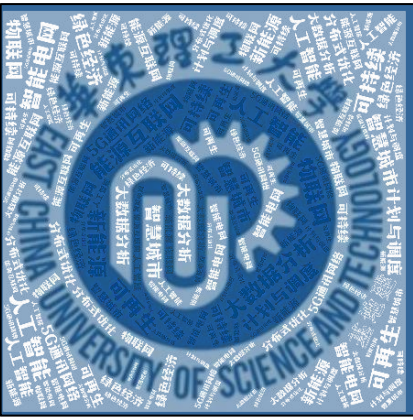


人工智能原理与应用

Artificial intelligence theory and application

华东理工大学

钟伟民 彭鑫 姜庆超 宋冰

课时内容

第5章 智能搜索技术

5 智能搜索技术

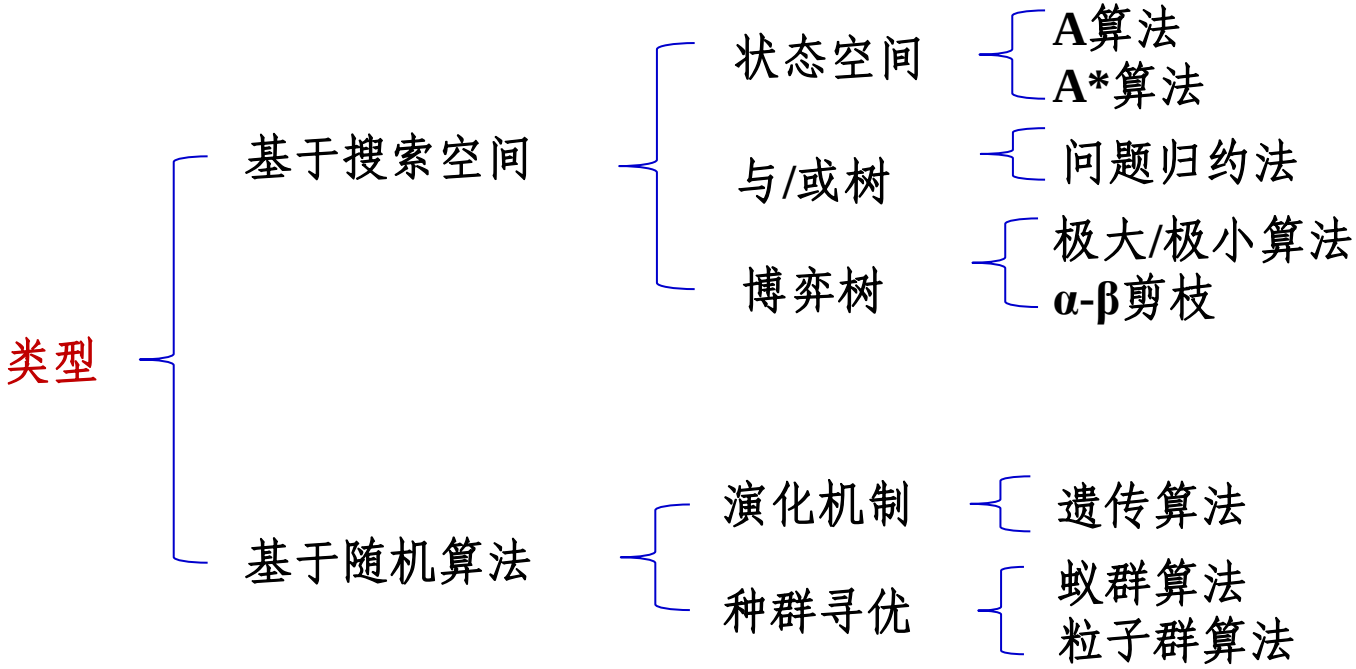
- **搜索概述**
  - a. **搜索的含义**
  - b. 状态空间问题求解方法
  - c. 问题规约求解方法
  - d. 进化搜索法概述
- **状态空间的启发式搜索**
- **与/或树的启发式搜索**
- **博弈树的启发式搜索**
- **进化搜索**

5 搜索的含义

◆ 搜索的含义

搜索：依靠经验，利用已有知识，根据问题的实际情况，不断寻找可利用知识，从而构造一条代价最小的推理路线，使问题得以解决的过程称为搜索。

智能搜索：是指可以利用搜索过程得到的中间信息来引导搜索项往最优方向发展的算法。



5 智能搜索技术

- **搜索概述**
  - a. 搜索的含义
  - b. 状态空间问题求解方法
  - c. 问题规约求解方法
  - d. 进化搜索法概述
- **状态空间的启发式搜索**
- **与/或树的启发式搜索**
- **博弈树的启发式搜索**
- **进化搜索**

5 状态空间的问题求解方法

◆ 状态空间问题表示

状态(State)

是表示问题求解过程中每一步问题状况的数据结构，它可形式地表示为：

$$S_k=\{S_{k0}, S_{k1}, \cdots\}$$

当对每一个分量都给以确定的值时，就得到了一个具体的状态。

操作(Operator)

也称为算符，它是把问题从一种状态变换为另一种状态的手段。操作可以是一个机械步骤，一个运算，一条规则或一个过程。操作可理解为状态集合上的一个函数，它描述了状态之间的关系。

5 状态空间的问题求解方法

◆ 状态空间问题表示

状态空间(State space)

用来描述一个问题的全部状态以及这些状态之间的相互关系。常用一个三元组表示为：

$$(S, F, G)$$

其中，S为问题的所有初始状态集合；F为操作的集合；G为目标状态的集合。

状态空间也可用一个赋值的有向图来表示，该有向图称为**状态空间图**。在状态空间图中，节点表示问题的状态，有向边表示操作。

5 状态空间的问题求解方法

◆ 状态空间问题求解

状态空间法求解问题的基本过程：

首先，为问题选择适当的“状态”及“操作”的形式化描述方法；

然后，从某个初始状态出发，每次使用一个“操作”，递增地建立起操作序列，直到达到目标状态为止；

最后，由初始状态到目标状态所使用的算符序列就是该问题的一个解。

5 智能搜索技术

- **搜索概述**
  - a. 搜索的含义
  - b. 状态空间问题求解方法
  - c. 问题规约求解方法
  - d. 进化搜索法概述
- **状态空间的启发式搜索**
- **与/或树的启发式搜索**
- **博弈树的启发式搜索**
- **进化搜索**

5 问题规约求解方法

◆ 问题的分解与等价变换

基本思想

当一问题较复杂时，可通过分解或变换，将其转化为一系列较简单的子问题，然后通过对这些子问题的求解来实现对原问题的求解。

分解

如果一个问题P可以归约为一组子问题$P_1,P_2,\ldots,P_n$，并且只有当所有子问题P_i都有解时原问题P才有解，任何一个子问题P_i无解都会导致原问题P无解，则称此种归约为问题的分解。

即分解所得到的子问题的“与”与原问题P等价。

等价变换

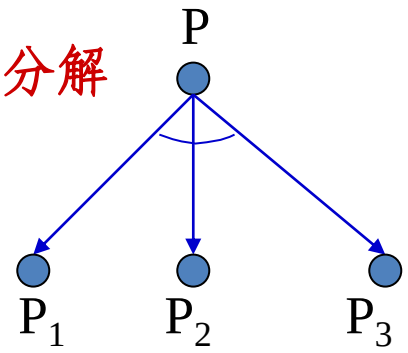
如果一个问题P可以归约为一组子问题$P_1,P_2,\ldots,P_n$，并且子问题P_i中只要有一个有解则原问题P就有解，只有当所有子问题P_i都无解时原问题P才无解，称此种归约为问题的等价变换，简称变换。

