是一个单点法，其性能依赖于初始解质量。当初始解质量较好时，算法能够较快的收敛，且结果质量较好；反之，初始解质量较差时，会降低禁忌搜索的收敛速度。

# 初始解

计算智  
能——禁忌搜  
索

柯良军

目录

基本思想

算法要素

算法步骤及流  
程图

例子

禁忌搜索的改  
进

禁忌搜索算法是一个单点法，其性能依赖于初始解质量。当初始解质量较好时，算法能够较快的收敛，且结果质量较好；

反之，初始解质量较差时，会降低禁忌搜索的收敛速度。在求解一些约束条件较复杂的组合优化问题时，如果采用随机法一般难以得到质量较好的解，甚至也难以得到一个可行解。

一个解决方法是：先利用问题的结构信息应用一些简单的算法得到一个较好初始解，再用禁忌搜索求解，从而提高算法的求解质量和效率。

# 搜索空间和邻域结构

计算智能——禁忌搜索

柯良军

搜索空间是由搜索过程中所有可能解组成的集合。

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

# 搜索空间和邻域结构

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

搜索空间是由搜索过程中所有可能解组成的集合。  
邻域结构与搜索空间密切相关。在禁忌搜索中，当前解 $S$ 的邻域结构 $\mathcal{N}(S)$ 由其所有局部修正组成。

# 搜索空间和邻域结构

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

搜索空间是由搜索过程中所有可能解组成的集合。

邻域结构与搜索空间密切相关。在禁忌搜索中，当前解 $S$ 的邻域结构 $\mathcal{N}(S)$ 由其所有局部修正组成。

邻域结构 $\mathcal{N}(S)$ 由搜索空间的如下子集定义：针对特定的待求解问题，往往有许多邻域结构的定义方法。

**例：** 在车辆路径问题中，当前解 $S$ 邻域结构可以定义为由如下方式得到的解组成的集合：将某个客户从其当前路径中删除，并插入到当前路径或另外一条路径中得到的解。而在选择客户进行删除时，可以采用随机法、最佳法（选择改变目标函数值最大的删除）等方式来删除；在插入时，有多种方式，比如，随机方式，最佳方式（选择改变目标函数值最小的插入），基于后悔值方式等。也可以定义更为复杂邻域结构。比如 $\lambda$ -交换，交叉等。这些邻域结构在求解具有时间窗约束的车辆路径问题时尤为有用。在应用禁忌搜索求解问题时，选择和定义搜索空间与邻域结构是其中的关键步骤，这要求设计算法时要对问题有充分的认识 and 了解。

# 适值函数

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

适值函数是对搜索结果的评价。

# 适值函数

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

适值函数是对搜索结果的评价。

一般而言，以目标函数直接作为适值函数是比较常用的方法。

# 适值函数

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

适值函数是对搜索结果的评价。

一般而言，以目标函数直接作为适值函数是比较常用的方法。

当目标函数的计算比较困难时，可以对**目标函数进行适当的变形作为适值函数**以便于计算，从而节省计算时间，但变形后的适值函数必须是严格单调的，且适值函数的最优性与目标函数的最优性必须一致。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表是禁忌搜索算法区别于局部搜索的关键，是禁忌搜索算法的核心。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表是禁忌搜索算法区别于局部搜索的关键，是禁忌搜索算法的核心。

为能使算法跳出局部极小点，需要接受非改进移动。禁忌表的作用在于避免循环，换言之，避免接受一个已经经历过的解。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表是禁忌搜索算法区别于局部搜索的关键，是禁忌搜索算法的核心。

为能使算法跳出局部极小点，需要接受非改进移动。禁忌表的作用在于避免循环，换言之，避免接受一个已经经历过的解。

从数据结构上讲，禁忌表是具有一定长度的先进先出的队列。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表是禁忌搜索算法区别于局部搜索的关键，是禁忌搜索算法的核心。

为能使算法跳出局部极小点，需要接受非改进移动。禁忌表的作用在于避免循环，换言之，避免接受一个已经经历过的解。

从数据结构上讲，禁忌表是具有一定长度的先进先出的队列。

比如，在求解车辆路径问题过程中，顾客 $v_1$ 从路径 $R_1$ 插入到路径 $R_2$ ，此时，如果在以后的若干代中避免将 $v_1$ 从路径 $R_2$ 插回到路径 $R_1$ ，就能实现避免循环。总之，通过避免访问以前搜索的区域，禁忌能使算法探索新的区域。在搜索过程中，禁忌对象保存在禁忌表中，而且通常只需要保存有限的信息。

