%% 计算路径长度函数

function [path\_value] = cal\_path\_value(pop, x)

[n, ~] = size(pop);

path\_value = zeros(1, n);

%循环计算每一条路径的长度

for i = 1 : n

single\_pop = pop{i, 1};

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

%路径有m个栅格，需要计算m-1次

for j = 1 : m - 1

% 点i所在列（从左到右编号1.2.3...）

x\_now = mod(single\_pop(1, j), x) + 1;

% 点i所在行（从上到下编号行1.2.3...）

y\_now = fix(single\_pop(1, j) / x) + 1;

% 点i+1所在列、行

x\_next = mod(single\_pop(1, j + 1), x) + 1;

y\_next = fix(single\_pop(1, j + 1) / x) + 1;

%如果相邻两个栅格为上下或左右，路径长度加1，否则为对角线，长度加根号2

if abs(x\_now - x\_next) + abs(y\_now - y\_next) == 1

path\_value(1, i) = path\_value(1, i) + 1;

else

path\_value(1, i) = path\_value(1, i) + sqrt(2);

end

end

end