即变换所得到的子问题的“或”与原问题P等价。

5 问题规约求解方法

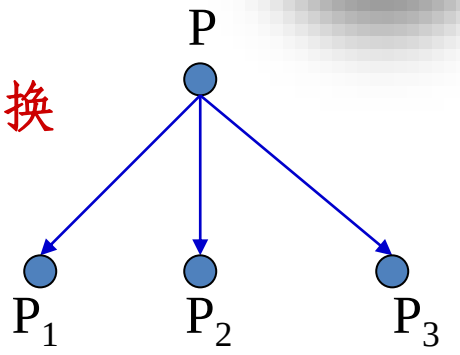
◆ 问题规约的与/或树表示

(1)与树　分解



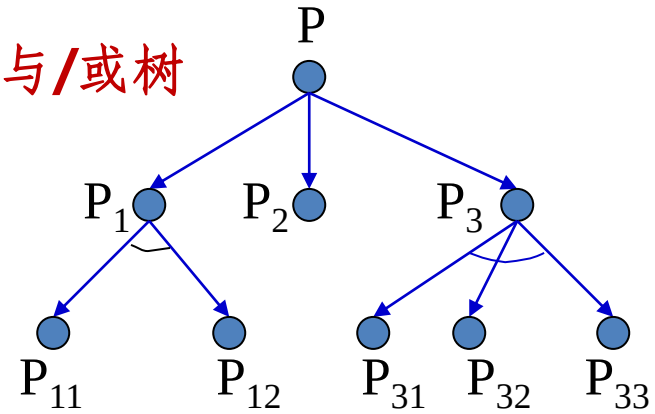
与树

(2) 或树　等价变换



或树

(3) 与/或树



与/或树

5 问题规约求解方法

◆ 问题规约的与/或树表示

本原问题是指那种不能（或不需要）再进行分解或变换且可以直接解答的问题。

(1) 端节点与终止节点

在与/或树中，没有子节点的节点称为端节点；本原问题所对应的节点称为终止节点。可见，终止节点一定是端节点，但端节点却不一定是终止节点。

(2) 可解节点与不可解节点

在与/或树中，满足以下三个条件之一的节点为可解节点：

①任何终止节点都是可解节点。

②对“或”节点，当其子节点中至少有一个为可解节点时，则该或节点就是可解节点。

③对“与”节点，只有当其子节点全部为可解节点时，该与节点才是可解节点。

同样，可用类似的方法定义不可解节点：

①不为终止节点的端节点是不可解节点。

②对“或”节点，若其全部子节点都为不可解节点，则该或节点是不可解节点。

③对“与”节点，只要其子节点中有一个为不可解节点，则该与节点是不可解节点。

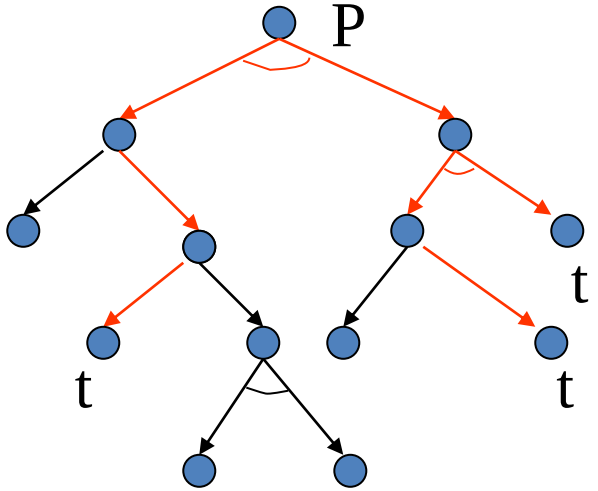
◆ 问题规约的与/或树表示

(3) 解树

由可解节点构成，并且由这些可解节点可以推出初始节点（它对应着原始问题）为可解节点的子树为解树。在解树中一定包含初始节点。

例如，右图给出的与或树中，用红线表示的子树是一个解树。在该图中，节点P为原始问题节点，用t标出的节点是终止节点。根据可解节点的定义，很容易推出原始问题P为可解节点。

问题归约求解过程就实际上就是生成解树，即证明原始节点是可解节点的过程。这一过程涉及到搜索的问题，对于与/或树的搜索将在后面详细讨论。



5 智能搜索技术

- **搜索概述**
  - a. 搜索的含义
  - b. 状态空间问题求解方法
  - c. 问题规约求解方法
  - d. 进化搜索法概述
- **状态空间的启发式搜索**
- **与/或树的启发式搜索**
- **博弈树的启发式搜索**
- **进化搜索**

5 进化搜索法概述

◆ 进化搜索的概念及其生物学基础

(1) 什么是进化搜索

进化搜索算法也称为模拟进化优化算法或进化计算（Evolutionary Computation，EC），是在达尔文进化论和孟德尔遗传变异理论的基础上产生的一种在基因和种群层次上模拟自然界生物进化过程与机制进行问题求解的自组织、自适应的随机搜索技术。主要包括

遗传算法（Genetic Algorithm，GA）

进化策略（Evoluenettionary Strategy, ES）

进化规划（Evolutionary Programming, EP）

遗传规划（Genetic Programming, GP）四大分支。

5 进化搜索法概述

◆ 进化搜索的概念及其生物学基础

(1) 什么是进化搜索

进化搜索以达尔文进化论的“物竞天择、适者生存”作为算法的进化规则，并结合孟德尔的遗传变异理论，将生物进化过程中的

繁殖（Reproduction）

变异（Mutation）

竞争（Competition）

选择（Selection）

引入到了算法中。

5 进化搜索法概述

◆ 进化搜索的概念及其生物学基础

(2) 进化计算的生物学基础

自然界生物进化过程是进化计算的生物学基础，它主要包括遗传(Heredity)、变异(Mutation)和进化(Evolution)理论。

① 遗传理论

遗传是指父代（或亲代）利用遗传基因将自身的基因信息传递给下一代（或子代），使子代能够继承其父代的特征或性状的这种生命现象。正是由于遗传的作用，自然界才能有稳定的物种。

在自然界，构成生物基本结构与功能的单位是**细胞**（Cell）。

细胞中含有一种包含着所有遗传信息的复杂而又微小的丝状化合物，人们称其为**染色体**（Chromosome）。

5 进化搜索法概述

◆ 进化搜索的概念及其生物学基础

(2) 进化计算的生物学基础

① 遗传理论

在染色体中，遗传信息由**基因**（Gene）所组成，基因决定着生物的性状，是遗传的基本单位。

染色体的形状是一种双螺旋结构，构成染色体的主要物质叫做**脱氧核糖核酸(DNA)**，每个基因都在DNA长链中占有一定的位置。

一个细胞中的所有染色体所携带的遗传信息的全体称为一个**基因组**(Genome)。

细胞在分裂过程中，其遗传物质DNA通过复制转移到新生细胞中，从而实现了生物的遗传功能。

◆ 进化搜索的概念及其生物学基础

(2) 进化计算的生物学基础

② 变异理论

变异是指子代和父代之间，以及子代的各个不同个体之间产生差异的现象。变异是一种随机、不可逆现象，是生物多样性的基础。

引起变异的主要原因：

杂交，是指有性生殖生物在繁殖下一代时两个同源染色体之间的交配重组，即两个染色体在某一相同处的DNA被切断后再进行交配重组，形成两个新的染色体。

复制差错，是指在细胞复制过程中因DNA上某些基因结构的随机改变而产生出新的染色体。

◆ 进化搜索的概念及其生物学基础

(2) 进化计算的生物学基础

③ 进化论

进化是指在生物延续生存过程中，逐渐适应其生存环境，使得其品质不断得到改良的这种生命现象。遗传和变异是生物进化的两种基本现象，优胜劣汰、适者生存是生物进化的基本规律。

达尔文的自然选择学说：在生物进化中，一种基因有可能发生变异而产生出另一种新的基因。这种新基因将依据其与生存环境的适应性而决定其增殖能力。通常，适应性强的基因会不断增多，而适应性差的基因则会逐渐减少。通过这种自然选择，物种将逐渐向适应于生存环境的方向进化，甚至会演变成为另一个新的物种，而那些不适应于环境的物种将会逐渐被淘汰。

5 进化搜索法概述

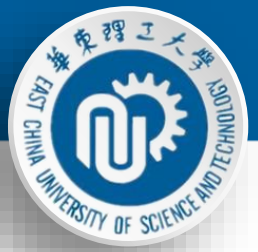
◆进化搜索的基本过程

进化计算尽管有多个重要分支，并且不同分支的编码方案、选择策略和进化操作也有可能不同，但它们却有着共同的进化框架。若假设P为种群(Population，或称为群体)，t为进化代数， P(t)为第t代种群 ，则进化计算的基本结构可粗略描述如下：

```
{ 确定编码形式并生成搜索空间;
  初始化各个进化参数，并设置进化代数t=0;
  初始化种群P(0);
  对初始种群进行评价（即适应度计算）；
  while（不满足终止条件）do
     {
     t=t+1;
     利用选择操作从P(t-1)代中选出P(t)代群体;
     对P(t)代种群执行进化操作;
     对执行完进化操作后的种群进行评价（即适应度计算）；
     }}
```

可以看出，上述基本结构包含了生物进化中所必需的选择操作、进化操作和适应度评价等过程。

5 智能搜索技术



- 搜索概述
- 状态空间的启发式搜索
  a. 启发性信息与估价函数
  b. A算法
  c. A*算法
  d. A*算法应用举例
- 与/或树的启发式搜索
- 博弈树的启发式搜索
- 进化搜索

