姓名: 周丙瑞

学号: E11514058

【人工智能实验报告】

　 遗传算法求解TSP问题

【实验目的】

理解遗传算法的原理, 解决实际问题

【实验原理】

１.算法：遗传算法(GA)

　　模拟生物进化的过程, 优化搜索

２.用遗传算法求解TSP问题

参数:

种群个数、城市(节点)个数、杂交概率、突变概率、遗传代数

数据结构:

１)Graph:

存放任意两城市之间的距离(有向图)，实验用随机函数生成

２)TSP\_Solution：种群个体

Path\_Length : 路径长度

Path:存放路径

Select\_Rate在当代被选择的概率

3.算法流程:

4.实现:

1) 输入起点

2) 根据起点用随机函数生成初始种群, 并计算路径长度选择概率

3) 根据选择概率用轮盘赌的原理按成对的方式选择相同数量的子代种群

4) 根据交叉概率选择子代中的部分进行交叉操作

5) 根据突变概率选择部分子代进行编译操作

6) 输出当代的种群信息和当前最优, 如果代数满足终止条件退出

5.细节:

1)所有

2)在进行两个个体的交叉操作之时,可能会产生冲突, 此时要根据冲突的序列相互在原路径中枚举修改

void GA\_Cross(int Father\_Index, int Mother\_Index)

{

int l = rand() % (CITY\_NUM - 1) + 1;

int r = rand() % (CITY\_NUM - 1) + 1;

if(l > r) Swap(l, r);

TSP\_Solution &father = Son\_Solution[Father\_Index];

TSP\_Solution &mother = Son\_Solution[Mother\_Index];

int Cross\_Length = r - l + 1, Conflict\_Length;

for(int i = l; i <= r; ++i)

{

Father\_Cross[i - l] = father.Path[i];

Mother\_Cross[i - l] = mother.Path[i];

}

int \* f\_conflict = getConflict(Father\_Cross, Mother\_Cross, Cross\_Length, Conflict\_Length);

int \* m\_conflict = getConflict(Mother\_Cross, Father\_Cross, Cross\_Length, Conflict\_Length);

for(int i = 0; i < Conflict\_Length; ++i)

{

for(int j = 1; j < CITY\_NUM; ++j)

if(father.Path[j] == m\_conflict[i])

{

father.Path[j] = f\_conflict[i];

break;

}

for(int j = 1; j < CITY\_NUM; ++j)

if(mother.Path[j] == f\_conflict[i])

{

mother.Path[j] = m\_conflict[i];

break;

}

}

for(int i = l; i <= r; ++i)

Swap(father.Path[i], mother.Path[i]);

}

【实验结果】





