%% 变异操作

% 函数说明

% 输入变量：pop：种群，pm：变异概率

% 输出变量：newpop变异以后的种群

function [new\_pop] = mutation(pop, pm, Grid, x)

[px, ~] = size(pop);

new\_pop = {};

%对每一行选择是否变异

for i = 1:px

% 初始化最大迭代次数

max\_iteration = 0;

single\_new\_pop = pop{i, 1};

[~, m] = size(single\_new\_pop);

% single\_new\_pop\_slice初始化

single\_new\_pop\_slice = [];

if(rand < pm)

while isempty(single\_new\_pop\_slice)

% 生成2到（m-1）的两个随机数,并排序

mpoint = sort(round(rand(1,2)\*(m-3)) + [2 2]);

%切除掉包含两个随机数在内的之间的路径节点，将切除部分及前后两个节点取出

single\_new\_pop\_slice = [single\_new\_pop(mpoint(1, 1)-1) single\_new\_pop(mpoint(1, 2)+1)];

%将取出的用于切除的部分路径重新联结成无间断路径(这一步可能变异 也可能不变异)

single\_new\_pop\_slice = generate\_continuous\_path(single\_new\_pop\_slice, Grid, x);

%max\_iteration = max\_iteration + 1;

if max\_iteration >= 100000

break

end

end

if max\_iteration >= 100000

new\_pop{i, 1} = pop{i, 1};

else

%将变异后的路径保存

new\_pop{i, 1} = [single\_new\_pop(1, 1:mpoint(1, 1)-1), single\_new\_pop\_slice(2:end-1), single\_new\_pop(1, mpoint(1, 2)+1:m)];

end

% single\_new\_pop\_slice再次初始化

single\_new\_pop\_slice = [];

else%不变异

new\_pop{i, 1} = pop{i, 1};

end

end