5 启发性信息与估价函数

◆ 概念

启发性信息

启发性信息是指那种与具体问题求解过程有关的，并可指导搜索过程朝着最有希望方向前进的控制信息。

启发信息的启发能力越强，扩展的无用结点越少。包括以下3种：

① 有效地帮助确定扩展节点的信息；

② 有效的帮助决定哪些后继节点应被生成的信息；

③ 能决定在扩展一个节点时哪些节点应从搜索树上删除的信息。

估价函数

用来估计节点重要性，定义为从初始节点S_0出发，约束经过节点n到达目标节点S_g的所有路径中最小路径代价的估计值。一般形式：

$$f(n)=g(n)+h(n)$$

其中，g(n)是从初始节点S_0到节点n的实际代价；h(n)是从节点n到目标节点S_g的最优路径的估计代价。

5 启发性信息与估价函数

例5.4 八数码难题。设问题的初始状态S_0和目标状态Sg如下图所示，且估价函数为

$$f(n)=d(n)+W(n)$$

其中：d(n)表示节点n在搜索树中的深度

W(n)表示节点n中“不在位”的数码个数。

请计算初始状态S_0的估价函数值$f(S_0)$

S_0

2	8	3
1		4
7	6	5

S_g

1	2	3
8		4
7	6	5

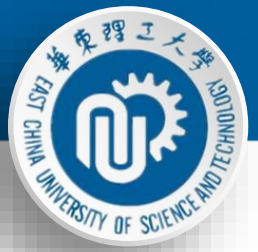
解：即g(n)=d(n)，h(n)=W(n)。 d(n)说明用从S_0到n的路径上的单位代价表示实际代价； W(n)说明用结点n中“不在位”的数码个数作为启发信息。

可见，某节点中的“不在位”的数码个数越多，说明它离目标节点越远。

对初始节点S_0，由于$d(S_0)=0$，$W(S_0)=3$，因此有

$$f(S_0)=0+3=3$$

5 智能搜索技术



- 搜索概述
- 状态空间的启发式搜索
  a. 启发性信息与估价函数
  b. A算法（只考A，不考A*）
  c. A*算法
  d. A*算法应用举例
- 与/或树的启发式搜索
- 博弈树的启发式搜索
- 进化搜索

5 A算法

◆ 概念和算法描述

在状态空间搜索中，**如果每一步都利用估价函数f(n)=g(n)+h(n)对Open表中的节点进行排序，则称A算法**。它是一种为启发式搜索算法。

类型：

全局择优：从Open表的所有节点中选择一个估价函数值最小的进行扩展。

局部择优：仅从刚生成的子节点中选择一个估价函数值最小的进行扩展。

全局择优搜索A算法描述：

(1)把初始节点S_0放入Open表中，$f(S_0)=g(S_0)+h(S_0)$；

(2)如果Open表为空，则问题无解，失败退出；

(3)把Open表的第一个节点取出放入Closed表，并记该节点为n；

(4)考察节点n是否为目标节点。若是，则找到了问题的解，成功退出；

(5)若节点n不可扩展，则转第(2)步；

(6)扩展节点n，生成其子节点n_i(i=1，2，…)，计算每一个子节点的估价值$f(n_i)$(i=1, 2, …)，并为每一个子节点设置指向父节点的指针，然后将这些子节点放入Open表中；

(7)根据各节点的估价函数值，对Open表中的全部节点按从小到大的顺序重新进行排序；

(8)转第(2)步。

5 A算法

例5.5 八数码难题。设问题的初始状态S_0和目标状态S_g如图所示，估价函数与例5.4相同。请用全局择优搜索解决该问题。

解：该问题的全局择优搜索树如下图所示。在该图中，每个节点旁边的数字是该节点的估价函数值。

例如，对节点S_2，其估价函数值的计算为：$f(S_2)=d(S_2)+W(S_2)=2+2=4$

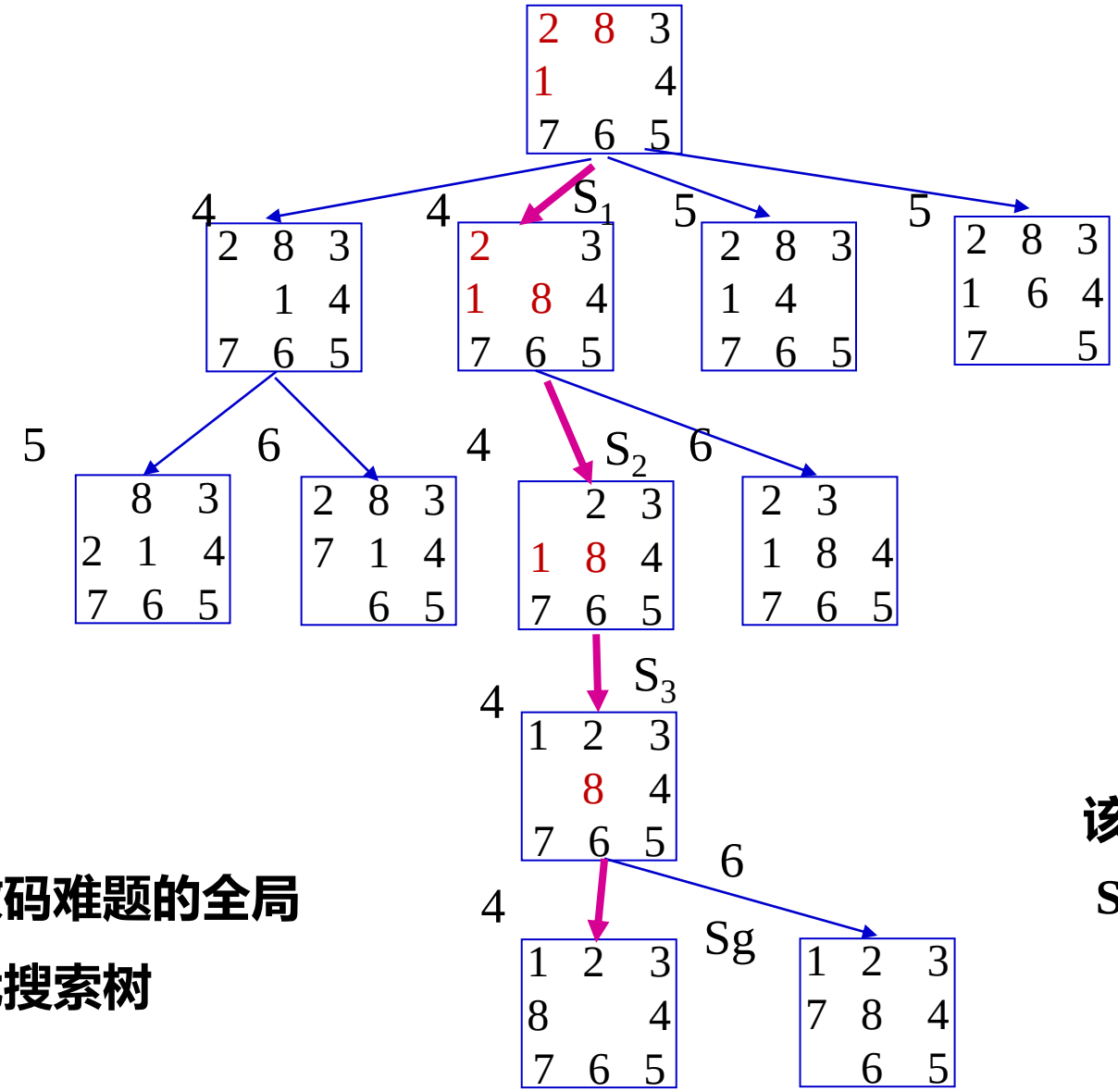
S_0

2	8	3
1		4
7	6	5

Sg

1	2	3
8		4
7	6	5

5 A算法



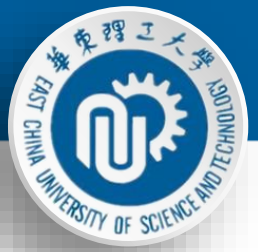
八数码难题的全局择优搜索树

$f(n)=d(n)+W(n)$

该问题的解为：

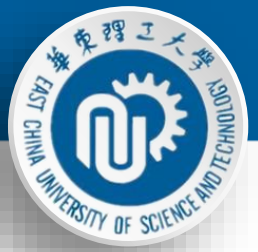
$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_g$

5 智能搜索技术



- 搜索概述
- 状态空间的启发式搜索
- 与/或树的启发式搜索
  a. 解树的代价和希望树
  b. 与/或树的启发式搜索过程
- 博弈树的启发式搜索
- 进化搜索

