

第1章 演化计算导引

计算机科学与技术学院
吉林大学

源自武汉大学计算机学院黄竞伟 朱福喜 康
立山老师的课件 (略有修改)

1.1 演化计算

- 生物进化是指一个种群经过漫长时间所发生的累积变化，这些变化是由于生物体的基因变异或在繁殖期间以不同方式重组基因所产生的，而且这些变化可以被遗传到生物体的后代。

1.1 演化计算

- 生物的进化可以看作是一个优化过程，而优化的结果是能够很好地适应环境的生物体。现在地球上的种类繁多、结构复杂的生物都是通过漫长的由简单到复杂、由低级到高级的进化过程而得到的优化结果。
- 生物的进化过程也可以看作是在众多可能性中搜索“解”的一种方法。在生物中，众多可能性的集合是可能遗传序列的集合，而所要求的“解”则是能够适应环境的生物体。譬如对细菌入侵身体问题的一个令人惊奇的进化解就是哺乳动物的免疫系统。

1.1 演化计算

- 受到生物进化的启发，人们想到生物的进化机制可以被用来设计求解各类复杂问题的算法，演化计算就是模拟自然界生物进化过程而发展起来的一种通用问题求解方法，目前已形成计算机科学的一个研究领域。

1.1 演化计算

- 自然进化与问题求解术语之间的联系

进化	问题求解
环境	问题
个体	可能解
适应度	质量

1.1 演化计算

- **1948, Turing:**
“genetical or evolutionary search”
- **1962, Bremermann**
optimization through evolution and recombination
- **1964, Rechenberg**
Evolution strategies
- **1965, L. Fogel, Owens and Walsh**
Evolutionary programming

1.1 演化计算

- 1975, Holland
Genetic algorithms
- 1992, Koza
Genetic programming
- 1985: First international conference
International Conference on Genetic Algorithms (ICGA)
- 1990: First international conference in Europe
Parallel Problem Solving from Nature (PPSN)

1.1 演化计算

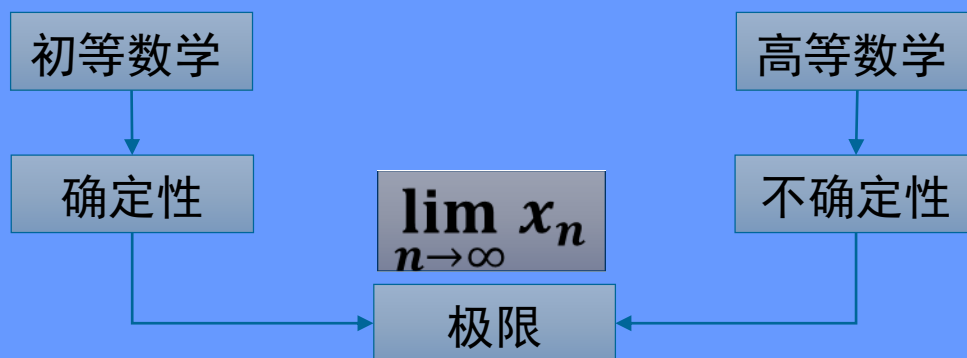
- **1993: First scientific EC journal (MIT Press)
Evolutionary Computation**
- **3 major EC conferences
The Congress on Evolutionary Computation (CEC),
Genetic and Evolutionary Computation Conference
(GECCO), and PPSN**
- **3 scientific core EC journals
Evolutionary Computation, IEEE Transactions on
Evolutionary Computation, Genetic Programming
and Evolvable Machine**

1.1 演化计算

- 目前，人们用演化计算表示模拟生物进化过程求解问题的整个研究领域，而遗传算法、遗传程序设计、演化策略和演化规划是演化计算研究领域的四个主要研究方向。

1.2 演化算法

- 演化计算所涉及的算法称为演化算法。有许多不同类型的演化算法，但所有演化算法的一个共同特点是求解问题的过程也就是模拟大自然生物进化/行为的过程。
- 演化算法的运行过程充满了**不确定性**



1.1 演化计算导引

- 演化算法模仿自然进化过程，在求解问题的过程中，保持一个个体的种群，每个个体表示问题的一个可能解，个体适应环境的程度用一个适应函数判断，每个个体按照适应函数来度量该个体作为问题解的好坏的程度，而度量值称为该个体的适应值。基于个体的适应值，某些较好的个体被选择，通过重组、变异等遗传算子的作用，产生一些新的个体，这些个体与种群中原来的个体竞争，形成下一代种群，继续这个过程，直到终止条件被满足。

