

README

项目简介(for 所有人)

这是一个仓储物流优化算法，旨在利用优化算法来解决复杂的物流分配问题。给定一系列的订单需求、仓库存货、以及物流运输时间，本算法的目标是找到一个最优的分配计划，以使得满足订单需求的同时，最大化总体满意度并最小化最大的运输时间。

项目使用Python语言进行开发，依赖于numpy库进行矩阵计算。

项目结构(for 开发人员)

主目录:

- `README.md`: 项目的基本介绍和使用指南。
- `requirement.txt`: Python环境依赖文件。
- `doc`: 文档目录，包含项目的设计、需求、测试等文档。
- `src`: 源代码目录。

doc目录:

- `Software_Architecture_Document.md`: 软件架构文档。
- `bug.md`: 记录项目过程中出现的bug。
- `figure`: 项目的各种图片资料。
- `ppt`: 项目需求和设计的演示文稿。
- `prompt`: 各类提示信息和问题记录。
- `requirement.md`: 项目需求文档。
- `test.md`: 测试文档。
- `todo`: 待完成的任务记录。
- `version.md`: 项目的版本信息。

src目录:

- `__pycache__`: Python的缓存文件目录。
- `data.json`: 数据文件。
- `optimizer.py`: 优化算法的实现文件。是本项目的核心算法
- `test`: 测试文件目录。。
- `util.py`: 包含一些实用的函数，如从json文件中加载数据并转换为矩阵的函数。

核心代码(for 开发人员)

optimizer.py:

该文件是项目的核心，其中定义了一个名为 `logistics_distribution` 的函数，该函数的目标是根据输入的矩阵X，Y，Z，O和W（分别表示单位运输时间、各订单的商品需求、各仓库的商品存货以及订单和仓库的优先级），输出一个满足需求的分配计划，以最大化总体满意度并最小化最大的运输时间。

环境配置(for 运维人员)

首先，你需要确保你有以下文件：

- Miniconda 的安装包：`Miniconda3-latest-Linux-x86_64.sh`
- 打包好的 conda 环境：`wuliu.zip`
- 项目源代码：`wuliu-master.zip`

然后，按照以下步骤操作：

1. 将光盘中的 `Miniconda3-latest-Linux-x86_64.sh`、`wuliu.zip` 和 `wuliu-master.zip` 文件复制到服务器的一个目录中，例如 `/home/user/`。
2. 打开一个终端，然后运行以下命令来安装 Miniconda：

```
chmod +x /home/user/Miniconda3-latest-Linux-x86_64.sh  
/home/user/Miniconda3-latest-Linux-x86_64.sh
```

在安装过程中，你会被提示接受许可协议并选择安装位置。默认情况下，Miniconda 会安装在你的主文件夹中。

3. 安装完成后，关闭并重新打开你的终端，或者运行以下命令来更新你的 shell：

```
source ~/.bashrc
```

4. 解压你的 `wuliu.zip` 文件到 conda 的环境文件夹下：

```
unzip /home/user/wuliu.zip -d ~/miniconda3/envs/
```

5. 激活 `wuliu` 环境：

```
conda activate wuliu
```

现在，你应该可以在你的离线计算机上使用你的 `wuliu` 环境，其中包含了所有你需要的 Python 库。

下一步，解压并使用你的项目源代码：

6. 解压项目源代码：

```
unzip /home/user/wuliu-master.zip -d /home/user/
```

7. 导航到你的项目源代码的文件夹：

```
cd /home/user/wuliu-master
```

运行项目

1. 打开终端，激活项目所在的conda环境：

```
conda activate wuliu
```

2. 运行项目的主应用：

```
python app.py
```

这将在本地启动服务，端口为8080，API路径为/getZytpcl，如：

```
http://127.0.0.1:8080/getZytpcl
```

如果在服务器上运行，请将127.0.0.1替换为你的服务器地址。

测试项目

1. 复制test_api.py文件，然后在你的环境中运行这个复制的文件以进行测试。

注意：确保你已经安装了项目所需的所有依赖，可以通过在项目根目录下运行以下命令来安装：

```
pip install -r requirement.txt
```

2. 为了测试各种情况，我们设计了5组其它数据，覆盖了以下情况

```
data_1: 单订单单仓库，可满足  
data_2: 单订单多仓库，可满足  
data_3: 单订单多仓库，不可满足  
data_4: 多订单多仓库，可满足  
data_5: 多订单多仓库，不可满足
```

你可以在data文件夹中找到这些数据。为了测试这些数据，将data文件夹中的数据代替data.json文件，然后运行test_api.py文件。

如果遇到任何问题或需要进一步的帮助，请查看项目的详细文档或向我们的支持团队寻求帮助。