5 与/或树的启发式搜索



与/或树的启发式搜索过程实际上是一种利用搜索过程所得到的启发性信息寻找**最优解树**的过程。

算法的每一步都试图找到一个最有希望成为**最优解树的子树**。

最优解树是指代价最小的那棵解树。它涉及到**解树的代价**与**希望树**。

5 解树的代价与希望树

◆ 解树的代价

解树的代价可按如下方法计算：

(1)若n为终止节点，则其代价h(n)=0；

(2)若n为**或节点**，且子节点为$n_1, n_2, \ldots, n_k$，则n的代价为：

$$h(n)=\min_{1\le i\le k}\{c(n,n_i)+h(n_i)\}$$

其中，$c(n, n_i)$是节点n到其子节点n_i的边代价。

(3)若n为**与节点**，且子节点为$n_1, n_2, \ldots, n_k$，则n的代价可用和代价法或最大代价法。

若用**和代价法**，则其计算公式为：

$$h(n)=\sum_{i=1}^{k}\{c(n,n_i)+h(n_i)\}$$

若用**最大代价法**，则其计算公式为：

$$h(n)=\max_{1\le i\le k}\{c(n,n_i)+h(n_i)\}$$

(4)若n是**端节点**，但又不是终止节点，则n不可扩展，其代价定义为$h(n)=\infty$。

(5)根节点的代价即为解树的代价。

5 解树的代价与希望树

◆ 解树的代价

例5.8 设下图是一棵与/或树，它包括两棵解树，左边的解树由S_0、A、t_1、C及t_2组成；右边的解树由S_0、B、t_2、D及t_4组成。在该树中，t_1、t_2、t_3、t_4为终止节点；E、F是端节点；边上的数字是该边的代价。请计算解树的代价。

解：先计算左边的解树

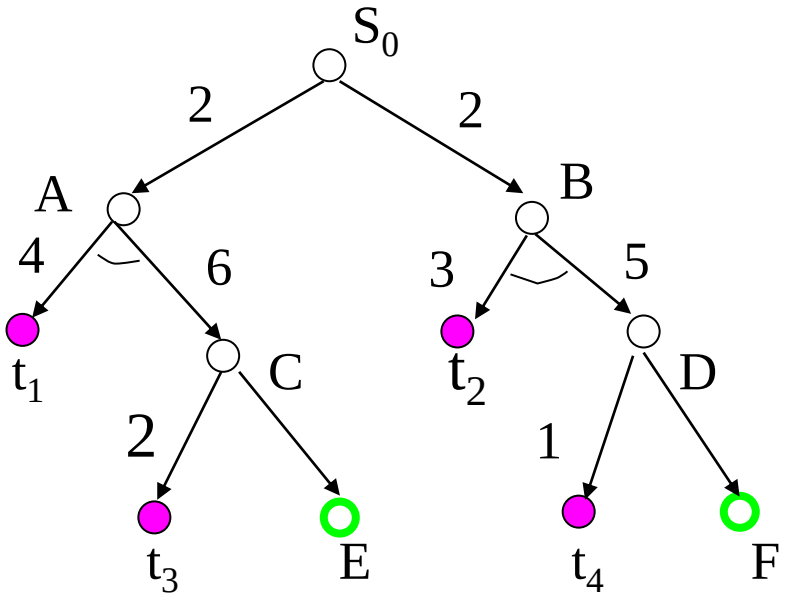
按和代价： $h(S_0)=2+4+6+2=14$

按最大代价： $h(S_0)=(2+6)+2=10$

再计算右边的解树

按和代价： $h(S_0)=1+5+3+2=11$

按最大代价： $h(S_0)=(1+5)+2=8$



与/或树的代价

5 智能搜索技术

- 搜索概述
- 状态空间的启发式搜索
- 与/或树的启发式搜索
- 博弈树的启发式搜索
  a. 博弈概述
  b. 极/大极小过程
  c. α-β剪枝
- 进化搜索

5 博弈概述

◆ 博弈

博弈的概念

博弈是一类具有智能行为的竞争活动，如下棋、战争等。

博弈的类型

双人完备信息博弈：两位选手（例如MAX和MIN ）对垒，轮流走步，每一方不仅知道对方已经走过的棋步，而且还能估计出对方未来的走步。

机遇性博弈：存在不可预测性的博弈，例如掷币等。

博弈树

若把双人完备信息博弈过程用图表示出来，就得到一棵与/或树，这种与/或树被称为博弈树。在博弈树中，那些下一步该MAX走步的节点称为MAX节点，下一步该MIN走步的节点称为MIN 节点。

5 博弈概述

◆ 博弈的特点

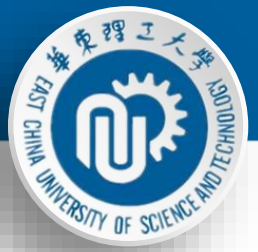
博弈树的特点

(1) 博弈的初始状态是初始节点；

(2) 博弈树中的“或”节点和“与”节点是逐层交替出现的；

(3) 整个博弈过程始终站在某一方的立场上，例如MAX方。所有能使自己一方获胜的终局都是本原问题，相应的节点是可解节点；所有使对方获胜的终局都是不可解节点。

5 智能搜索技术



- 搜索概述
- 状态空间的启发式搜索
- 与/或树的启发式搜索
- 博弈树的启发式搜索
  a. 博弈概述
  b. 极大极小过程
  c. α-β剪枝
- 进化搜索

5 极大极小过程

◆ 概念

对简单的博弈问题，可生成整个博弈树，找到必胜的策略。

对于复杂的博弈问题，不可能生成整个搜索树，如国际象棋，大约有10^{120}个节点。 一种可行的方法是用当前正在考察的节点生成一棵部分博弈树，并利用估价函数f(n)对叶节点进行静态估值。

对叶节点的估值方法是：那些对MAX有利的节点，其估价函数取正值；那些对MIN有利的节点，其估价函数取负值；那些使双方均等的节点，其估价函数取接近于0的值。

为非叶节点的值，必须从叶节点开始向上倒推。其倒推方法是：

对于MAX节点，由于MAX 方总是选择估值最大的走步，因此，MAX节点的倒推值应该取其后继节点估值的最大值。

对于MIN节点，由于MIN方总是选择使估值最小的走步，因此MIN节点的倒推值应取其后继节点估值的最小值。

这样一步一步的计算倒推值，直至求出初始节点的倒推值为止。这一过程称为极大极小过程。

5 智能搜索技术

- 搜索概述
- 状态空间的启发式搜索
- 与/或树的启发式搜索
- 博弈树的启发式搜索
- 进化搜索
  a. 蚁群算法
  b. 遗传算法
  c. 粒子群优化算法

5 蚁群算法

◆群智能

- 群智能（Swarm Intelligence, SI）

  人们把群居昆虫的集体行为称作“群智能”（“群体智能”、“群集智能”、“集群智能”等）

- 特点

  个体的行为很简单，但当它们一起协同工作时，**却能够突现（涌现）**出非常复杂（智能）的行为特征。

5 蚁群算法

◆群智能

- **描述**

  群智能作为一种新兴的演化计算技术已成为研究焦点，它与人工生命，特别是进化策略以及遗传算法有着极为特殊的关系。

- **特性**

  指无智能的主体通过合作表现出智能行为的特性，在没有集中控制且不提供全局模型的前提下，为寻找复杂的分布式问题求解方案提供了基础。

5 蚁群算法

◆群智能

- 优点

  灵活性：群体可以适应随时变化的环境；

  稳健性：即使个体失败，整个群体仍能完成任务；自我组织：活动既不受中央控制，也不受局部监管。

- 典型算法

  蚁群算法（蚂蚁觅食）

  粒子群优化算法（鸟群捕食）人工蜂群

  算法（蜜蜂采蜜）

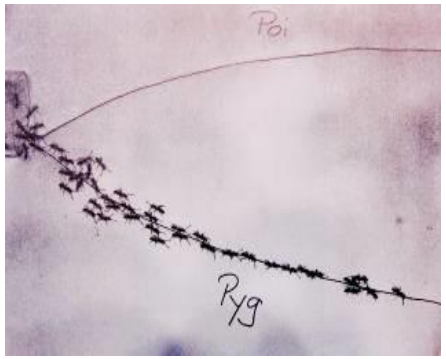
5 蚁群算法

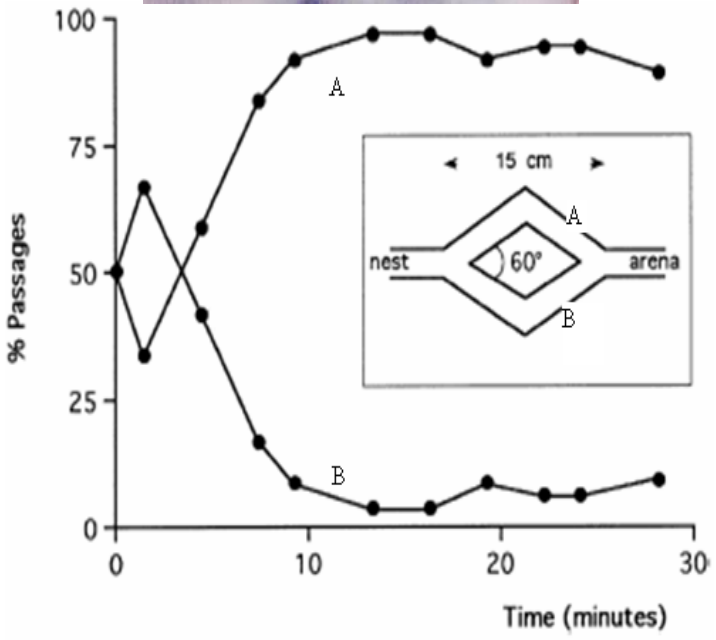
◆ 蚁群优化算法原理

- 蚁群的自组织行为

  “双桥实验”

