

遗传算法综述及简单应用实例课件



| CATALOGUE |

目录

- 遗传算法概述
- 遗传算法的组成要素
- 遗传算法的应用领域
- 遗传算法的简单应用实例
- 遗传算法的优缺点分析
- 遗传算法的未来展望与研究方向

01

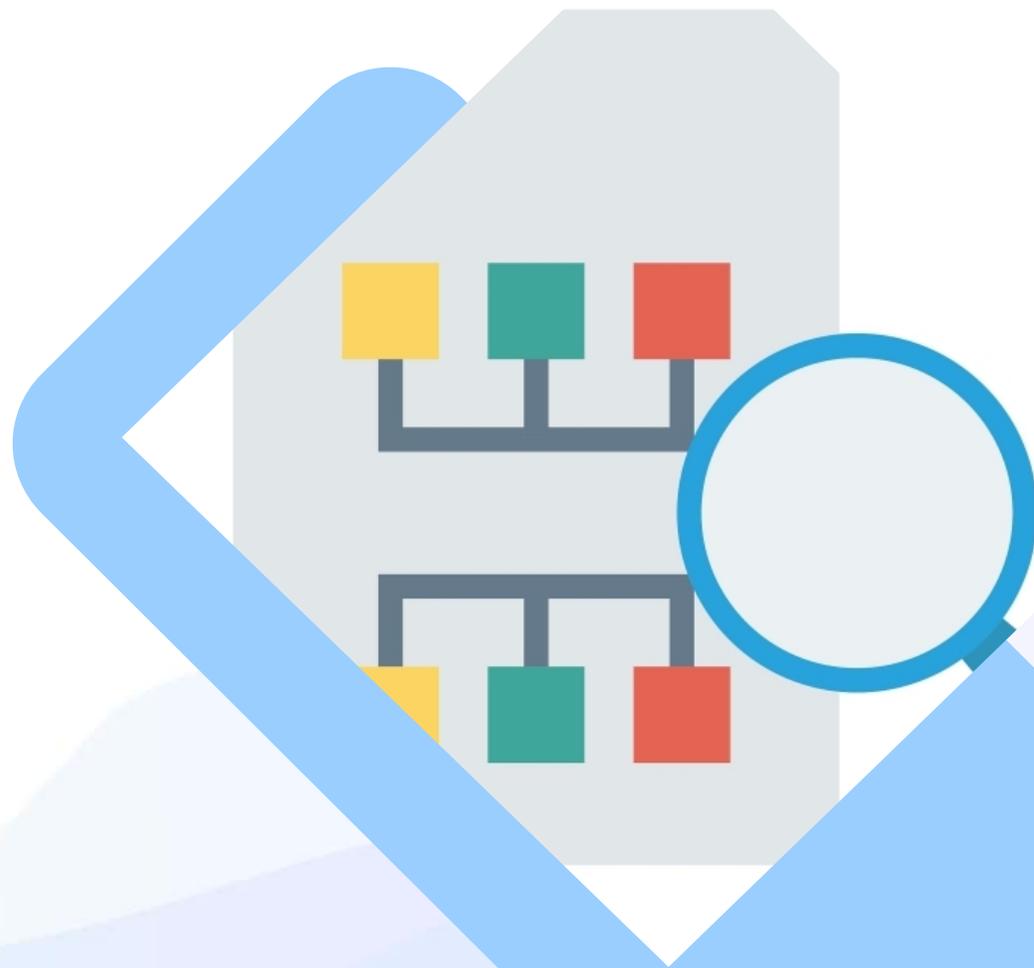


遗传算法概述



遗传算法的基本概念

- 遗传算法是一种模拟自然选择和遗传机制的优化算法，通过模拟生物进化过程中的基因选择、交叉和变异等过程，寻找最优解。
- 它将问题的解空间映射到生物基因的编码空间，每个解称为一个个体或染色体，并按照适应度函数评估个体的适应度。
- 遗传算法通过不断迭代，根据适应度选择、交叉和变异等操作，逐步淘汰适应度低的个体，保留适应度高的个体，最终得到适应度最高的个体作为最优解。





遗传算法的起源与发展

遗传算法的起源可以追溯到20世纪60年代，当时美国密歇根大学的John Holland教授提出了“适应性系统”的概念，为遗传算法奠定了理论基础。

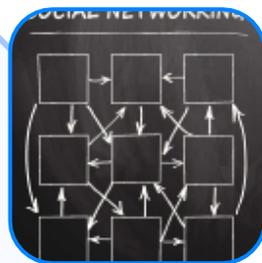
1975年，Holland教授出版了《自然系统和人工系统的适配性》一书，正式提出了“遗传算法”这一术语。

