

# 遗传算法总结

By xudong

## 内容提要

遗传算法的实现过程

遗传算法基本理论

遗传算法的三个算例

- TSP 问题
- 装箱问题
- 网络结构优化

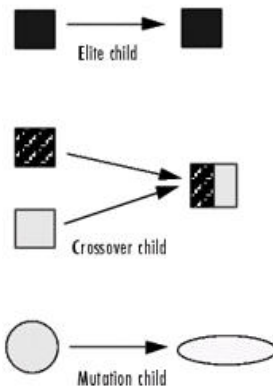
## 遗传算法的实现过程

产生初始种群

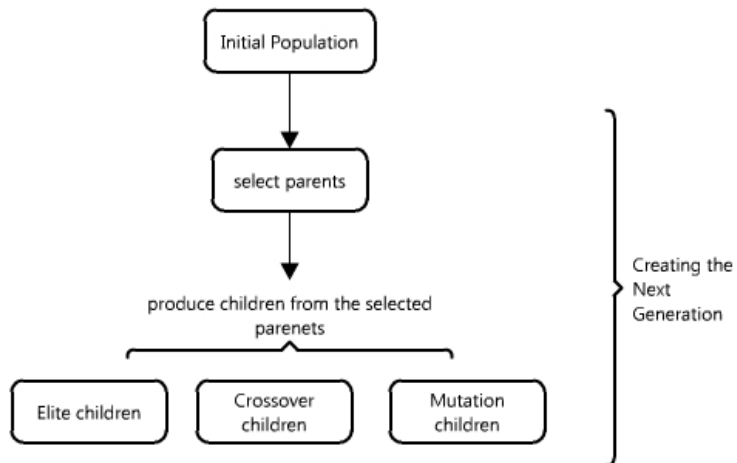
在适应度函数的基础上选择 parents

从 parents 中产生 children 有三种途径:

- 将 parents 中最优秀的个体直接保留到 children
- 杂交
- 变异



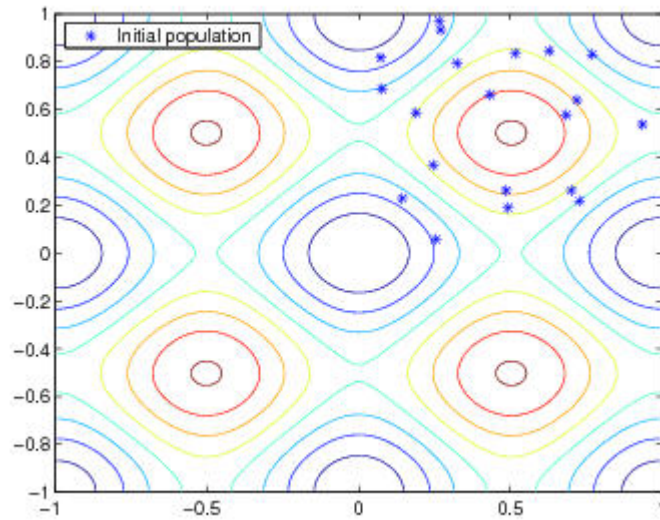
遗传算法实现流程



下面以一个函数优化的过程为例说明遗传算法的工作过程，材料来自 matlab 中遗传算法的算法说明。

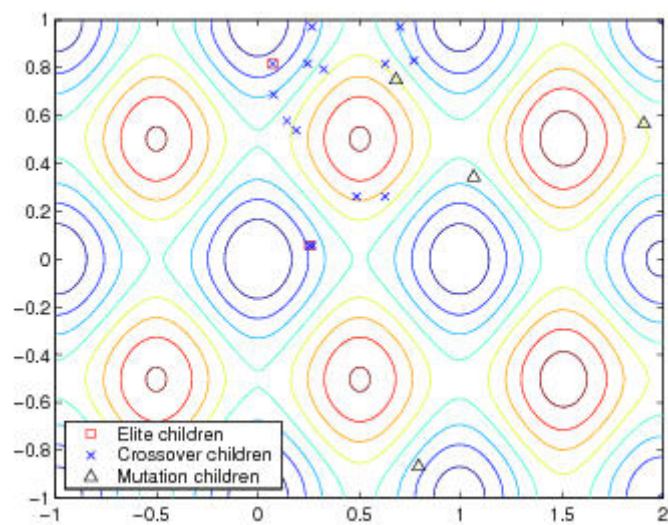
### 【生成初始种群】

这是函数的等值线图，图中的蓝点表示初始种群中的个体。



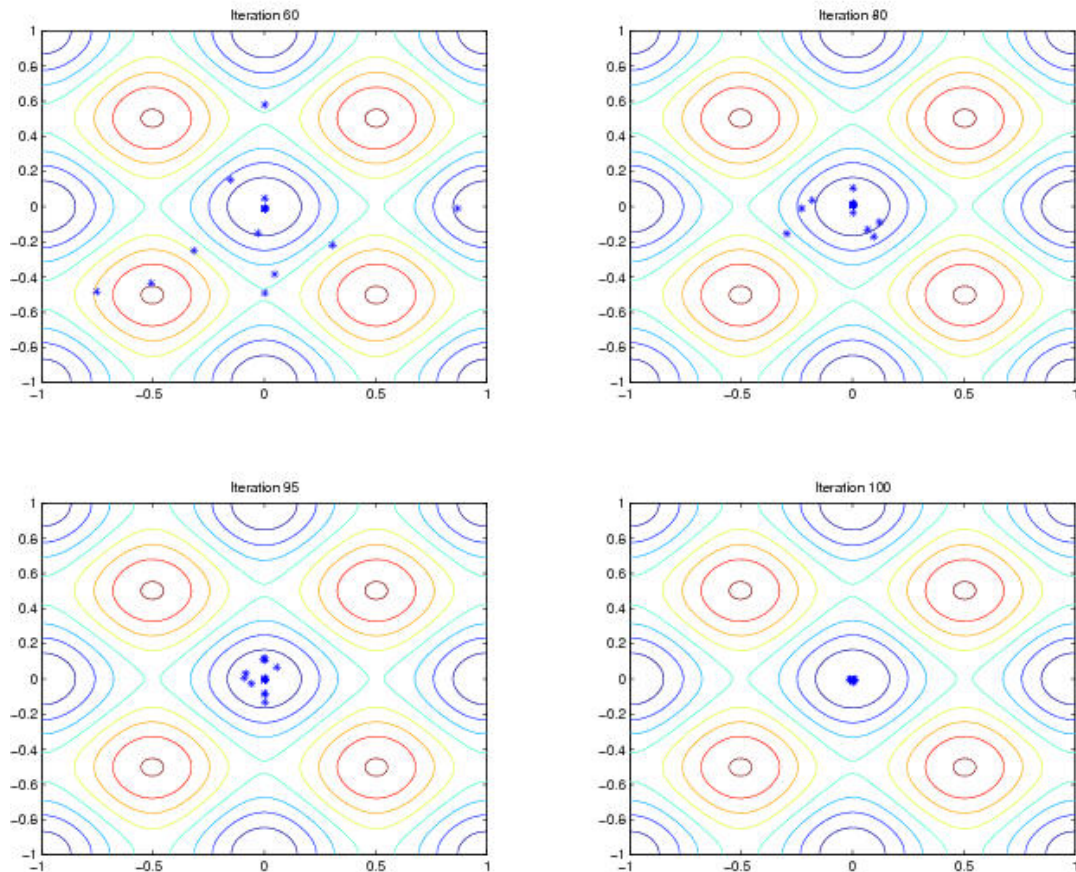
### 【生成下一代种群】

图中的符号表示下一代个体，具体不同符号表示不同的产生途径。



### 【迭代过程】

可以看出，随着迭代的增加，整个种群逐渐收敛到最优解。



## 遗传算法基本理论

这里只叙述定理的内容和评价，定理中所涉及的概念和定理的推导过程大家感兴趣的话可以翻翻书。

### 【模式定理】

内容：遗传算法中，在选择、交叉、变异算子的作用下，具有低阶、短的定义长度，并且平均适应度高于群体平均适应度的模式将按指数级增长。

评价：看到这个定理首先想到的就是《自私的基因中》的核心观点，基因是进化的基本单位。短小，并且适应度高的基因在进化过程中被保留下来，并且在群体中不断增加。

### 【积木块假定】

内容：基因能够互相拼接在一起，形成适应度更高的编码串。

评价：模式定理是说遗传算法能够产生高质量的建筑砖头（基因），积木块假定是说可以用这些高质量的建筑砖头建造高质量的大楼（编码串）。需要注意的是，我们把上述内容称为假定，而非定理，这是因为有时候积木块假定并不成立，也就是说高质量的砖头不一定能够建成高质量的大楼。出现这种情况的原因是  $1+1 < 2$  的现象，各个基因之间高度相关，适应度很高的两个基因之间存在很强的负交互作用，结果导致整体的适应度下降。

### 【遗传算法收敛性】

内容

1. 不保留优秀个体，收敛于最优解的概率小于 1
2. 保留优秀个体，遗传算法收敛于最优解的概率等于 1

评价：这两个定理说明了我们为什么在遗传算法中保留优秀个体的一个原因，保留优秀

个体的另外一个原因是保证遗传算法每一代的解总是单调减少的(假设目标是极小化目标函数)。但是优秀个体不宜过多,优秀个体保留过度导致种群的多样性下降,种群过早的收敛到局部最优解。

## 遗传算法的三个算例

### 【装箱问题和 TSP 问题】

装箱问题和 TSP 问题不做详细说明了, matlab 的 demo 中有使用遗传算法工具箱求解 TSP 问题的例子,有兴趣的朋友可以看看。

	解的编码方式	适应度函数	杂交	变异
