生鲜农产品冷链物流配送路径优化模型构建及算法实现

且行且安~

已于 2023-06-20 23:28:19 修改

阅读量3k

收藏 79

点赞数 5

分类专栏： # VRP路径规划进阶之路 文章标签： 生鲜冷链 路径优化 蚁群算法 改进蚁群算法 单目标模型

版权

VRP路径规划进阶之路

专栏收录该内容

3 篇文章4 订阅

订阅专栏

摘要：本案例讲述的案例为生鲜农产品冷链物流配送路径优化，涉及的目标函数成本包括碳排放成本、固定成本、运输成本、货损变质成本、时间惩罚成本。

目标种类：单目标模型。

求解方法：基础版蚁群算法+改进版蚁群算法。

整体对标层级：硕士

目录

1、问题描述及假设条件

1.1 问题描述

1.2 假设条件

2、参数说明及模型成本构成

2.1 成本构成

3、模型构建

4、蚁群算法

5、蚁群算法改进

5.1 转移规则改进

5.2 信息素更新策略的改进

6 、源码与结果部分

6.1 源码

6.2 结果部分

7、总结展望

1、问题描述及假设条件

1.1 问题描述

本文主要研究生鲜农产品冷链物流车辆配送环节运输路径的优化问题，可表述为：生鲜农产品配送中心以冷链运输车辆为工具，在已知一定数量的客户，客户需求量及位置坐标情况下，尽可能满足客户需求，为每个客户提供生鲜农产品配送服务，使运输成本最小、耗费时间最小。在农产品冷链配送环节涉及到的要素包含配送中心、客户、货物、车辆、运输路径。

1.2 假设条件

根据上述描述，结合实际情况，对配送模型的建立做如下假设前提：

假设1，已知客户需求量，有一个配送中心仓库，且该仓库能够满足各客户点需求。

假设2，配送中心车辆车型唯一，且都只能从配送中心仓库出发。

假设3，每辆车匀速行驶且不能超载运输。

假设4，配送道路通畅，不考虑交通拥挤状况

假设5，每个客户对送货时间有明确要求，惩罚系数提前做好规定。

假设6，所有客户所需货物量不大于配送车辆的最大承载量。

假设7，每个客户点的货物仅由一辆车配送完成，一辆车可服务多个客户点

2、参数说明及模型成本构成

说明：由于公式中带有很多字母参数表达，本文中部分函数表达式就直接采用图片的方式。

2.1 成本构成

模型中考虑的生鲜农产品冷链物流配送过程中的成本包含：碳排放成本；配送车辆固定成本；配送车辆运输成本；配送过程中生鲜农产品变质成本；违反时间窗惩罚成本。具体成本构成如图所示。

①碳排放成本

现实中其碳排放量受到载重车辆类型、载重负荷、车辆的行驶速度和行驶距离等影响。这里研究中已假定车辆类型一致且均匀速行驶，因此主要考虑载重负荷和行驶距离的影响，相关参数及公式如下：

②固定成本

③运输成本

④货损变质成本

生鲜农产品在运输过程中需要一定的温度保鲜，货物储存条件较为严格。运输时间越久，温度越高都会导致货物变质，综合考虑运输及卸货环节的货损成本，引入生鲜农产品腐败函数

,（其中，表示这批生鲜农产品的初始质量，表示运输过程中的农产品腐败率，表示卸货过程中的农产品腐败率）。C1表示运输过程中农产品的变质成本，C2表示在到达客户点后由于卸货开启车门使得冷空气流失，造成车厢内温度升高从而引起的货物变质成本，相关参数及公式如下：

综上，可得整个配送过程中的货损成本

⑤时间窗成本

时间窗成本相对较为好理解。

3、模型构建

基于上述相关参数描述、公式设定以及总成本最低的目标，建立生鲜农产品低碳物流配送路径优化的数学模型如下：

4、蚁群算法

基础版本的蚁群算法讲解已在智能优化算法系列中讲述到

可跳转查阅【智能优化算法之蚁群算法】

5、蚁群算法改进

5.1 转移规则改进

考虑到基本的蚁群算法随机性搜索易导致局部最优出现停滞现象，采用确定性搜索和随机性搜索相结合的策略，改进后的转移规则如下，蚂蚁k按照式2选择下一个节点j：

其中，g是区间[0，1]内的一个随机数，为假定的固定阀值，在区间[0，1]内。若，采用确定搜索算法，用信息素和启发式因子影响因素乘积选择下一节点；若，按照式2根据转移概率选择下一节点。

5.2 信息素更新策略的改进

基本的蚁群算法在计算中容易陷入搜索停滞和过早收敛于非全局最优解的情况，因此改进信息素更新策略，初始化信息素值，设为最大值。考虑将每条边的信息素浓度限制在[，]之间。在t+n时刻，只有找到最短路径的蚂蚁才能在其经过的路径上释放信息素，路径（i，j）上信息调整如下：

