# 物资配送程序用户手册

## 程序用途

本程序用于在运输能力或者供给受限的情况下给需求点配送物资，并且优先装权重高的物资，走的路线是让需求点等待最短的路线。

程序需要满足，车辆的总载重（或者总体积）是小于等于或者略大于总分配量的重量（或者体积），假如车辆的总载重（或者总体积）远大于总分配量的重量（或者体积），则不属于程序考虑的范围，程序不会进行后续的分配方案，退出程序。比如**下面的规格**就**不符合**：

“

总需求点重量： 968.04 总需求点体积： 12.9336

总供应量重量： 836.8 总供应量体积： 10.368

总分配量重量： 825.6 总分配量体积： 10.276

方案 1 ：总车辆重量： 5240.0 总车辆体积： 33.3

方案 2 ：总车辆重量： 8040.0 总车辆体积： 51.3

“

## 程序使用方法

该程序会用到Excel文件，使用环境为Windows OS。

### 确保安装了必要软件

* 确保安装了Pthon3

**若没有安装Python3则请先安装最新版本Python3（https://www.python.org/downloads/windows/）**

* 确保安装了python3的**xlrd包**

需要自己单独安装xlrd包（若以前没有安装的话），该包用于读取Excel文件，**在运行程序前请务必先安装上xlrd包，否则程序会报错**。安装命令为：在windows的**cmd中键入pip install xlrd**。

准备工作需要完成上面两步。

### 在excel表中输入数据

Excel表中数据如下：

#### 需求点数据

需求点1 单位重量（kg） 单位体积（m3） 需求量（件） 配送量（件） 权重

物资1 0.008 0.00017 1770 1.21

物资2 0.28 0.0023 390 1

物资3 0.083 0.0015 940 1.05

物资4 0.5 0.005 1200 1.03

物资5 0.1 0.001 600 1.08

需求点2 单位重量（kg） 单位体积（m3） 需求量（件） 配送量（件） 权重

物资1 0.008 0.00017 1120 1.01

物资2 0.28 0.0023 190 1.2

物资3 0.083 0.0015 760 1

物资4 0.5 0.005 1180 1.04

物资5 0.1 0.001 200 1.8

#### 车辆组合方案数据

车辆类型 车型A 车型B 车型C 车型D 车型E

载重量（kg） 1800 480 1000 500 500

载重体积限制（m3） 13.5 3.6 4.5 2.6 3

辆数 1 3 2 2 2

车辆类型1 车型A 车型B 车型C 车型D 车型E

载重量（kg） 1800 480 1000 500 500

载重体积限制（m3） 13.5 3.6 4.5 2.6 3

辆数 2 3 3 2 2

#### 供应量数据

供应量 物资1 物资2 物资3 物资4 物资5

件数 8400 1800 3200 8860 3500

#### 需求点距离数据（0点代表配送中心）

距离 0 1 2 3 4 5 6 7 8

0 0 40 60 75 90 200 100 160 80

1 40 0 65 40 100 100 75 110 100

2 60 65 0 75 100 100 75 75 75

3 75 40 75 0 100 50 90 90 150

4 90 100 100 100 0 100 75 75 100

5 200 100 100 50 100 0 70 90 75

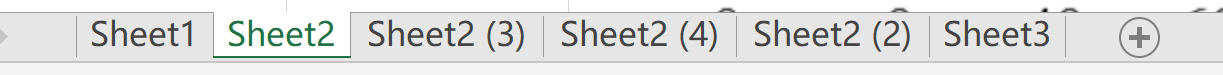
6 100 75 75 90 75 70 0 70 100

7 160 110 75 90 75 90 70 0 100

8 80 100 75 150 100 100 100 100 0

### 3.在Python shell的窗口中查看文字输出和图像输出

#### 输入数据

在输入好Excel中的数据之后，保存文件。Python代码只会读取第二张表中的数据，

你可以多准备几张表，把需要分配的表移到第二张表的位置保存即可。

#### 运行程序

用Python shell或者其他工具**运行xunzhaoluxiannew.py**文件即可。结果会输出寻找路线的等待时间图线，以及从读取数据到最终分配方案的全流程文字数据。

需要注意在第二步：寻找路线前有如下提示：

!!!!!!!!!! 请注意:观察最短总等待时间，不合适则重新运行程序 !!!!!!!!!!

