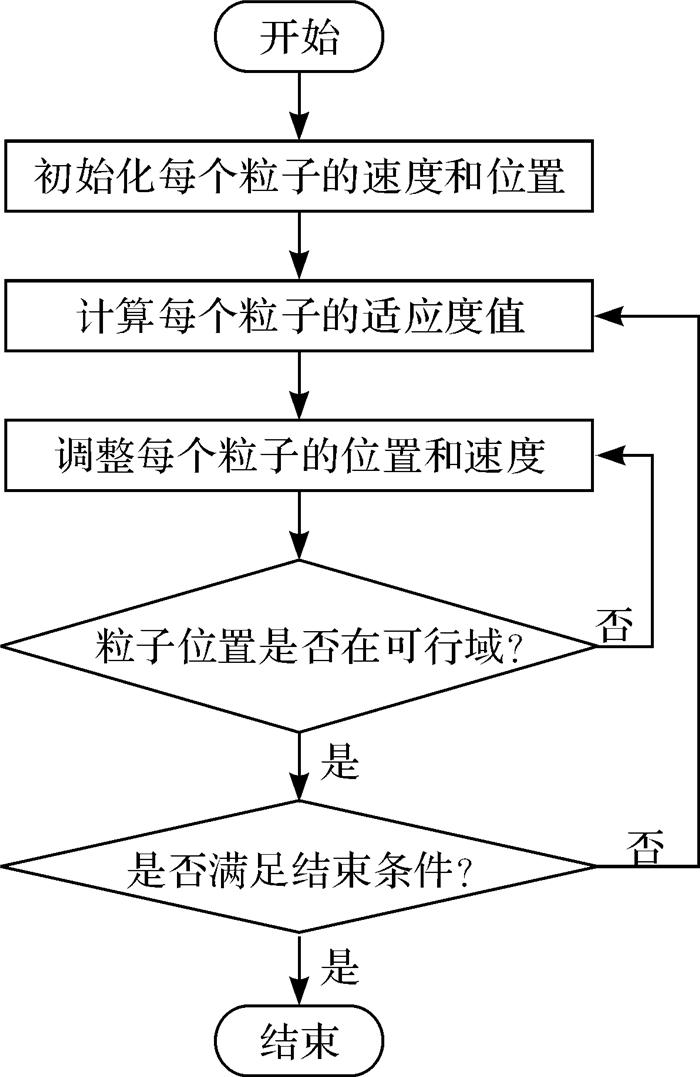
粒子群算法

## 算法思想

粒子群是一种全局搜索算法，同时他也是一种模拟自然界的生物活动以及群体智能的随机搜索算法。粒子群优化算法一方面吸取了人工生命、鸟群觅食、鱼群学习和群理论的思想，另一方面又具有进化算法的特点，和遗传算法，进化策略，进化规划等算法有相似的搜索和优化能力。其整体思想可以仿照鸟群理解。

有一群分散的鸟在随机的飞行觅食，每个鸟都不知道食物的具体位置，但有一个方法可以让他们知道自己距离食物的位置，各个小鸟就会在飞行过程中不断地记录和更新自己曾经到达过得距离食物最近的距离，并向鸟群这个群体通告，鸟群在接受通告后会更新自己的全局的最近距离和位置，并将这个位置和距离通告给所有的鸟，并以此影响每只鸟的行动。

## 算法设计



## 代码设计

[Gbest\_fitness,Gbest]=find\_best(ws,we,genmax,Pbest,Gbest,pp,c1,c2,m,n,w,v,fitness,Pbest\_fitness,Gbest\_fitness,city\_distance);

find\_best 函数如下：

function [Gbest\_fitness,Gbest] = find\_best(ws,we,genmax,Pbest,Gbest,pp,c1,c2,m,n,w,v,fitness,Pbest\_fitness,Gbest\_fitness,city\_distance)

gen=1;

while gen<genmax

gen=gen+1;

w = ws - (ws-we)\*(gen/genmax)^2;

change1=position\_minus\_position(Pbest,pp);

change1=constant\_times\_velocity(c1,change1);

change2=position\_minus\_position(repmat(Gbest(gen-1,:),m,1),pp);

change2=constant\_times\_velocity(c2,change2);

v=constant\_times\_velocity(w,v);

for i=1:m

for j=1:n

if change1(i,j)~=0

v(i,j)=change1(i,j);

end

if change2(i,j)~=0

v(i,j)=change2(i,j);

end

end

end

pp=position\_plus\_velocity(pp,v);

fitness=zeros(m,1);

for i=1:m

for j=1:n-1

fitness(i)=fitness(i) + city\_distance(pp(i,j),pp(i,j+1));

end

fitness(i)=fitness(i) + city\_distance(pp(i,end),pp(i,1));

end

for i=1:m

if fitness(i)<Pbest\_fitness(i)

Pbest\_fitness(i)=fitness(i);

Pbest(i,:)=pp(i,:);

end

end

[minvalue,min\_index]=min(fitness);

if minvalue<Gbest\_fitness(gen-1)

Gbest\_fitness(gen)=minvalue;

Gbest(gen,:)=pp(min\_index,:);

else

Gbest\_fitness(gen)=Gbest\_fitness(gen-1);

Gbest(gen,:)=Gbest(gen-1,:);

end

Length\_ave(gen)=mean(fitness);

end

end

## 源代码

见附件

## 实验测试

个体学习因子 0.1

社会学习因子 0.075

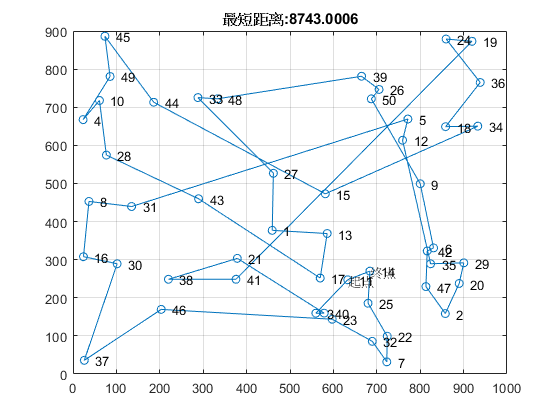
惯性因子 1

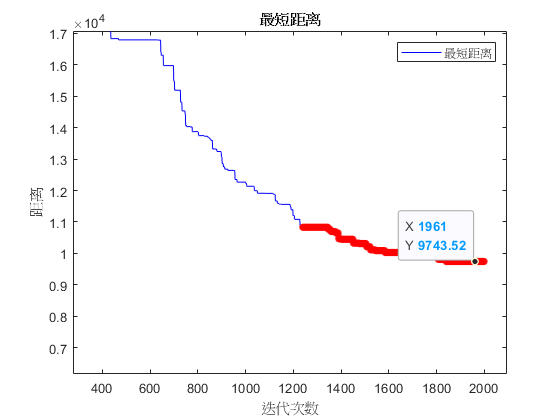
粒子数量 500

迭代次数 2000

惯性因子最大值 1

惯性因子最小值 0.8





## 结论

初始实验结果路径并不完美，还是存在了路线交叉的问题，而路线交叉就意味着没有找到最优解，但可以看出整个城市路径结构完整，可能是一个局部最优解。在经过参数调整之后，发现以我的能力，结果无法更好了，只能接受这个局部最优解，通过观察路线规划图可以得到，对于大部分路径来说这个都还是可以接受的。而且当参数设置为c1 = 1,c2=3时，算法不收敛，wM=1,wm=0.5时也不收敛，粒子群数目越多计算效率越低，但找到的解效果会变好。