  通过遗留在来往路径上的信息素（Pheromone）的挥发性化学物质来进行通信和协调。





5 蚁群算法

◆ 蚁群优化算法起源

- **提出蚁群算法**

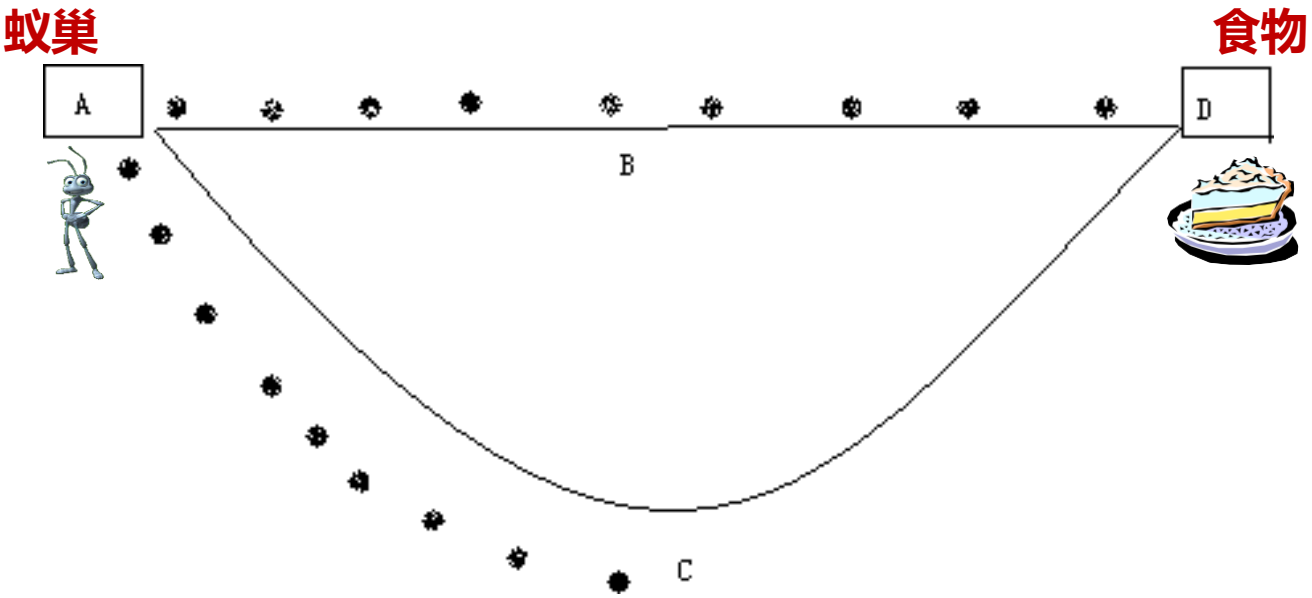
上世纪90年代初，意大利学者M. Dorigo在其博士论文中提出蚂蚁算法（Ant System）。