假如觉得给出的最短时间不佳，则重新运行程序试一试。某些时候蚁群算法并没有找出最优的路线，可能只是次优路线。

##### 手动设置参数

**xunzhaoluxiannew.py**文件末尾的“手动设置参数区域”可以自己设置参数，包括路线相关参数以及蚂蚁匹配相关参数。路线相关参数主要控制的是蚁群算法中的参数，比如迭代次数，信息素浓度权重、期望权重、蚂蚁继续访问概率、信息素挥发率等。蚂蚁匹配参数也就是公平系数，为了更好的蚂蚁匹配，可以自己设置，也可不用设置，程序自动给出。

#########手动设置参数区域#######################

###路线相关参数----------------------------

###最大迭代次数N

N=8000

###alpha:信息素浓度Taoij的权值

alpha=6

###belta：期望Eataij的权值

belta=1

###每只蚂蚁继续访问的概率

Padd=0.2

###信息素挥发率

rou=0.

###----------------------------

###蚂蚁匹配相关参数----------------------------

####设置公平系数

gongpingxishu=0.7\*np.ones(len(Zclzl),dtype=float)##初始化的公平系数都为0.7

#########手动设置参数区域#########################

#### 结果解读

用户应该略去查看前面的信息直接跳到第五步的信息，这一步包含了汇总信息以及具体方案信息。

##### 汇总信息

汇总信息里面都是总体的统计信息，需要注意的地方已经加粗，如下：

====================================================================================================

\*\*\*\*\*\*\*\*\*\* 第五步：最终分配（开始） \*\*\*\*\*\*\*\*\*\*

--------------------------------------------------

\*\*\*\*\* 汇总信息 \*\*\*\*\*

总需求点数： 8

方案 1 的总车辆数： 10

方案 2 的总车辆数： 12

总需求点重量： 6741.04 总需求点体积： 70.6636

总供应量重量： 5616.8 总供应量体积： 58.168

总分配量重量： 5583.6 总分配量体积： 57.856

方案 1 ：总车辆重量： 7240.0 总车辆体积： 44.5

方案 1 ：总配送重量： 4389.368 总配送体积： 44.4929

方案 2 ：总车辆重量： 10040.0 总车辆体积： 62.5

方案 2 ：总配送重量： 5414.4 总配送体积： 56.359

需求点物资1到物资 5 的需求量

[11680, 1760, 4600, 10890, 3280]

配送中心物资1到物资 5 的供应量

[8400, 1800, 3200, 8860, 3500]

进行分配时物资1到物资 5 的可分配量

[8400, 1760, 3200, 8860, 3280]

**按方案 1 分配,物资1到物资 5 的实际分配量**

**[4470, 780, 876, 7669, 2280]**

**按方案 1 分配后,物资1到物资 5 的剩余可分配量**

**[3930, 980, 2324, 1191, 1000]**

**按方案 2 分配,物资1到物资 5 的实际分配量**

**[8400, 1370, 3200, 8860, 2680]**

**按方案 2 分配后,物资1到物资 5 的剩余可分配量**

**[0, 390, 0, 0, 600]**

--------------------------------------------------

实际分配量就是该方案运送的数据，剩余可分配量就是下一次运输可以继续分配的量。

##### 具体方案信息

具体方案信息就是物资怎么装载怎么运的问题。包括三个信息，分配矩阵，装车信息，需求点矩阵。

###### 分配矩阵

用于总览配送过程，比如某个需求点每种物资配送多少件，体积重量多少，匹配蚂蚁用到的重量阈值和体积阈值、哪一只蚂蚁，什么类型的蚂蚁、需求点等待多久等。例子如下：

按方案 1 分配,分配矩阵为：

下表标题：1物资1配送数量 ，2物资2配送数量 ，3物资配送3数量 ，4物资4配送数量 ，5配送重量 ，6配送体积 ，7重量阈值 ，8体积阈值 ，9蚂蚁序号 ，10蚂蚁类型号 ，11路线序号 ，12等待时间(min) ，13满意度(暂未定义)