# 禁忌表

计算智能——禁忌搜索

柯良军

在设计禁忌表时，要考虑对什么禁忌，以及禁忌多久。

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计禁忌表时，要考虑对什么禁忌，以及禁忌多久。

## 1) 禁忌对象

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计禁忌表时，要考虑对什么禁忌，以及禁忌多久。

## 1) 禁忌对象

一般地，禁忌对象主要有以下几类：

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计禁忌表时，要考虑对什么禁忌，以及禁忌多久。

## 1) 禁忌对象

一般地，禁忌对象主要有以下几类：

(a) **解本身或解变化**：将移动或从当前解到新解的改变作为对象。禁忌这类对象的不足在于存储大而且在检查一次操作是否在禁忌表中是费时的，因此，实际中很少采用这种方案；

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计禁忌表时，要考虑对什么禁忌，以及禁忌多久。

## 1) 禁忌对象

一般地，禁忌对象主要有以下几类：

(a) **解本身或解变化**：将移动或从当前解到新解的改变作为对象。禁忌这类对象的不足在于存储大而且在检查一次操作是否在禁忌表中是费时的，因此，实际中很少采用这种方案；

(b) **解的分量或解变化的分量**：当一个移动导致多个分量变化时，可以选其中若干个分量作为禁忌对象。这种方法禁忌范围较大，且存储量小，检查禁忌表的时间少。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计禁忌表时，要考虑对什么禁忌，以及禁忌多久。

## 1) 禁忌对象

一般地，禁忌对象主要有以下几类：

(a) **解本身或解变化**：将移动或从当前解到新解的改变作为对象。禁忌这类对象的不足在于存储大而且在检查一次操作是否在禁忌表中是费时的，因此，实际中很少采用这种方案；

(b) **解的分量或解变化的分量**：当一个移动导致多个分量变化时，可以选其中若干个分量作为禁忌对象。这种方法禁忌范围较大，且存储量小，检查禁忌表的时间少。

(c) **目标函数值**：将目标函数值相同的解作为禁忌对象。这种方法禁忌范围大。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在求解车辆路径问题时，可以禁忌许多类型的对象。

- 禁忌整个解；
- 记录少量的变换信息并将逆变换作为禁忌对象。

例如将禁忌对象记为三元组 $(v_1, R_2, R_1)$ 并保存到禁忌表中。不过，这种禁忌对象可能达不到既定目的。比如如果 $v_1$ 先移动到 $R_3$ ，再移动到 $R_1$ 。此时，又可能会导致循环。另一种定义禁忌对象的方法是禁止 $v_1$ 移动到 $R_1$ ，采用二元组 $(v_1, R_1)$ 来记录该禁忌对象。也可以定义禁忌对 $v_1$ 进行任何操作，采用一元组 $v_1$ 。这三种定义方式逐渐增强，限制更多的操作。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 2) 禁忌表个数

在算法运行时，可以考虑多个禁忌表，即对不同类型的操作定义不同的禁忌表。

# 禁忌表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 3) 禁忌长度

基本禁忌搜索算法中，禁忌长度一般较短，功能和人类的短期记忆相似，也称为“短期表”。实际应用中，禁忌表的长度可以是不变的，也可以是与问题规模相关的一个量，甚至可以是动态变化。研究表明，动态设置的禁忌表比静态设置的禁忌表具有更好的性能。

# 渴望水平

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索中，某些情况下，仅仅采用禁忌表会禁止一些可能得到高质量解得操作，甚至会导致算法停滞（即陷入局部最优）。因此，有必要采用措施以取消禁忌。即当某个移动满足某个条件时，不论该移动是否在禁忌表中，都接受这个移动，并更新当前解和当前最优解。这种使移动不受禁忌表禁忌的条件称为渴望水平，或称为**特赦准则**、**藐视准则**。

# 渴望水平

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 渴望水平的设定方法：

### (1) 基于适配值的原则

如果一个操作能得到新的适配值优于历史最优解 (best so far) 的解，则无论该操作是否被禁忌，都被接受并执行。这种设定是最简单且最常见的设定方法，可以理解为算法搜索到了一个更好的解。但是这种设定方法比较片面，有可能错过一些当前解比历史最优解差，但迭代几步之后会优于历史最优解的区域。因此，又出现了其他的渴望水平的设定。