2000年， M. Dorigo等在《Nature》发表了蚁群算法的综述。

近年来， M. Dorigo等人进一步将蚂蚁算法发展为一种通用的优化技术——蚁群优化（ant colony optimization, ACO）。

5 蚁群算法

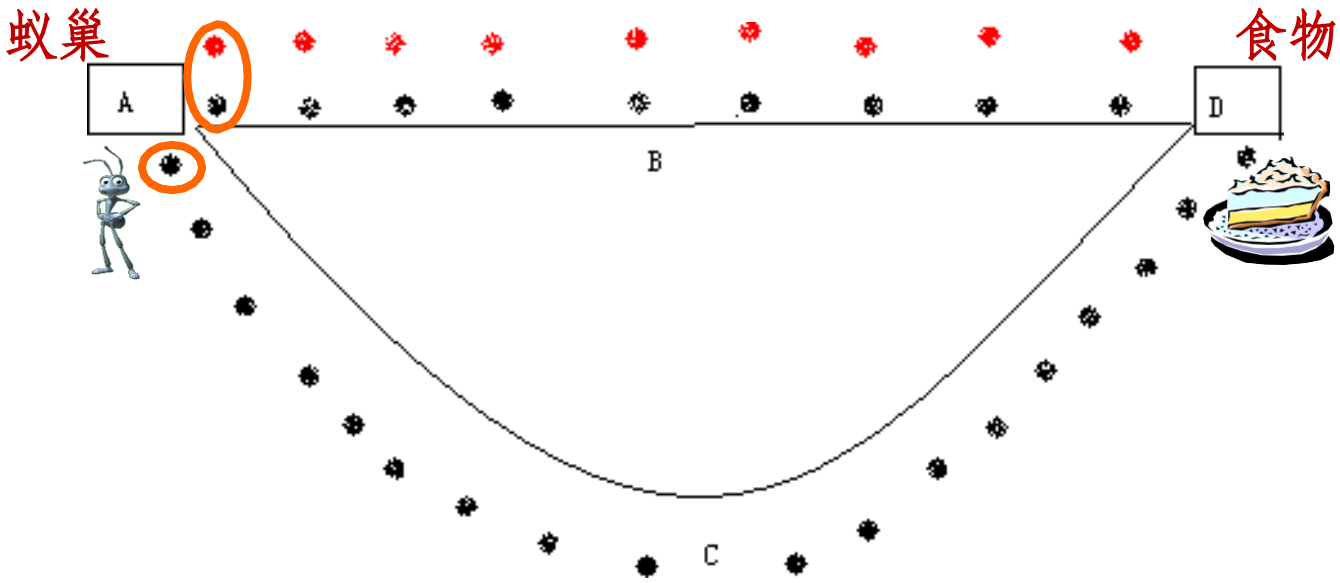
◆ 蚁群优化算法



蚂蚁从A点出发，随机选择路线ABD或ACD。经过9个时间单位时：走ABD的蚂蚁到达终点，走ACD的蚂蚁刚好走到C点。

5 蚁群算法

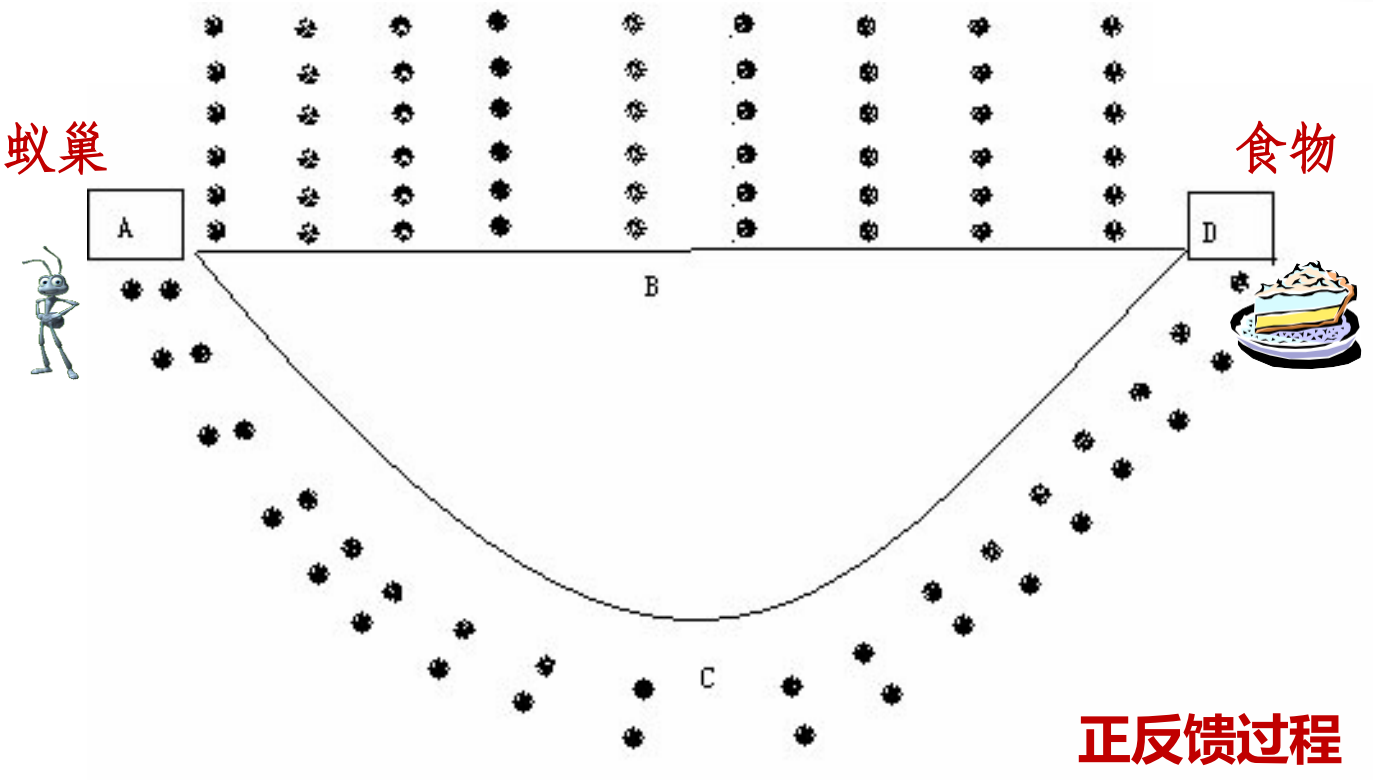
◆ 蚁群优化算法



经过18个时间单位时：走ABD的蚂蚁到达终点后得到食物又返回了起点A，而走ACD的蚂蚁刚好走到D点。

5 蚁群算法

◆ 蚁群优化算法



正反馈过程

最后的极限是所有的蚂蚁只选择ABD路线。

5 蚁群算法

◆ 蚁群优化算法

人工蚂蚁与真实蚂蚁的异同

- **相同点**
  - ➢两个群体中都存在个体相互交流的通信机制
  - ➢都要完成寻找最短路径的任务
  - ➢都采用根据当前信息进行路径选择的随机选择策略
- **不同点**
  - ➢人工蚂蚁**具有记忆能**力
  - ➢人工蚂蚁选择路径并非完全盲目，受问题空间特征的启发，**按一定算法规律有意识地寻找最短路径**
  - ➢人工蚂蚁生活在**离散空间**，而真实蚂蚁生活在连续时间的环境中

5 蚁群算法

◆ 基本蚁群优化算法

- 解决TSP问题

假设蚁群算法中的每只蚂蚁都是具有下列特征的简单智能体

1、每次周游，每只蚂蚁在其经过的支路（i, j）上都留下信息素

2、蚂蚁选择城市的概率与城市之间的距离和当前连接支路上所包含的信息素余量有关

3、为了加强蚂蚁进行合法的周游，直到一次周游完成后，才允许蚂蚁游走已访问的城市　　（可由禁忌表来控制）

蚁群算法的介绍，离不开TSP问题

5 蚁群算法

◆ 基本蚁群优化算法

- 解决TSP问题

TSP问题与基本变量和参数介绍

在算法的初始时刻，将m只蚂蚁随机放到n座城市;

将每只蚂蚁 k的禁忌表tabuk(s)的第一个元素tabuk(1)设置为它当前所在城市;

设各路径上的信息素$\tau_{ij}(0)=C$（C为一较小的常数）；

5 蚁群算法

◆ 基本蚁群优化算法

- **解决TSP问题**

  每只蚂蚁根据路径上的信息素和启发式信息（两城市间距离）独立地选择下一座城市：

  在时刻t，蚂蚁k从城市i 转移到城市j 的概率为

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s\in J_k(i)}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & j\in J_k(i)\\ 0, & j\notin J_k(i)\end{cases}$$

下一步允许的城市的集合

α、β分别表示信息素和启发式因子的相对重要程度。

$$J_k(i)=\{1,2,\cdots,n\}-tabu_k,$$

表示从 i 转移到 j 的期望程度，先验知识

5 蚁群算法

◆ 基本蚁群优化算法

- 解决TSP问题

当所有蚂蚁完成一次周游后，各路径上的信息素将进行更新：

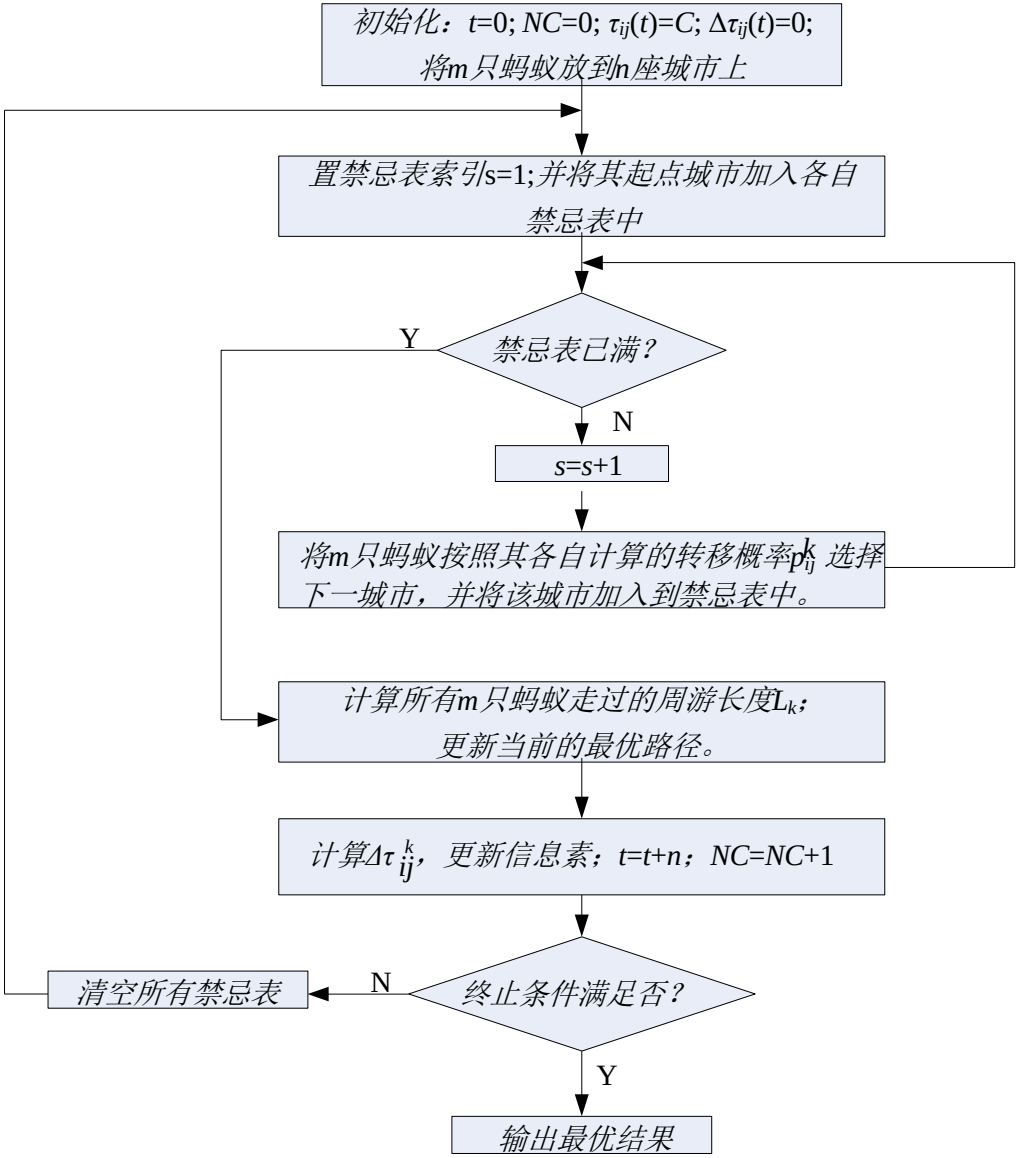
$$\tau_{ij}(t+n)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k},\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{若蚂蚁k在本次周游中经过边ij}\\ 0, & \text{否则}\end{cases}$$

其中，ρ（0≦ ρ <1）表示路径上信息素的蒸发系数；Q为正常数，表示蚂蚁循环一周或一个过程在经过的路径上所释放的信息素总量，影响算法的收敛速度；Lk表示第k只蚂蚁在本次周游中所走过路径的长度。