6 源码与结果部分

6.1 源码

clear

clc

close all

tic

%% 用importdata这个函数来读取文件

c101=importdata('data.txt');

cap=300; %车辆最大装载量

%% 提取数据信息

E=c101(1,5); %配送中心时间窗开始时间

L=c101(1,6); %配送中心时间窗结束时间

vertexs=c101(:,2:3); %所有点的坐标x和y

customer=vertexs(2:end,:); %顾客坐标

cusnum=size(customer,1); %顾客数

v\_num=10; %车辆最多使用数目

demands=c101(2:end,4); %需求量

a=c101(2:end,5); %顾客时间窗开始时间[a[i],b[i]]

b=c101(2:end,6); %顾客时间窗结束时间[a[i],b[i]]

width=b-a; %顾客的时间窗宽度

s=c101(2:end,7);

speed = 35;

dist = zeros(cusnum+1,cusnum+1) ; %距离矩阵

for i =1:cusnum+1

for j =1:cusnum+1

if i==j

dist(i,j) =0;

else

dist(i,j)=cal\_distanceFun(vertexs(i,1),vertexs(i,2),vertexs(j,1),vertexs(j,2));

end

end

end

%% 初始化参数

m=50; %蚂蚁数量

alpha=1; %信息素重要程度因子

beta=3; %启发函数重要程度因子

gama=2; %等待时间重要程度因子

delta=3; %时间窗跨度重要程度因子

r0=0.5; %r0为用来控制转移规则的参数

rho=0.85; %信息素挥发因子

Q=5; %更新信息素浓度的常数

Eta=1./dist; %启发函数

Tau=ones(cusnum+1,cusnum+1); %信息素矩阵

Table=zeros(m,cusnum); %路径记录表

iter=1; %迭代次数初值

iter\_max=300; %最大迭代次数

Route\_best=zeros(iter\_max,cusnum); %各代最佳路径

c0 = 0.3; %单位距离碳排放税收价格

eta0 = 2.61; %单位距离碳排放系数

theta0 = 1; %空载时单位距离燃料消耗量

theta\_start = 2; %满载时单位距离燃料消耗量

p2 = 10; %生鲜农产品单位价值

zeta1 = 0.002; %运输过程中农产品腐败率

zeta2 = 0.003; %卸货过程中农产品腐败率

u1 = 10; %早于期望时间到达成本

u2 = 30; %晚于期望时间到达成本

Cost\_best=zeros(iter\_max,1);

%% 迭代寻找最佳路径

while iter<=iter\_max

%逐个蚂蚁选择

for i=1:m

%逐个顾客选择

for j=1:cusnum

np=next\_point(i,Table,Tau,Eta,alpha,beta,gama,delta,a,b,width,s,L,dist,cap,demands);

Table(i,j)=np;

end

end

%% 计算各个蚂蚁的成本

cost=zeros(m,1);

NV=zeros(m,1);

TD=zeros(m,1);

for i=1:m

VC=decode(Table(i,:),cap,demands,a,b,L,s,dist);

[carbon\_cost]=carbancostFun(VC,dist,demands,c0,eta0,theta0,theta\_start,cap);

[cost\_oil,nv,TD]=cost\_oilFun(VC,dist);

[total\_lose\_cost]=lose\_costFun(VC,dist,demands,p2,zeta1,zeta2,s);

[time\_cost]=timecostFun(VC,dist,a,b,s,u1,u2);

cost(i,1) = carbon\_cost+cost\_oil+total\_lose\_cost+time\_cost;

NV(i,1) = nv;

TD(i,1) = TD;

end

%% 计算最小成本及平均成本

if iter == 1

[min\_Cost,min\_index]=min(cost);

Cost\_best(iter)=min\_Cost;

Route\_best(iter,:)=Table(min\_index,:);

else

[min\_Cost,min\_index]=min(cost);

Cost\_best(iter)=min(Cost\_best(iter - 1),min\_Cost);

if Cost\_best(iter)==min\_Cost

Route\_best(iter,:)=Table(min\_index,:);

else

Route\_best(iter,:)=Route\_best((iter-1),:);

end

end

%% 更新信息素

bestR=Route\_best(iter,:);

[bestVC,bestNV,bestTD]=decode(bestR,cap,demands,a,b,L,s,dist);

% Tau=updateTau(Tau,bestR,rho,Q,cap,demands,a,b,L,s,dist);

Tau=updateTau(Tau,Table,rho,Q,cap,demands,a,b,L,s,dist);

%% 打印当前最优解

disp(['第',num2str(iter),'代最优解:'])

disp(['车辆使用数目：',num2str(bestNV),'，车辆行驶总距离：',num2str(bestTD)]);

fprintf('\n')

%% 迭代次数加1，清空路径记录表

iter=iter+1;

Table=zeros(m,cusnum);

end

%% 结果显示

bestRoute=Route\_best(end,:);

[bestVC,NV,TD]=decode(bestRoute,cap,demands,a,b,L,s,dist);

draw\_Best(bestVC,vertexs);

for i =1:length(vertexs)

text(vertexs(i,1),vertexs(i,2),num2str(i-1));

end

%% 绘图

figure(2)

plot(1:iter\_max,Cost\_best,'b')

xlabel('迭代次数')

ylabel('成本')

title('各代最小成本变化趋势图')

toc

6.2 结果部分

1）基础版的运行结果

2）改进版的运行结果

7、总结展望

这里介绍了基本蚁群算法的原理、运算公式以及运算流程，针对基本蚁群算法的缺陷提出了改进的蚁群算法，详细阐述了改进蚁群算法运算流程，在此基础上进行算例分析，以并进行算例仿真，验证了该改进蚁群算法在进行路径规划求解时效果优于基本蚁群算法。

————————————————

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

原文链接：https://blog.csdn.net/qq\_20412595/article/details/130952886