# 渴望水平

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## (2) 基于搜索方向的原则

如果操作上次被禁忌时改善了适配值，而且这次这个被禁忌的操作相应的候选解适配值又优于当前解，那么这个操作不被禁忌。这个原则可以有效的避免算法在搜索过程中反复来回搜索。

# 渴望水平

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## (3) 基于影响力的原则

在算法搜索过程中，不同的操作对适配值的影响一般不同。可以结合操作进入禁忌表的时间、适配值和操作对适配值的影响，综合起来设定渴望水平。这种方法的简单理解为：解禁一个对适配值影响较大的操作，可能会使算法更快地搜索到一个更好的解。

渴望水平的设定比较灵活，设计算法时应结合实际问题进行考虑，可以采用以上介绍的一种或者多种原则综合。禁忌表和渴望水平是禁忌搜索算法的核心，两者是一对矛盾体，禁忌是为了避免循环，而没有循环发生时则可以不禁忌。

# 停止准则

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

和遗传算法等其他启发式算法类似，禁忌搜索无法确定是否找到全局最优解，故无法自动停止搜索。因此需要人为设定停止准则来停止搜索。往往采用如下停止条件：

- 在给定运行代数或运行时间后停止；
- 在最优当前解连续给定代没有改进时终止；
- 在当前最优解满足给定优性条件时终止。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索的基本步骤为：

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索的基本步骤为：

(1) 产生初始解，清空禁忌表。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索的基本步骤为：

- (1) 产生初始解，清空禁忌表。
- (2) 判断是否满足终止准则。如果满足，停止搜索，输出当前最优解；否则，转第(3)步。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌搜索的基本步骤为：

- (1) 产生初始解，清空禁忌表。
- (2) 判断是否满足终止准则。如果满足，停止搜索，输出当前最优解；否则，转第(3)步。
- (3) 进行邻域搜索，找到邻域内的候选解，计算其目标函数值，。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 禁忌搜索的基本步骤为：

- (1) 产生初始解，清空禁忌表。
- (2) 判断是否满足终止准则。如果满足，停止搜索，输出当前最优解；否则，转第(3)步。
- (3) 进行邻域搜索，找到邻域内的候选解，计算其目标函数值，。
- (4) 依次选择当前最优候选解，判断是否达到渴望水平。如果满足，转第(5)步；否则，转第(6)步。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 禁忌搜索的基本步骤为：

- (1) 产生初始解，清空禁忌表。
- (2) 判断是否满足终止准则。如果满足，停止搜索，输出当前最优解；否则，转第(3)步。
- (3) 进行邻域搜索，找到邻域内的候选解，计算其目标函数值，。
- (4) 依次选择当前最优候选解，判断是否达到渴望水平。如果满足，转第(5)步；否则，转第(6)步。
- (5) 更新当前解和渴望水平，继续下一步。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 禁忌搜索的基本步骤为：

- (1) 产生初始解，清空禁忌表。
- (2) 判断是否满足终止准则。如果满足，停止搜索，输出当前最优解；否则，转第(3)步。
- (3) 进行邻域搜索，找到邻域内的候选解，计算其目标函数值，。
- (4) 依次选择当前最优候选解，判断是否达到渴望水平。如果满足，转第(5)步；否则，转第(6)步。
- (5) 更新当前解和渴望水平，继续下一步。
- (6) 若当前最优候选解的移动被禁忌，转第(4)步；否则，继续下一步。

# 基本步骤

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

## 禁忌搜索的基本步骤为：

- (1) 产生初始解，清空禁忌表。
- (2) 判断是否满足终止准则。如果满足，停止搜索，输出当前最优解；否则，转第(3)步。
- (3) 进行邻域搜索，找到邻域内的候选解，计算其目标函数值，。
- (4) 依次选择当前最优候选解，判断是否达到渴望水平。如果满足，转第(5)步；否则，转第(6)步。
- (5) 更新当前解和渴望水平，继续下一步。
- (6) 若当前最优候选解的移动被禁忌，转第(4)步；否则，继续下一步。
- (7) 更新当前解、当前最优解、禁忌表，转第(2)步。

# 流程图

计算智能——禁忌搜索

柯良军

目录

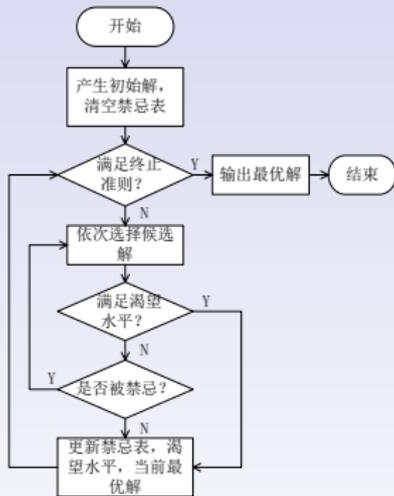
基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进