5 蚁群算法

◆ 算法流程


初始化：t=0; NC=0; τij(t)=C; Δτij(t)=0;
将m只蚂蚁放到n座城市上
置禁忌表索引s=1;并将其起点城市加入各自禁忌表中
禁忌表已满？
Y
N
s=s+1
将m只蚂蚁按照其各自计算的转移概率pijk 选择下一城市，并将该城市加入到禁忌表中。
计算所有m只蚂蚁走过的周游长度Lk；
更新当前的最优路径。
计算Δτijk，更新信息素；t=t+n；NC=NC+1
终止条件满足否？
N
清空所有禁忌表
Y
输出最优结果

5 蚁群算法

◆ 基本蚁群优化算法

- **初始参数**

  城市数30;

  蚂蚁数30;

  α=1;

  β=5;

  ρ=0.5;

  最大迭代代数200;

  Q=100;

蚂蚁转移概率

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s\in J_k(i)}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & j\in J_k(i)\\ 0, & j\notin J_k(i)\end{cases}$$

$$J_k(i)=\{1,2,\cdots,n\}-tabu_k,\quad \eta_{ij}=1/d_{ij}$$

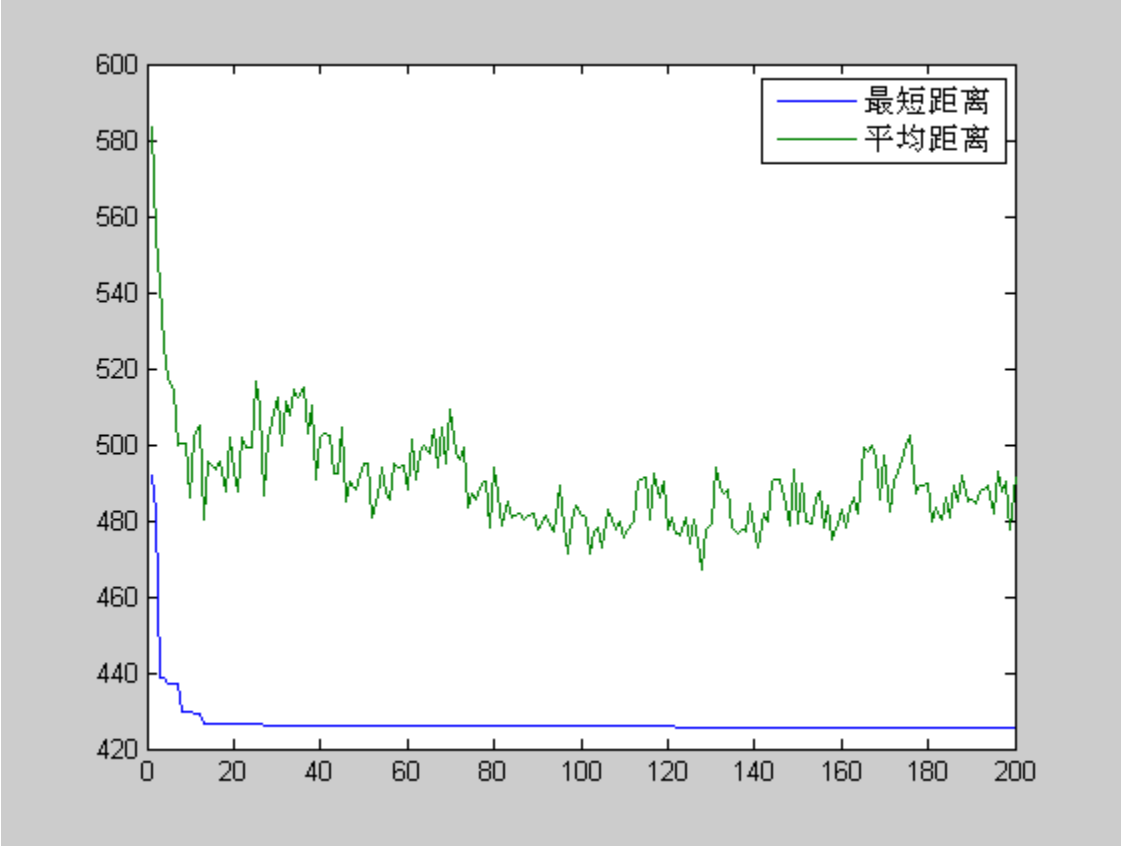
信息素

$$\tau_{ij}(t+n)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k},\ \Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{若蚂蚁k在本次周游中经过边ij}\\ 0, & \text{否则}\end{cases}$$

5 蚁群算法

◆ 基本蚁群优化算法



5 蚁群算法

◆ 基本蚁群优化算法

- **三种模型（信息素）**

ant-cycle：
（蚁周）

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{蚂蚁k在本次周游中经过ij}\\ 0, & \text{否则}\end{cases}$$

全局信息

ant-quantity：
（蚁量）

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{d},, & \text{蚂蚁k在时刻t和t+1经过ij}\\ 0, & \text{否则}\end{cases}$$

ant-density:
（蚁密）

$$\Delta\tau_{ij}^{k}=\begin{cases}Q, & \text{蚂蚁k在时刻t和t+1经过ij}\\ 0, & \text{否则}\end{cases}$$

局部信息

5 蚁群算法

◆ 基本蚁群优化算法

- 三种模型

在Ant-density和Ant-quantity中蚂蚁在两个位置节点间每移动一次后即更新信息素（局部信息），而在Ant-cycle中当所有的蚂蚁都完成了自己的行程后（全局信息）才对信息素进行更新。

◆ 基本蚁群优化算法

- **三种模型的比较**

模型ant-density, ant-quantity, ant-cycle的比较(M. Dorigo,1996)

模型	参数集		
	最好参数集	平均结果	最好结果
ant-density	α = 1,β=5,ρ=0.01	426.740	424.635
ant-quantity	α = 1,β=5,ρ=0.01	427.315	426.255
ant-cycle	α = 1,β=5,ρ=0.5	424.250	423.741

5 蚁群算法

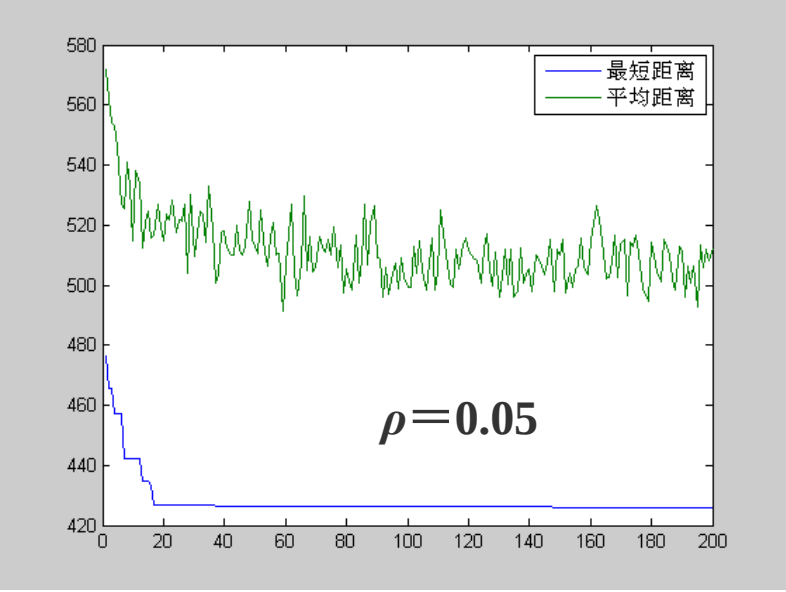
◆ 蚁群优化算法的参数设置和基本属性

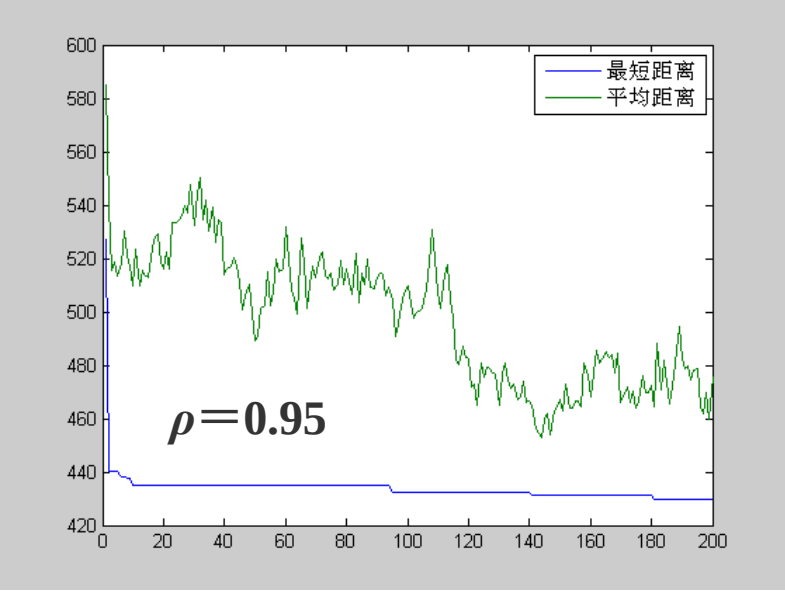
- **信息素的分布**

$$\tau_{ij}(t+n)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}$$

蒸发系数的影响：

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k},\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{若蚂蚁k在本次周游中经过边ij}\\ 0, & \text{否则}\end{cases}$$