之后，遗传算法在许多领域得到了广泛的应用和发展，包括函数优化、机器学习、模式识别、数据挖掘等。



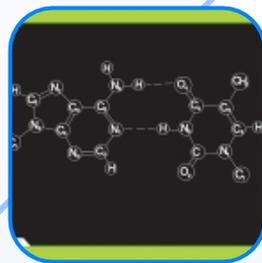
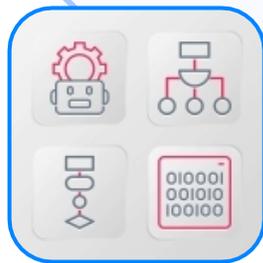
遗传算法的基本原理

遗传算法的基本原理是通过模拟自然选择和遗传机制来寻找最优解。



它将问题的解空间映射到生物基因的编码空间，每个解称为一个个体或染色体。

适应度函数用于评估个体的适应度，根据适应度进行选择、交叉和变异等操作。



通过不断迭代，逐步淘汰适应度低的个体，保留适应度高的个体，最终得到适应度最高的个体作为最优解。

02



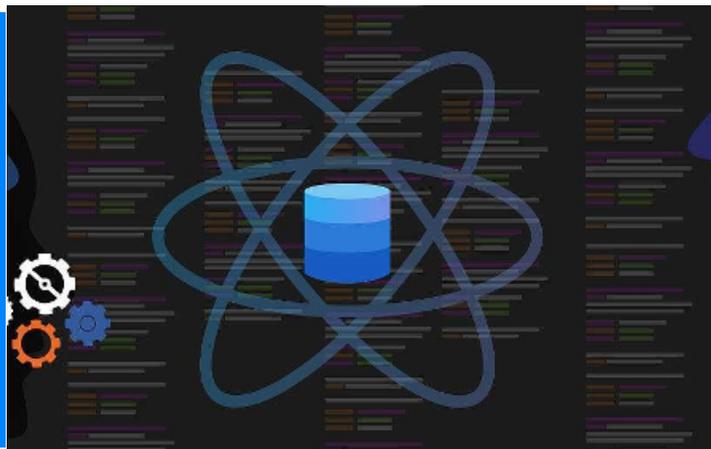
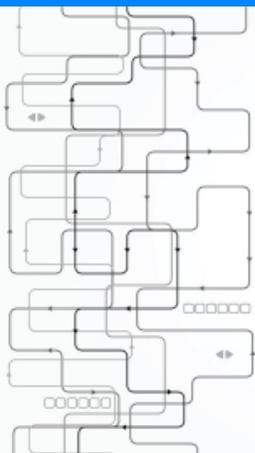
遗传算法的组成要素



