# 遗传算法；解决背包最优化问题

**问题描述**： 个物品，对物品 ，价值为 ，重量为 ，背包容量是 。如何选取物品装入背包，使背包中的价值最大。

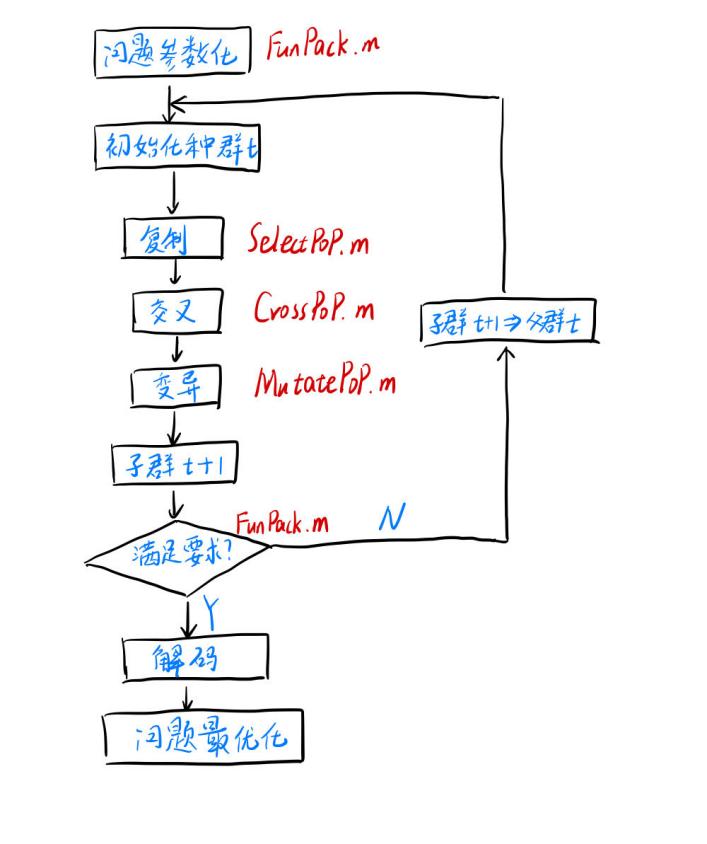
**例如**： 个物品，对物品 ，价值为 ，重量为 ，背包容量是 ，见下表。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

考察编码： ，对应物品 被放入背包，此时背包重量为 ，价值为 ，经验证，为该情况最优解。

**大作业要求**： 个物品，价值、重量、背包容量自定义，设计遗传算法求最优解。

**程序设计**：

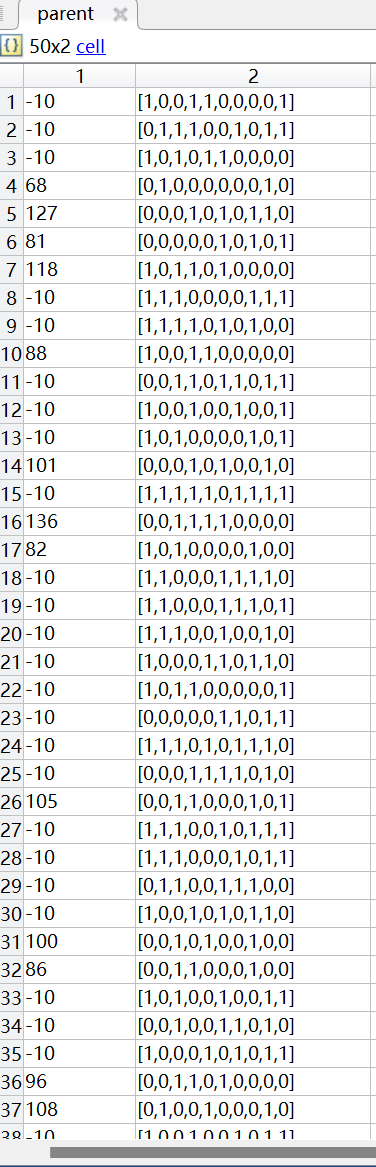


**程序分析**：

1. **InitPop.m**

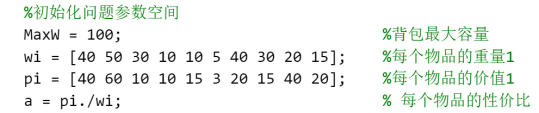
该文件包含函数parent = InitPop(nPop,nVar)，声明parent数组内存空间，每行为每个个体的适应度水平以及对应的DNA数据，数据为初始化随机种群，见下图。