5 蚁群算法

◆ 蚁群优化算法的参数设置和基本属性

- 参数α、 β对算法性能的影响

停滞现象（Stagnation behavior）（α）：所有蚂蚁都选择相同的路径，即系统不再搜索更好的解。原因在于较好路径上的信息素远大于其他边上的，从而使所有蚂蚁都选择该路径。

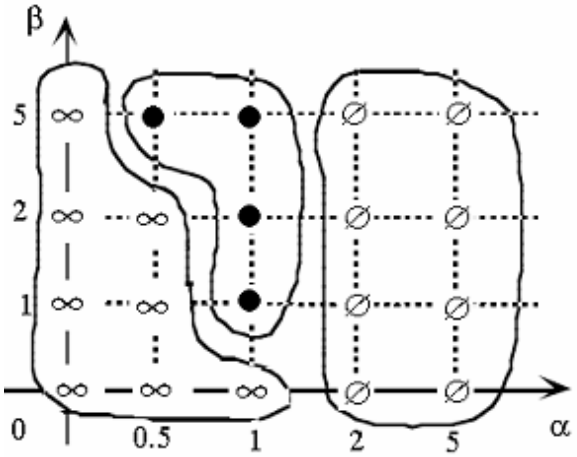
5 蚁群算法

◆ 蚁群优化算法的参数设置和基本属性

- 参数α、 β对算法性能的影响

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s\in J_k(i)}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & j\in J_k(i)\\ 0, & j\notin J_k(i)\end{cases}$$

α取较大值时，意味着在选择路 径 时，路径上的信息素非常重要;
当α取较小值时，变成随机的贪婪算法。



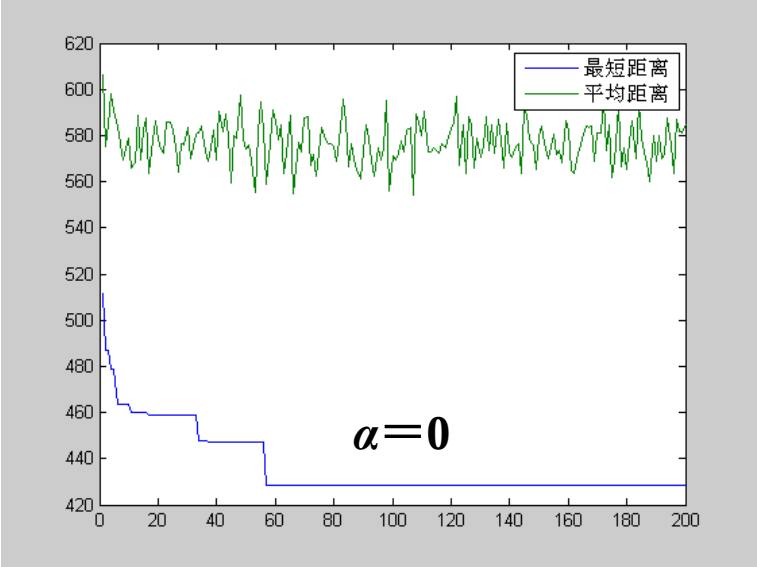
Ant-cycle behavior for different combinations of α-β parameters.

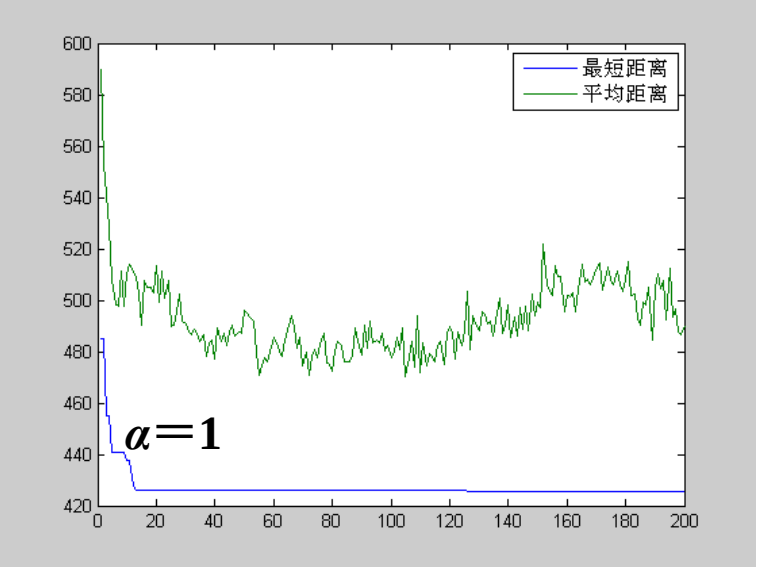
● - The algorithm finds the best known solution without entering the stagnation behavior.
∞ - The algorithm doesn't find good solutions without entering the stagnation behavior.
∅ - The algorithm doesn't find good solutions and enters the stagnation behavior.

5 蚁群算法

◆ 蚁群优化算法的参数设置和基本属性

- 参数α、 β对算法性能的影响
- $\alpha = 0$，蚂蚁之间无协同作用；$\alpha = 1$，有协同作用

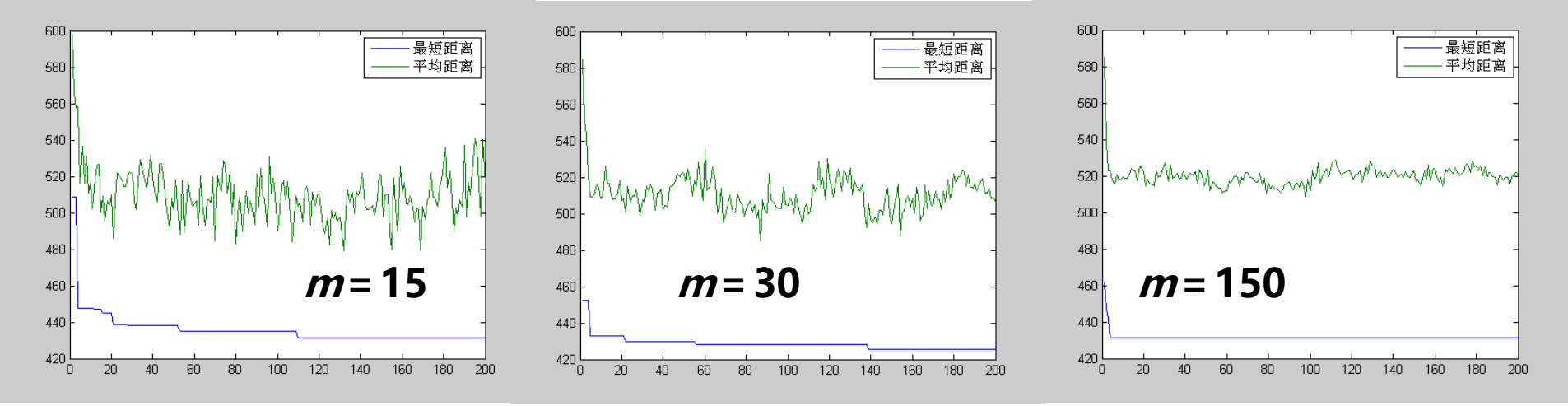




5 蚁群算法

◆ 蚁群优化算法的参数设置和基本属性

- 蚂蚁数m对算法的影响
- n=1m—1.5m 时，ant-cycle可以在有较好的全局收敛性和收敛速度。



◆ 蚁群优化算法的参数设置和基本属性

- 蚂蚁的初始分布

两种情况实验：

1 所有蚂蚁初始时刻放在同一城市；

2 蚂蚁分布在不同城市中。

第（2）中情况可获得较高性能。

在不同城市分布时，随机分布与统一分布的差别不大。

5 蚁群算法

◆ 蚁群算法优点与不足

- **优点**

较强的鲁棒性；

分布式计算；

易于与其他方法结合。

- **缺点**

计算量大、搜索时间较长；

容易出现停滞现象；

基本蚁群算法本质上是离散的，不适用于连续优化问题。

END