# 算例的信息

计算智  
能——禁忌搜  
索

柯良军

目录

基本思想

算法要素

算法步骤及流  
程图

例子

禁忌搜索的改  
进

# 实验设置

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

定义邻域映射为对换两个城市位置，选定A城市为起点和终点，设置禁忌表长度为3，迭代次数为5次。

# 第一步

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

令初始解为  $x_0 = (ABCDE)$ ，目标值  $f(x_0) = 55$ ，历史最优值为55，禁忌表为空。交换任意两个城市间的位置（A城市不变），得到新的解得目标函数与当前目标值的差。

禁忌表		移动	适配值差值
1	∅	B,C	-2
2	∅	D,E	-1
3	∅	B,D	0
		C,E	14
		B,E	15
		C,D	15

图：第一步禁忌表、移动和适配值差值信息。

# 第二步

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表		移动	适配值差值
1	<i>B,C</i>	<i>D,E</i>	0
2	∅	<i>C,D</i>	1
3	∅	<i>B,E</i>	15
		<i>C,E</i>	16
		<i>B,D</i>	16

图：第二步禁忌表、移动和适配值差值信息。

# 第二步

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表		移动	适配值差值
1	$B,C$	$D,E$	0
2	$\emptyset$	$C,D$	1
3	$\emptyset$	$B,E$	15
		$C,E$	16
		$B,D$	16

图：第二步禁忌表、移动和适配值差值信息。

由上图可以看出，对目标函数值改善最大的移动是 $(B,C)$ ，且这个移动不在禁忌表中，所以选择此移动。此时，当前解为 $x_1 = (ACBDE)$ ，目标函数值为 $f(x_1) = 53$ ，历史最优值为53，将移动 $(B,C)$ 加入禁忌表中。与第一步相同，进行新一次交换。

# 第五步

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

禁忌表		移动	适配值差值
1	<i>B,D</i>	<i>C,B</i>	-2
2	<i>C,E</i>	<i>D,C</i>	15
3	<i>D,E</i>	<i>E,B</i>	15

图：第五步禁忌表、移动和适配值差值信息。

第四步中，所有移动都不能改善当前目标函数值，选择移动(*B,D*)，并将其加入到禁忌表中，更新禁忌表,当前解变为  $x_4 = (AEDCB)$ , 目标函数值为  $f(x_4) = 55$ , 历史最优值为53。进行新一次交换。结果如图5所示

# 第六步

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

对目标函数值改善最大的移动是 $(C, B)$ ，当前解变为 $x_5 = (AEDBC)$ ，目标函数值为 $f(x_5) = 53$ ，历史最优值不变，为53。此时，已达到最大迭代次数，算法停止，最优目标值为53，最优解为 $(ACBED)$ 和 $(AEDBC)$ 。

# 中期表和长期表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

前面介绍了基本禁忌搜索算法的禁忌表，即短期表，短期表能避免算法在小范围内的搜索循环，却无法避免较大范围的搜索循环，即循环的操作数超过了禁忌表的长度。因此，又引入了中期表和长期表的概念。应用中期表和长期表，能够实现算法搜索的强化和多样化。以下分别给出介绍。

## (1) 中期表

中期表也称为**频数表或频率表**，可以记录搜索过程中某些操作或者某些适配值出现的频数，对某些出现频率较高的操作或者适配值增加一定的惩罚，以减少其出现的频率，从而使算法更高效。

## (2) 长期表

**长期表**可以记录多个初始解，从每个初始解开始分别进行禁忌搜索，避免了禁忌搜索只采用一个初始解进行搜索局限于一个区域。长期表的引入使得算法能够更好地进行全局搜索。

# 随机化禁忌搜索和候选链表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

通常，禁忌搜索要对领域结构中的每个元素的目标函数进行比较。而这会消耗大量的计算时间。

# 随机化禁忌搜索和候选链表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

通常，禁忌搜索要对领域结构中的每个元素的目标函数进行比较。而这会消耗大量的计算时间。

为解决这一问题，可以采用在邻域结构中**随机抽样**。这样能显著减少计算负担。另外，随机采样有时也能避免循环。此时，可以采用长度较短的禁忌表。当然，随机采样可能会错过好解。