编码方式

二进制编码

是最常用的编码方式，通过二进制位表示解空间中的解，具有易于实现、搜索空间大、运算简单等优点。



实数编码

将问题的解用实数表示，适用于连续型优化问题，能够提高算法的搜索效率。

排列编码

适用于组合优化问题，如旅行商问题等，通过排列方式表示解。





适应度函数

● 定义

用于评估解的优劣程度，是遗传算法中选择的依据。

● 设计原则

应尽量简单、明了，能够反映解的优劣程度，且计算量不宜过大。

● 多目标优化问题

适应度函数的设计需考虑多目标优化问题中的权重和优先级。





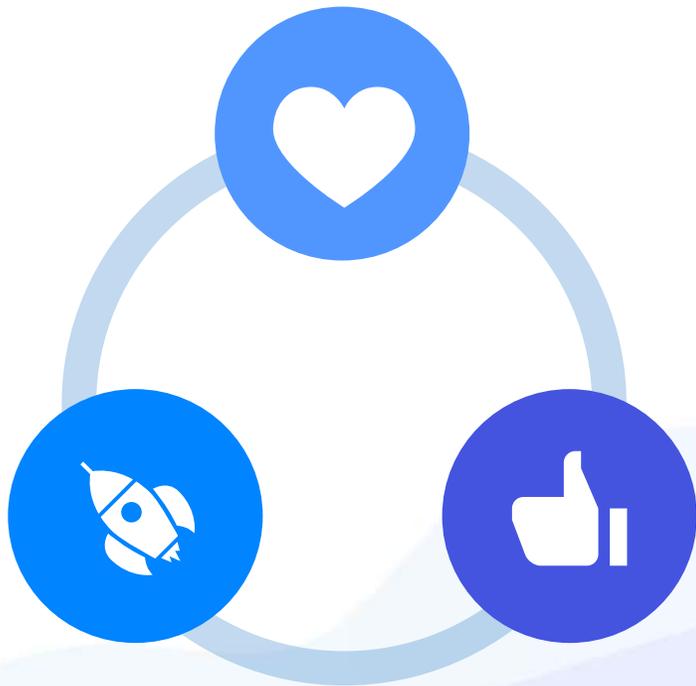
选择操作

轮盘赌选择

根据适应度值的大小决定解被选中的概率。

锦标赛选择

从群体中随机选择一定数量的解，再从中选择适应度最好的解。



秩选择

根据适应度值将解排序，按顺序选择。



交叉操作

● 单点交叉

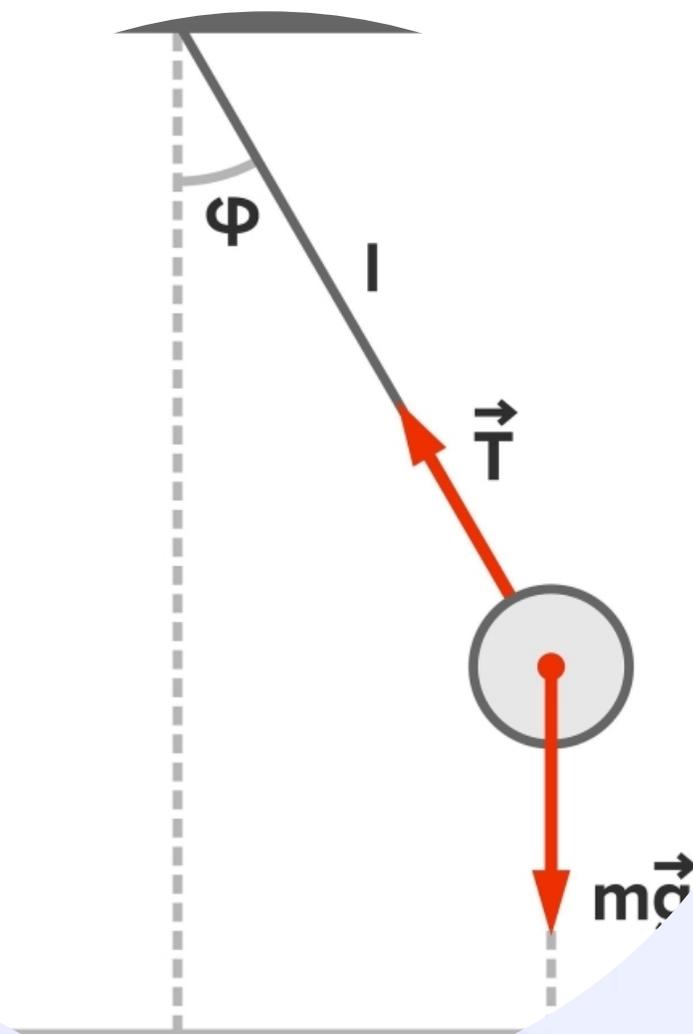
在某一位点处将两个父代解进行交叉。

● 多点交叉

在多个位点处进行交叉，产生新的解。

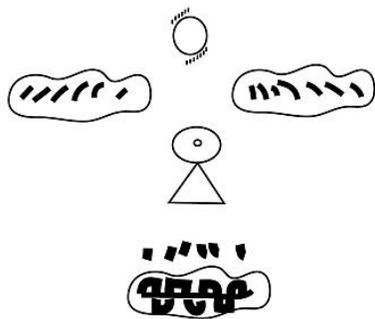
● 均匀交叉

两个父代解的每一位都有一定的概率被交叉。





变异操作



定义

通过随机改变解的一小部分基因，增加种群的多样性。

方法

常见的变异操作有倒位、交换、突变等。

作用

避免算法陷入局部最优解，提高全局搜索能力。

03



遗传算法的应用领域



函数优化

函数优化问题

遗传算法可用于求解连续函数优化问题，通过不断迭代和选择，寻找函数的最优解。

应用实例

例如，在机器学习中的特征选择问题，可以使用遗传算法来优化特征权重，提高分类准确率。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/668064134046006060>