# 数据集说明

本文档以example.txt数据集为例，来说明数据集的要求，含义，计算说明等。

example.txt是一个测试数据集。其中的数据均为随机生成的。其特点是数据量比较小，仅有7个结点，7条边。example.txt就在与此文件同级的文件夹中。

目录

[数据集说明 1](#_Toc95127099)

[1 基本要求 2](#_Toc95127100)

[2 各字段含义及要求 3](#_Toc95127101)

[3 数组类型字段的详细要求 4](#_Toc95127102)

[3.1 NODE 4](#_Toc95127103)

[3.2 EDGE 4](#_Toc95127104)

[3.3 DEMAND 4](#_Toc95127105)

[4 RISK\_MATRIX的计算方法 5](#_Toc95127106)

[5 数据一致性检查 6](#_Toc95127107)

[6 关于MOPSO算法中的邻接矩阵 6](#_Toc95127108)

[7 待填充的空白数据集表 8](#_Toc95127109)

## 1 基本要求

* 数据集由一个个字段构成，每个字段必须由三部分（如图1.1）构成。

A picture containing table

Description automatically generated

图1.1 example.txt中的第一个字段

* 字段名是由大写英文字母和下划线构成
* 字段值如果不能在一行写完，就需要换行，使得字段名与字段值不出现在同一行，如RISK\_MATRIX字段
* 字段值可能由多个数字构成，此时每个数字间要么用空格分隔，要么用换行分隔
* 若无明确指出，单位默认使用：千克、千米、元、千米/小时等

## 2 各字段含义及要求

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 要求 |
| DATASET | 字符串 | 数据集名称，使用一个简单的英文单词即可，与文件名同名 |
| NODE\_COUNT | 正整数 | 包含配送中心在内结点总数 |
| NODE | 数组 | 所有节点的编号及坐标值，坐标允许小数值。第一行是配送中心，编号为0；后面是剩下需求点的坐标，从1开始编号 |
| EDGE\_COUNT | 正整数 | 边的数量 |
| EDGE | 数组 | 边集，接下来每行用起始节点编号和结束节点编号代表一条边，起始编号＜结束编号 |
| VEHICLE\_COUNT | 正整数 | 车辆数量 |
| VEHICLE\_CAPACITY | 正数 | 车辆载重，需大于任意单一需求点需求量(份数) |
| VEHICLE\_SHIPPING\_COST | 正数 | 车辆单位距离运输成本(元/km) |
| VEHICLE\_FIXED\_COST | 正数 | 车辆的固定成本(元/辆) |
| VEHICLE\_DISINFECTION\_COST | 正数 | 车辆的消毒成本(元/辆) |
| VEHICLE\_VELOCITY | 正数 | 车辆的行驶速度(km/h) |
| DEMAND | 数组 | 各需求点对套餐的需求量(份数)，需求点编号和上面相对应 |
| RISK\_MATRIX | 0-1矩阵 | 风险矩阵，包括配送原点到其他需求点的风险，是NODE\_COUNT×NODE\_COUNT大小的矩阵 |
| EOF | 特殊标记 | End of File标记，用于标记文件到头了，无需填写字段值 |

## 3 数组类型字段的详细要求

### 3.1 NODE

配送中心结点编号 横坐标 纵坐标

第一个需求点编号 横坐标 纵坐标

...

最后一个需求点编号 横坐标 纵坐标

### 3.2 EDGE

第一条边的起始节点编号 结束节点编号

...

最后一条边的起始节点编号 结束节点编号

### 3.3 DEMAND

1 编号为 1 的需求点对套餐的需求量，正数

2 编号为 2 的需求点对套餐的需求量，正数

...

最后一个需求点的编号 对应的套餐需求量，正数

## 4 RISK\_MATRIX的计算方法

在example.txt中，假设7个结点的风险程度分别是：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 风险程度 | 48 | 59 | 67 | 40 | 0 | 89 | 91 |

按照风险程度降序排列：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 6 | 5 | 2 | 1 | 0 | 3 | 4 |
| 风险程度 | 91 | 89 | 67 | 59 | 48 | 40 | 0 |

所以风险矩阵为：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

## 5 数据一致性检查

由于数据来自模拟环境，因此还需要对数据的正确性、一致性进行检查。下面是已知的检查内容：

* 风险矩阵：假如结点1风险大于结点2风险，结点2风险大于结点3风险，那么x12=1，但x21=0；x23=1，但x32=0；并且x13因该也是1 而不是0！总之要确保不会出现1大于2，2大于3，3又大于1这种衔尾蛇效应。
* 边集：检查会不会导致有的需求点根本没被联通到，如果使用已有的数据集，一般不会有这个问题；比如如果有10个结点，但边只有2个，显然是有问题的。
* 车辆载重：必须大于单个需求点的需求量，如此才能满足一个需求点有且仅有一辆车服务的约束条件
* 车辆数量：至少大于需求点个数，以满足一个需求点有且仅有一辆车服务的约束条件
* ……

还可能有其他一致性问题不能穷尽，需要在编制数据集的过程中仔细检查。

## 6 关于算法中的邻接矩阵

example.txt数据集中有7条边，通过欧几里得公式，算出两个结点间的欧氏距离，得出原始的邻接矩阵如表6.1。其中Inf是Infinite的缩写，表示两节点无直接边相连。

经过弗洛伊德算法（计算任意两点间的最短路径）计算后的邻接矩阵如表6.2。

表6.1 表6.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 0 | Inf | 145 | Inf | 223 | Inf | Inf |  | 0 | 0 | 346 | 145 | 196 | 223 | 471 | 351 |
| 1 | Inf | 0 | 201 | Inf | Inf | 125 | Inf |  | 1 | 346 | 0 | 201 | 252 | 569 | 125 | 272 |
| 2 | 145 | 201 | 0 | 51 | Inf | Inf | Inf |  | 2 | 145 | 201 | 0 | 51 | 368 | 326 | 206 |
| 3 | Inf | Inf | 51 | 0 | Inf | Inf | 155 |  | 3 | 196 | 252 | 51 | 0 | 419 | 302 | 155 |
| 4 | 223 | Inf | Inf | Inf | 0 | Inf | Inf |  | 4 | 223 | 569 | 368 | 419 | 0 | 694 | 575 |
| 5 | Inf | 125 | Inf | Inf | Inf | 0 | 147 |  | 5 | 471 | 125 | 326 | 302 | 694 | 0 | 147 |
| 6 | Inf | Inf | Inf | 155 | Inf | 147 | 0 |  | 6 | 351 | 272 | 206 | 155 | 575 | 147 | 0 |

## 7 待填充的空白数据集表

图7.1 空白数据集

DATASET:

NODE\_COUNT:

NODE:

EDGE\_COUNT:

EDGE:

VEHICLE\_COUNT:

VEHICLE\_CAPACITY:

VEHICLE\_SHIPPING\_COST:

VEHICLE\_FIXED\_COST:

VEHICLE\_DISINFECTION\_COST:

VEHICLE\_VELOCITY:

DEMAND:

RISK\_MATRIX:

EOF