[[1770. 0. 876. 457. 600. 375.368 4.4999

654.974 4.077 1. 3. 1. 40. 0. ]

[ 0. 190. 0. 1180. 200. 663.2 6.537

547.764 3.427 2. 41. 2. 60. 0. ]

[1190. 170. 0. 931. 300. 552.62 5.5483

538.153 3.348 3. 13. 3. 75. 0. ]

[ 0. 0. 0. 1380. 300. 720. 7.2

900.844 5.443 4. 10. 4. 90. 0. ]

[1510. 170. 0. 384. 80. 259.68 2.6477

403.962 2.574 3. 13. 3. 125. 0. ]

[ 0. 0. 0. 1320. 0. 660. 6.6

774.855 4.696 5. 8. 5. 100. 0. ]

[ 0. 250. 0. 1277. 0. 708.5 6.96

767.274 4.681 2. 41. 2. 135. 0. ]

[ 0. 0. 0. 740. 800. 450. 4.5

478.254 2.889 6. 3. 6. 80. 0. ]]

###### 装车信息

装车信息中可以读出每辆车走哪条路线，装多少物资，剩余多少空间，例子如下;

按方案 1 分配,装车情况为：

蚂蚁序号或路线号 1 按顺序经过 [1] 点，车辆的空间剩余情况和物资装载情况：

车型C 剩余重量： 625.0 剩余体积： 0.0 运送物资个数： [1770, 0, 876, 457, 600]

蚂蚁序号或路线号 2 按顺序经过 [2, 7] 点，车辆的空间剩余情况和物资装载情况：

车型A 剩余重量： 428.0 剩余体积： 0.0 运送物资个数： [0, 440, 0, 2457, 200]

蚂蚁序号或路线号 3 按顺序经过 [3, 5] 点，车辆的空间剩余情况和物资装载情况：

车型D 剩余重量： 239.0 剩余体积： 0.0 运送物资个数： [876, 170, 0, 352, 300]

车型D 剩余重量： 242.0 剩余体积： 0.0 运送物资个数： [1111, 170, 0, 388, 80]

车型E 剩余重量： 207.0 剩余体积： 0.0 运送物资个数： [713, 0, 0, 575, 0]

蚂蚁序号或路线号 4 按顺序经过 [4] 点，车辆的空间剩余情况和物资装载情况：

车型B 剩余重量： 120.0 剩余体积： 0.0 运送物资个数： [0, 0, 0, 660, 300]

车型B 剩余重量： 120.0 剩余体积： 0.0 运送物资个数： [0, 0, 0, 720, 0]

蚂蚁序号或路线号 5 按顺序经过 [6] 点，车辆的空间剩余情况和物资装载情况：

车型B 剩余重量： 120.0 剩余体积： 0.0 运送物资个数： [0, 0, 0, 720, 0]

车型E 剩余重量： 200.0 剩余体积： 0.0 运送物资个数： [0, 0, 0, 600, 0]

蚂蚁序号或路线号 6 按顺序经过 [8] 点，车辆的空间剩余情况和物资装载情况：

车型C 剩余重量： 550.0 剩余体积： 0.0 运送物资个数： [0, 0, 0, 740, 800]

###### 需求点矩阵

需求点矩阵中可以看出需求点需要的物资和实际得到的物资，以及物资的权重是怎样的，可以看作是配送中心给需求点的一张收据，在车辆卸货的时候，卸下配送数量这么多物资给需求点，完成配送任务。例子如下：

