

遗传算法解决图论最大割问题

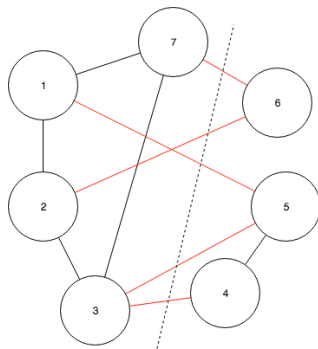
1004161209 郝进 计算机科学与技术二班

摘要

遗传算法通常实现方式为一种计算机模拟。对于一个最优化问题，一定数量的候选解（称为个体）可抽象表示为染色体，使种群向更好的解进化，从而求得问题的优质解。最大割问题是将一个无向图切成 2 个部分（子图），从而使得 2 个子图之间的边数最多。最大割问题属于 NP 问题，本文研究了如何将遗传算法应用于最大割问题的求解上。

关键字：遗传算法、图论、最大割

一、 问题分析



图一

最大割问题：给定一个无向图，将其切成 2 个部分（子图），从而使得 2 个子图之间的边数最多。即从边集中选择一个最大的边割子集。这个问题是图论中一个典型的 NP 难组合优化问题。

二、 建模

设遗传算法表示如下：

$$SGA=(C,E,P_0,M,\phi,\Gamma,\psi,T)$$

其中，C——个体的编码方法；E——个体适应度评价函数； P_0 ——初始种群；M——种群大小； ϕ ——选择算子； Γ ——交叉算子； ψ ——变异算子；T——遗传运算终止条件。

具体步骤如下：

1. 染色体编码

即把一个问题的可行解从其解空间转换到遗传算法的搜索空间的转换方法。遗传算法在进行搜索之前先将解空间的解表示成遗传算法的基因型串(也就是染色体)结构数据，这些串结构数据的不同组合就构成了不同的

点。常见的编码方法有二进制编码、格雷码编码、浮点数编码、各参数级联编码、多参数交叉编码等。

2. 初始群体生成

设置最大进化代数 T，群体大小 M，交叉概率 P_C ，变异概率 P_M ，随机生成 M 个个体作为初始群体 P_0 。

3. 适应度评估检测

适应度函数表明个体或解的优劣性。对于不同的问题，适应度函数的定义方式不同。根据具体问题，计算群体 P(t)中各个个体的适应度

4. 遗传算子

(1) 选择

选择操作从旧群体中以一定概率选择优良个体组成新的种群，以繁殖得到下一代个体。个体被选中的概率跟适应度值有关，个体适应度值越高，被选中的概率越大。

(2) 交叉

交叉操作是指从种群中随机选择两个个体，通过两个染色体的交换组合，把父串的优秀特征遗传给子串，从而产生新的优秀个体。

(3) 变异

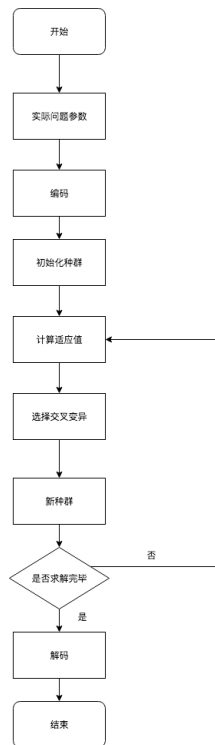
为了防止遗传算法在优化过程中陷入局部最优解，在搜索过程中，需要对个体进行变异，在实际应用中，主要采用单点变异，也叫位变异，即只需要对基因序列中某一个位进行变异，以二进

制编码为例，即 0 变为 1，而 1 变为 0。

群体 $P(t)$ 经过选择、交叉、变异运算后得到下一代群体 $P(t+1)$ 。

5. 终止判断条件

a 若 $t \leq T$ ，则 $t \leftarrow t+1$ ，转到步骤 2；若 $t > T$ ，则以进化过程中所得到的具有最大适应度的个体作为最好的解输出，终止运算。



图二

在解决最大割问题中：

1. 步骤一

编码方式为：对无向图的每个节点进行编号，把无向图切成两个子图，划为子图 1 的用 1 表示，划为子图 2 的用 0 表示。

2. 步骤二

假设有 7 个节点，则我们可以设定种群的大小为 10，编码位数为 7 位（因为有 7 个节点）

3. 步骤三

设适应度函数 $F(x)$ 为两部分之间的边数。

4. 步骤四

(1) 用轮盘赌方法从群体中随机选

择两个父代。

(2) 对选取的父代进行杂交得到子代，其中杂交方法为若两个父代的同一节点在相同集合中，则保留；否则，对随机分配该节点至任意集合中。

(3) 设定遗传概率，在 0.05 的概率下，将子代的某个节点从一个集合移动到另一个集合中。即将 0 改为 1。

三、 实现

代码通过 C++ 实现

代码分为 8 个函数模块构成，其中

main() 函数构建了算法的主体框架；

Init() 函数则是完成所有动态数组的初始化处理，读入数据，并存储图；

Genetic_Construction(), Genetic_Crossover(), Genetic_Mutation(), Genetic_Update() 这 4 个

函数则为整个遗传算法（初始化种群、选择、交叉、变异和更新群体）的实现过程；

Check() 函数则用以检验分配方案的实际被切割边数与存储的被切割边数是否一致；

Output() 函数则设置了结果的输出格式。

(b). 数据输入格式说明：

数据分为两部分第一部分为以 'p' 开头的总概栏，指明了总的点数和边数；第二部分则是以 'e' 开头的两个点的标号，代表这两个点相连接。

项目地址：

<https://github.com/Bocity/MaxCut>

```
200
p 7 18
e 1 4
e 1 5
e 1 6
e 1 7
e 2 3
e 2 4
e 2 5
e 2 6
e 2 7
e 3 4
e 3 5
e 3 6
e 3 7
e 4 5
e 4 6
e 5 6
e 5 7
e 6 7
*****
Check_Max_Cut = 12
Max_Cut = 12
Distribution of each vertex :
0 0 0 1 1 1 1
*****
```

图三

四、 参考资料

[1] <https://zh.wikipedia.org/wiki/%E9%81%97%E4%BC%A0%E7%AE%97%E6%B3%95>