1.2 演化算法

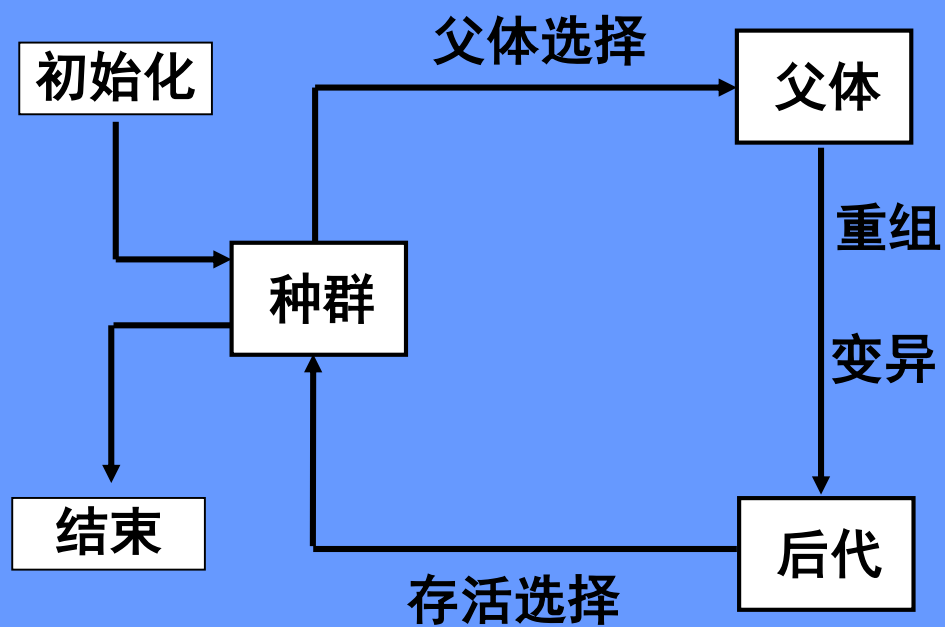


图1.1演化算法流程图

演化算法基本框架

Procedure Evolutionary_Algorithm

Begin

$t \leftarrow 0;$

initialize ($P(t)$); evaluate($P(t)$);

while (not termination condition) do

$P'(t) \leftarrow \text{parent_select}(P(t));$

$P''(t) \leftarrow \text{recombine}(P'(t));$

$P'''(t) \leftarrow \text{mutate}(P''(t));$

evaluate($P'''(t)$);

$P(t+1) \leftarrow \text{survivor_select}(P(t) \cup P'''(t));$

$t \leftarrow t + 1;$

end while

return the best solution;

End.

1.3 演化算法的设计

设计演化算法需要详细指明以下构成成分：

- 个体的编码；
- 适应函数；
- 父体选择策略；
- 变化算子；
- 存活选择策略；
- 参数设置；
- 种群的初始化；
- 终止准则。

1.3 演化算法的设计

- 个体的编码

个体（问题的解）的编码的目的是为了能够有效地执行遗传操作。

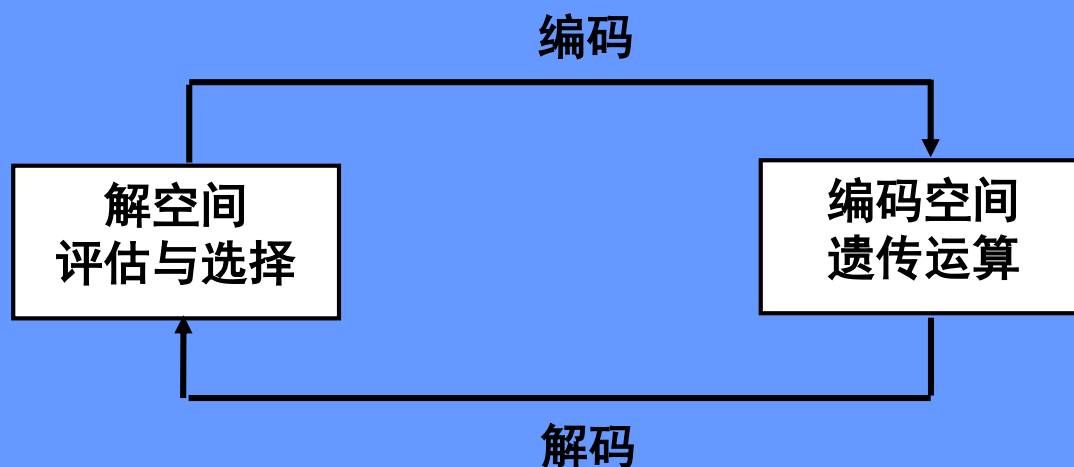


图1.2 编码空间与解空间

1.3 演化算法的设计

- 譬如，若一个优化问题的解空间由整数构成，那么一种编码方式是采用整数的二进制编码。
- 譬如，整数18编码为10010。
- 借用生物的术语，解的编码通常也称为染色体，或基因型，或个体。编码空间也称为基因型空间或搜索空间，演化算法对个体的遗传操作是在基因型空间中完成的。
- 一个染色体解码后所对应的解称为该染色体的表现型。解空间也称为表现型空间，演化算法对个体的评估和选择是在表现型空间中完成的。