按方案 1 分配,需求点的需求矩阵为：

下表的标题为：单位重量，单位体积，需求个数，配送个数，物资权重

[[[ 0.008 0.00017 1770. 1770. 1.21 ]

[ 0.28 0.0023 390. 0. 1. ]

[ 0.083 0.0015 940. 876. 1.05 ]

[ 0.5 0.005 1200. 457. 1.03 ]

[ 0.1 0.001 600. 600. 1.08 ]]

[[ 0.008 0.00017 1120. 0. 1.01 ]

[ 0.28 0.0023 190. 190. 1.2 ]

[ 0.083 0.0015 760. 0. 1. ]

[ 0.5 0.005 1180. 1180. 1.04 ]

[ 0.1 0.001 200. 200. 1.8 ]]

[[ 0.008 0.00017 1190. 1190. 1.21 ]

[ 0.28 0.0023 170. 170. 1.3 ]

[ 0.083 0.0015 640. 0. 1.03 ]

[ 0.5 0.005 1160. 931. 1.05 ]

[ 0.1 0.001 300. 300. 1.2 ]]

[[ 0.008 0.00017 1290. 0. 1.02 ]

[ 0.28 0.0023 140. 0. 1.03 ]

[ 0.083 0.0015 370. 0. 1. ]

[ 0.5 0.005 2200. 1380. 1.2 ]

[ 0.1 0.001 300. 300. 1.3 ]]

[[ 0.008 0.00017 1510. 1510. 1.2 ]

[ 0.28 0.0023 170. 170. 1.1 ]

[ 0.083 0.0015 680. 0. 1.01 ]

[ 0.5 0.005 840. 384. 1.02 ]

[ 0.1 0.001 80. 80. 1.08 ]]

[[ 0.008 0.00017 1320. 0. 1.01 ]

[ 0.28 0.0023 180. 0. 1.01 ]

[ 0.083 0.0015 410. 0. 1. ]

[ 0.5 0.005 1800. 1320. 1.3 ]

[ 0.1 0.001 400. 0. 1.2 ]]

[[ 0.008 0.00017 2200. 0. 1.01 ]

[ 0.28 0.0023 250. 250. 1.02 ]

[ 0.083 0.0015 480. 0. 1.01 ]

[ 0.5 0.005 1660. 1277. 1.02 ]

[ 0.1 0.001 600. 0. 1.02 ]]

[[ 0.008 0.00017 1280. 0. 1.04 ]

[ 0.28 0.0023 270. 0. 1.03 ]

[ 0.083 0.0015 320. 0. 1.04 ]

[ 0.5 0.005 850. 740. 1.2 ]

[ 0.1 0.001 800. 800. 1.5 ]]]

###### 分配总结

根据这三张表的信息就可以形成配送计划了。

##### 出错

###### 提前退出

假如车辆的总载重（或者总体积）远大于总分配量的重量（或者体积），则不属于程序考虑的范围，即不属于剩余空间问题，程序退出，需要调整数据到合适范围。例如程序提前，可以在第一步：读取数据中看到如下信息：

总需求点重量： 968.04 总需求点体积： 12.9336

总供应量重量： 836.8 总供应量体积： 10.368

总分配量重量： 825.6 总分配量体积： 10.276

方案 1 ：总车辆重量： 5240.0 总车辆体积： 33.3

方案 2 ：总车辆重量： 8040.0 总车辆体积： 51.3

证明了我们的车辆的运输能力是远远大于了分配量，我们要么减小车辆运输能力（减少车辆），要么提高分配量（增加物资数量或者种类）。

在修改Excel数据后，再次[运行程序](#_运行程序)即可。

###### 报错

假如出现了File not Found之类的提示很有可能没有安装必要包，请先安装[**xlrd包**](#_确保安装了必要软件)**。**

## 反馈

更多程序问题，请在<https://github.com/truehang/wuzifenpei/issues>反馈，谢谢合作。