# 随机化禁忌搜索和候选链表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

通常，禁忌搜索要对领域结构中的每个元素的目标函数进行比较。而这会消耗大量的计算时间。

为解决这一问题，可以采用在邻域结构中**随机抽样**。这样能显著减少计算负担。另外，随机采样有时也能避免循环。此时，可以采用长度较短的禁忌表。当然，随机采样可能会错过好解。

另外，也可以随机化应用禁忌准则，即以一定概率考虑禁忌表。

# 随机化禁忌搜索和候选链表

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

通常，禁忌搜索要对领域结构中的每个元素的目标函数进行比较。而这会消耗大量的计算时间。

为解决这一问题，可以采用在邻域结构中**随机抽样**。这样能显著减少计算负担。另外，随机采样有时也能避免循环。此时，可以采用长度较短的禁忌表。当然，随机采样可能会错过好解。

另外，也可以随机化应用禁忌准则，即以一定概率考虑禁忌表。

为了节省计算时间，候选链表也是广泛采用的方法。它以一定策略来从邻域结构子集从选取操作。但是，要注意到候选链表的定义关系到搜索的好坏。

# 强化机制

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在日常生活中，会有这样的体验：**在看上去有前途的区域进行反复搜索，有可能找到更优的解。**

这就是所谓的强化机制的思想来源。在搜索过程中，往往要采用强化搜索。中期记忆用于记录在不同解元素连续出现在当前解中的代数。一种强化策略是从当前最优解出发，并固定一些好的解元素。另一种策略是改变邻域结构以选择更强大（但更复杂）或更多样的操作。强化机制在禁忌搜索并不是必需的，尤其是在正常的搜索能得到好解的情况下，没有必要花费更多的计算代价以在某个区域进行强化搜索。

# 多样化机制

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计局部搜索算法时，多样化机制具有至关重要的重要。在所有基于局部搜索的算法中，保持搜索的多样性对于算法性能有重要影响。

虽然禁忌搜索采用禁忌这一手段，但是它仍然会局限于一个邻域。邻域搜索有可能得到好解，不过，它往往不能探索更加有前途的搜索区域，导致所得到的解远逊于最优解。

多样化机制就是通过使算法能搜索那些搜索空间中未探索区域而采取的（强制）措施。长期记忆在禁忌搜索中起到保持多样性的目的。长期记忆（如频率记忆）记录一个不同解元素出现的次数或操作被选的次数。如果能确定搜索空间中好的区域，频率记忆用于在不同区域搜索的迭代代数。

# 多样化机制

计算智能——禁忌搜索

柯良军

目录

基本思想

算法要素

算法步骤及流程图

例子

禁忌搜索的改进

在设计局部搜索算法时，多样化机制具有至关重要的重要。在所有基于局部搜索的算法中，保持搜索的多样性对于算法性能有重要影响。

虽然禁忌搜索采用禁忌这一手段，但是它仍然会局限于一个邻域。邻域搜索有可能得到好解，不过，它往往不能探索更加有前途的搜索区域，导致所得到的解远逊于最优解。

多样化机制就是通过使算法能搜索那些搜索空间中未探索区域而采取的（强制）措施。长期记忆在禁忌搜索中起到保持多样性的目的。长期记忆（如频率记忆）记录一个不同解元素出现的次数或操作被选的次数。如果能确定搜索空间中好的区域，频率记忆用于在不同区域搜索的迭代代数。

主要有两类多样化方法。一类是重启式多样化。它在当前解中加入一些很少访问的解元素并且重新初始化搜索过程。另一种是连续性多样化，它在搜索过程中直接融入多样化。通过在目标函数值中加入解元素频率相关项使得搜索偏好于一些操作。

# 允许得到不可行解

计算智  
能——禁忌搜  
索

柯良军

目录

基本思想

算法要素

算法步骤及流  
程图

例子

禁忌搜索的改  
进

在求解有约束的优化问题时，由于约束的影响，如果只在可行空间中搜索，将会过分限制搜索过程。

约束松弛通过定义更大的搜索空间而使得在邻域结构中探索变得容易。它通过在定义搜索空间时不考虑一些约束，而在目标函数值中加入罚项实现。

当然，带来的问题是如何选取罚项的权重。权重自校准技术是解决这类问题有效方法，它通过在搜索过程中依据搜索历史来动态调整权重，具体地，如果在过去若干代中都只能找到不可行解，则增加权重，否则减小权重。通过修正罚项能使搜索通过可行区域边界，有利于搜索的多样性。这也称为策略振荡。