1.3 演化算法的设计

- 适应函数

在演化算法中，适应函数是区别种群中个体好坏的唯一方法，是进行选择的依据，直接影响到演化算法的收敛速度和效率。

概念上，一个适应函数是从编码空间到实数域的一个函数。适应函数的作用是对基因空间中的每个染色体指定一个实数，称之为适应值，使得我们可以区分染色体的好坏。一般来说，一个染色体的适应值越大，表明该染色体越好。反之亦然。

1.3 演化算法的设计

- 例

$$\max \sum_{i=1}^n x_i, \quad x_i \in \{0,1\}$$

该问题的一个解为一个n 维向量 (x_1, x_2, \dots, x_n) 。
两个适应函数：

$$f(x) = \sum_{i=1}^n x_i \quad f(x) = \prod_{i=1}^n x_i$$

1.3 演化算法的设计

- 父体选择策略

演化算法通过从当前代种群产生下一代种群的方式，使得种群不断演化，从而逐步逼近问题的最优解。

类似于生物，产生下一代个体的一个重要手段是对当前代中的个体进行重组。而父体的选择是重组的基础。父体选择的目的是期望较好个体的重组能够得到更好的后代。若一个个体被选择通过重组产生新的个体，那么该个体称为一个父体。

1.3 演化算法的设计

- 并不是最好的一些个体总是被选择为父体。这是因为选择压力过大，搜索会过早终止，算法容易收敛到局部最优；当然，若选择压力过小，虽然可以保持种群的多样性，增加算法收敛到全局最优的概率，但算法的收敛的速度缓慢。
- 有许多不同的父体选择策略，通常父体的选择是随机的，使得具有较高适应值的个体被选择作为父体的概率较大，具有较低适应值的个体被选择作为父体的概率较小。

1.3 演化算法的设计

- 变化算子

重组算子:重组算子将两个或多个父体的信息进行组合, 得到一个或多个后代染色体, 期望通过重组较好父体的信息得到更好的后代染色体。

变异算子:变异算子改变一个染色体的信息, 得到一个新的染色体。其目的是通过引入新的染色体来增加种群的多样性, 扩大搜索空间, 从而避免演化算法陷入局部最优。

1.3 演化算法的设计

- 在不同的演化算法中，变化算子所起的作用有所不同
- 在遗传算法中，重组算子是一个主要的变化算子，变异算子起辅助性的作用。
- 在演化策略中，变异算子是一种主要的遗传算子，重组算子起辅助性的作用。
- 在演化规划中，只使用变异算子，根本不使用重组算子。

1.3 演化算法的设计

- 存活选择策略
- 存活选择的作用是从当前代种群和由父体所产生的后代个体中选择出个体以形成下一代种群。
- 通常父体选择是随机的，而存活选择是确定的。
- 存活选择倾向于适应值较高的个体。譬如说，按照当前代种群中的个体和所有后代个体按照适应值排序，然后选择出最好的一些个体形成下一代种群。

1.3 演化算法的设计

- 参数设置

不同的演化算法有不同的参数。譬如说，遗传算法的种群规模、杂交概率和变异概率；演化策略中的变异步长；遗传程序设计中的树的深度等。

- 参数设置有以下两种主要方法：

(1) 参数调整：参数调整是算法设计中的一个典型方法，也是实际中用的最多的方法之一。

(2) 参数控制：参数控制是预先给定初始参数值，在算法的运行过程中，对参数进行动态地变更。

1.3 演化算法的设计

- 种群的初始化

- (1) 随机地产生初始种群

- 随机产生的目的是使初始种群中的个体均匀地分布在搜索空间中。

- (2) 使用某种启发式算法产生初始种群

- 使用某种启发式算法产生初始种群，可使初始种群中包含较好的解，这可能使算法能更快地发现问题的最好解。但也可能会导致算法过早收敛到局部最优解。

1.3 演化算法的设计

- 终止条件

- (1) 算法的运行时间达到预先指定的最大运行时间；
- (2) 算法计算个体适应值的总数达到预先指定的最大次数
- (3) 算法的演化代数达到预先指定的最大代数；
- (4) 在连续若干代以后，个体适应值没有明显的改进；
- (5) 种群的多样性降低到某个预先指定的阈值。