（注：其中，适应度为-10部分，为超出该问题中背包容量，对种群适应度所做出的惩罚，对应的DNA数据中，第i位中的1代表第i个物品被放入背包中，0反之。）



1. **FunPack.m**

该文件包含函数y = FunPack(x)，其中可以定义问题的参数空间，如下代码。



并且该函数对输入的个体x，进行适应度解码，并对超出最大背包容量个体适应度（即背包中容量价值）做出惩罚，使进化朝有利方向前进。

（注：引入惩罚机制是为了更方便的控制种群朝满足问题参数空间的方向进化。曾尝试过将惩罚机制设为：若超出，种群适应度减少固定值；若超出，种群适应度为0。对于前者，若减少量没有增加一件物品适应度来的多，那么进化方向错误；对于后者，虽然将适应度设为0，但若整体物品价值均较小接近0，那么会导致进化方向不稳定。故将惩罚设置为一个负数，理论上讲，只要不是较小的负数，那么该算法鲁棒性较好。该代码为-10。）

1. **SelectPop.m**

该文件包含函数y = SelectPop(parent)，其中利用二元锦标赛选择适应度高的个体被复制到子代。

1. **CrossPop.m**

该文件包含函数[y1,y2] = CrossPop(x1,x2)，其中对传入的染色体分段进行多次交叉。首先从染色体DNA个数中随机选择交换次数，并且随机选择一个交换点，然后判断允许交换的最大DNA数（即该交换点后DNA数），再随机选择允许交换个数进行交叉。

（注：若只随机选择一个位置，对后半段染色体DNA进行交换，会导致越靠后的染色体DNA存活概率大。该算法通过随机选择交换点，然后随机选择允许的交换数，解决了前述问题。随机选择交换次数只是为了扩大进化可能的搜索方向，因为该问题非常容易陷入局部最优解，并且局部最优解与全局最优解的步长较大。背包物品相差很多，但总价值相差很小的情况时常出现。）

1. **MutatePop.m**

该文件包含函数y = MutatePop(x,mu,mn)，其中对输入的染色体x进行mn次变异，每次变异的概率为mu。并且随机选择一个染色体中的DNA进行变异，变异在该问题中代表是否放入背包的状态取反。

1. **project.m**

该文件是工程代码总目录，并且记录有测试数据中的最优个体。首先经过了上述算法的初始化种群、种群的复制交叉变异，然后对父群和子群融合，在将其适应度按降序排列，选择种群数量的最优个体进入下一代的进化。最终，我们记录了进化以来所出现过的最优染色体maxPop，以及最终的种群parent。

（注：在该问题中，该算法应用了较多的交叉以及变异，是为了扩大搜索空间，使问题跳出局部最优解，达到全局最优。在该算法中，设置了最大迭代次数为1000。经过测试，该算法在第数十代就进化出了最优个体，并且很好的遗传了下来。如果该问题不使用较多的随机交叉以及大概率的变异，那么即使经过了大量的进化过程，也很难走出局部最优解，获得很鲁棒的最优解。究其原因，在于该问题的局部最优解之间，以及局部最优解与全局最优解之间的参数空间距离很远，例如个体[0 0 0 1 1 1 0 1 0 0]的价值为128与个体[0 1 1 0 0 0 0 1 1 0]的价值为128，尽管染色体差异巨大，汉明距离为6，但价值持平。）

**附录**：

1. 数据库

MaxW = 100; %背包最大容量

wi = [40 50 30 10 10 5 40 30 20 15]; %每个物品的重量1

pi = [40 60 10 10 15 3 20 15 40 20]; %每个物品的价值1

wi = [40 20 30 17 14 5 13 11 22 10]; %每个物品的重量2

pi = [20 27 24 10 15 3 20 15 40 15]; %每个物品的价值2

wi = [16 25 36 17 14 5 13 11 6 10]; %每个物品的重量3

pi = [20 23 33 28 15 2 20 15 9 14]; %每个物品的价值3

wi = [13 18 14 16 10 12 17 11 19 15]; %每个物品的重量4

pi = [20 23 13 28 15 7 11 21 7 6]; %每个物品的价值4

wi = [27 25 23 21 29 27 25 22 26 30]; %每个物品的重量5

pi = [22 29 34 26 40 36 23 26 39 19]; %每个物品的价值5

1. 数据库最优解

记录历史出现的适应度最大的个体

最优染色体 价值

数据库1：0 1 0 0 1 1 0 0 1 1 138

数据库2：0 1 0 0 1 1 1 1 1 1 135

数据库3：1 1 0 1 0 0 1 1 1 1 129

数据库4：1 1 1 1 1 0 1 1 0 0 131

数据库5：0 0 1 1 1 0 0 0 1 0 139(重量99)/ 0 0 1 0 1 0 0 1 1 0 139(重量100)