# 自动组卷的实现

自动组卷就是用户自己选择想出的题型及其数量，设置难度，知识点分布(大纲)，系统自动生成试卷。这里主要用到了遗传算法的相关知识。

遗传算法是一种启发式搜索，其灵感来自查尔斯·达尔文的自然进化论。该算法反映了自然选择的过程，其中选择最适合的个体进行繁殖以产生下一代的后代。

在应用中，遗传算法的难点主要有三个，染色体的编码的设计、适应度函数的涉及、交叉点的设计，接下来，分析这三个难点的实现。

1. 染色体的编码的设计

采用实数编码方案，将一份试卷映射为一个染色体，组成该试卷的每道题的题号作为基因，基因的值直接用试题号表示，每种题型的题号放在一起，按题型分段，在随后的遗传算子操作时也按段进行，保证了每种题型的题目总数不变。

1. 适应度函数的设计

适应度函数跟试卷难度系数和知识点覆盖率有关。

试卷难度系数公式：P=∑Di×Si/∑Si；其中i=1，2，…N，N是试卷所含的题目数，Ｄi，Ｓi分别是第i题的难度系数和分数。

知识点覆盖率系数公式：D = (1 – M/N)，例如期望本试卷包含N个知识点，而一个个体中所有题目知识点的并集中包含M个（M<=N），则知识点的覆盖率为M/N。用户的期望难度系数EP与试卷难度系数P之差越小越好，知识点覆盖率越大越好，因此适应度函数为：f=1- D \*f1-|EP-P|\*f2。其中f1为知识点分布的权重，f2为难度系数所占权重。当f1=0时退化为只限制试题难度系数，当f2=0时退化为只限制知识点分布。这里(1- M/N)是因为覆盖率越大，M/N这个值就越大，因为适应度是一个正向的反馈值，所以要用(1-M/N)。

所以仅需要输入好试卷标题、试卷难度、组卷数量、各类型题目的题型和数量之后，系统将会根据输入的试 卷难度和课程设置的大纲下的各个章节的掌握度，自动计算出整张试卷每个章节的题目占所有题目的权重，在最后得到符合输入的试卷整体难度的试卷。

1. 交叉点的设计

随机交叉点。

通过遗传算法，将多张试卷进行交叉遗传并变异，直到难度系数与用户难度系数相当，知识点覆盖率最大的试卷为止。