TSP	所有城市的一种排列	按给定排列走完所有城市的路程	随机选定排列的一个片段, 翻转	随机选定排列中的两个城市, 交换位置
装箱问题	所有物品的一种排列	装完所有物品需要的箱子数量	同上	同上

### 【网络优化问题】

#### 网络的可靠性

给定一个网络如果网络中任意两点是连通的,那么称这个网络处于正常工作状态。由于网络中两点之间的连接可能坏掉,如果连接损坏到一定程度,整个网络就无法保持正常工作状态。我们定义整个网络处于正常工作状态的概率为网络可靠性。

#### 网络可靠性的计算

给定一个网络,网络中每条边正常工作的概率为  $p$ ,如何计算整个网络的可靠性?使用组合方法精确的计算概率十分困难,计算量也随着网络规模以指数速度增加。我们使用随机试验的方法来获得给定网络的可靠性,3000 次的随机试验可以保证可靠性的精度在 1% 以内。

#### 网络优化问题

网络中连接越多可靠性越高,但是再两个节点之间建立连接是有成本的。现在的问题是如何优化网络中的连接结构使得满足可靠性要求的前提下最小化建立网络的成本。

### 【网络优化的遗传算法】

#### 解得编码方式:

对网络中的节点编号,一个 0-1 矩阵就能表示一个网络的连接情况。为了方便杂交合变异,我们将连接矩阵转换成 0-1 编码串,  $x_{12} x_{13} \dots x_{23} \dots x_{n-1,n}$ 。

#### 适应度函数:

我们使用成本作为适应度函数,目标是 minimize 成本。

$$C = \sum_{i < j} c_{ij} x_{ij}$$

但是还有一个问题,0-1 编码方式在杂交变异之后可能会出现不满足约束条件的情况,为解决这个问题,我们在成本函数中添加罚函数,如果可靠性小于要求时就施加惩罚。添加罚函数之后的适应度函数为

$$\min \sum_{i < j} c_{ij} x_{ij} + K \delta(r_{\min} - r)$$

其中  $\delta(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x > 0 \end{cases}$ ,  $K$  表示惩罚系数,  $K$  越大表示惩罚越重。

#### 杂交

采用单点杂交的方式：随机选择一个位置，两个父母交换该位置之后的编码串。

### 变异

以一定概率将 01 串中的 0 变为 1，1 变为 0

## 总结：

传统的函数优化方法都是以梯度为基础的，对于那些较好的函数(能够获得梯度信息)，传统的优化方法也是最优的。但是对于那些性态很差的函数优化问题，组合优化问题传统的方法无法解决。而遗传算法能够高效的获得满足要求的解，以上几个组合优化的例子很好的说明了这一点。