算法的实际终止条件可能是上述若干条件的析取。

最常用的是预先指定一个最大演化代数。

1.4 演化算法的特点

- 演化算法具有以下特点：
 - (1) 演化算法采用种群的方式组织搜索，它从多个点而不是一个点开始搜索；
 - (2) 演化算法不是利用函数的导数或其它相关知识，而是仅仅利用个体适应值的信息来指导搜索；
 - (3) 演化算法使用概率转移规则而不是确定性的转移规则。

演化算法的主要优点在于概念简单，易于实现。

1.5 演化算法的性能评估

- 设计演化算法的目的

设计一个演化算法的目的可以是为了求解一个应用问题，也可以是为了进行学术研究。

- 应用问题可以划分为如下两类：

- (1) 设计型问题；
- (2) 重复型问题。

1.5 演化算法的性能评估

对于学术研究来说，下面是一些常见的实验目标：

- 对某个给定问题，演化算法可以得到一个好的解；
- 证明演化算法可应用于某个领域；
- 证明具有某些新特征的演化算法比某些演化算法要好；
- 证明对某些问题演化算法比某些传统算法要好；
- 观察算法的性能与参数设置之间的关系。

无论是学术研究还是应用研究，所有实验最为重要的目标是算法性能的度量。

1.5 演化算法的性能评估

性能度量

三种基本的算法度量标准：成功率，有效性和效率。

(1) 成功率:在给定的时间内，算法终止时找到所需质量解的运行次数与运行总次数之比。

(2) 有效性:有效性是指在指定的运行时间或运行代数内算法的求解质量。

- 平均最好适应值；
- 历史最好 (Best-ever) 适应值；
- 历史最差 (Worst-ever) 适应值。

1.5 演化算法的性能评估

(3) 效率

演化算法的效率可用算法得到满意解时所花费的计算量或计算时间来度量。

用计算时间度量效率的方式依赖于所用的计算机的类型、操作系统和编译软件等。

1.5 演化算法的性能评估

演化算法是一种产生-检验类型的算法，对这一类算法，一种公共的效率度量方法是计数搜索空间中被访问点的个数。因为演化算法对每个新产生的个体都要进行评估（计算适应值），这种度量方式相当于计算个体的评估次数。但由于演化算法的随机特性，我们常用独立运行算法多次的平均个体评估次数来度量算法的效率。

1.5 演化算法的性能评估

个体的平均评估次数通常可以给出算法效率的公平比较，但也存在以下缺陷：

- 有些隐藏的工作是不可见的。例如，若变异算子使用的某种局部搜索策略，这种额外的计算可以提高算法的性能，但在对个体的评估是不可见的；
- 某些个体评估方式可能比其它的个体评估方式要花费更多的时间；
- 若在一次演化代中，个体评估所花费的时间比执行其它步骤所花费的时间要少的多的话，个体的平均评估次数就不能正确反映算法的效率。

1.5 演化算法的性能评估

- 测试问题

测试问题实例通常用以下三种方式产生：

(1) 用预先定义的问题实例；

用预先定义的问题实例相当于从互连网上的问题实例库或其它参考文献中得到准备好的问题实例。

T. Back和Z. Michalwicz建议测试函数集中应包含以下函数：

- 若干单峰函数；
- 若干具有很多局部最优的多峰函数；

1.5 演化算法的性能评估

- 带有随机扰动的目标函数；
- 带有约束条件的目标函数；
- 高维目标函数。

(2) 用问题实例生成器

近年来，演化计算领域对用问题实例生成器进行实验研究越来越感兴趣。通常，这些问题实例生成器有一些与问题相关的参数，通过指定这些参数，可以得到具有指定参数值的随机问题实例。

1.5 演化算法的性能评估

(3) 用现实世界问题

从应用的角度来看，用现实世界问题实例进行实验研究是十分恰当的。然而，这种方式也有一些不足之处。实际问题可能过于复杂，实验数据较少，而且实验结果不易推广。

参考文献

- **A.E.Eiben and J.E.Smith. Introduction to Evolutionary Computing. Springer- Verlag Berlin, 2003.**
- **潘正军, 康立山, 陈毓屏. 演化计算. 清华大学出版社, 北京, 1998.**
- **J. R. Koza. Genetic Programming. MIT Press, Cambridge, MA, 1992.**