基于移动储能车的优化调度和策略

摘要

随着我国经济的快速发展，用电负荷持续飙升，在电网系统中接入可以灵活调节的储能设备去平抑功率波动。本文针对如何调度接入的储能设备的问题做出来详细的研究，基于网络流的模型和混合整数线性规划模型，考虑台区变压器的负荷需求，以投入的移动储能车获得的最大的收益最大为目标。以此为目标，建立约束方程，并使用 Yalmip 求解器求解。

针对问题一：通过对于各台区变负荷数据的分析，找到其中需要充电的时间点和区域，按照满足客户优先级进行统筹规划构建所有需要满足的约束条件，建立收益的目标函数最终得到最优的调度方案，收益为 551 元

针对问题二：在第一题的基础上，加入了光伏发电板也作为自发自用的设备，也用于为台区变压器降低负载率。我们在问题一的基础上，加入新的约束条件，得到合理的调配方案，收益为 611 元

针对问题三：在之前的基础上加入拥堵系数，考虑到了道路拥堵等不确定性对于调度的进一步影响，此时收益受到影响只有 445 元

关键词：移动储能车 调度优化 混合整数线性规划

目录

基于移动储能车的优化调度和策略	I
摘要	I
1 问题综述	1
1.1 问题背景	1
1.2 问题提出	1
1.3 资料条件	2
2 模型假设与符号说明	2
2.1 模型基本假设	2
2.2 符号说明	2
3 数据预处理	3
3.1.1 附件 1: 负荷、电价数据	3
3.1.2 附件 3: 29 个节点数据	3
4 问题分析与模型建立	3
4.1 问题分析	3
4.2 调度模型建立	3
4.2.1 问题一	3
4.2.2 问题二	6
4.2.3 问题三	7
5 模型评价与推广(或改进)	8
5.1 模型的优点	8
5.2 模型的不足	9
5.3 模型的推广【或改进】	9
参考文献	10
附 录	11
附录 I: 表示调度方案的矩阵	11
附录 II: 主要程序/关键代码	11

1 问题综述

1.1 问题背景

随着经济的高速发展,城市规模的不断扩大,用电负荷持续飙升。在用电的高峰期,人口密集的主城区中的变电所往往面临着供电容量不足的局面。从而产生电压波动、闪变、谐波等电能质量的问题。因此,我们需要考虑配电网的调配,在能力范围内快速地调整负荷,满足用户的用电需求,保证供电质量以及可靠性。

基于上述的需求,国家电网将在配电系统中接入可移动式的储能设备,即多功能移动储能车(Mobile Vehicle)。它具有灵活调节的特性,能够减轻台区变压器高峰时期的用负载率,缓解功率波动、削峰填谷。当有多台移动储能车,人们可以相互协调多台储能设备共同完成供电任务。这就需要人们通过分析不同时间段台区变压器的负荷变化为移动储能车制定运行模式,不断优化运行调度,从而在满足用电需求的同时实现移动储能设备在工作周期内的收益最大化。

目前不少国内外学者对储能系统控制策略已有研究,他们将这样对于移动储能车调度的问题归结为车辆路径问题。常见的研究思路是根据整个电网系统中台区变压器的位置信息、储能车的位置信息、载荷需求等约束去建立目标函数,然后利用混合整数规划、启发式算法、遗传算法、粒子群随机算法(PSO)等优化方法建立移动储能车的调度模型,追踪储能车在一段时间内的状态和位置,得到最佳的调度方案。

与已有研究不同,本文面对的条件约束多而复杂,应用场景更接近于实际生活。在此基础上,我们根据实际情况将不同用户的用电需求、储能车充放电的可行轨迹路线等更多实际需求结合起来,调用 YALMIP 求解器,得出使得收益最大化的储能车调配方案。

1.2 问题提出

本文的移动储能车的优化调度问题混合整数规划、图论最短路径等多方面的问题,往往需要考虑不同优先级的用户需求、储能车的电池容量、储能车的行驶状态等。具体地,需要满足下列点约束下:

- 状态约束:每辆车有仅仅有一个状态
- 负载约束:移动储能车为台区变充电使得负载率不超过 1
- 储能车电量约束:储能车电量在各个台区变压器处充满之后才能回到初始位置,一天调度结束后
- 状态转移约束:刚作为终点驶入的站点,下一时刻如果移动储能车要移动,必须走以该站点为起点的路径

需要从最短路径、电荷负载等多方面建立约束条件,解决以下三个问题:

- (1) 问题 1: 根据 24 小时分时电价的相关数据和电池额定功率/额定容量,合理的调配 10 辆移动储能车。按照客户优先级满足各台区变的负荷需求,结合通路网数据建立可行的移动调度方案的数学模型并求得解,最终使得移动储能车总充放电服务收益最大。
- (2) 问题 2: 在问题 1 的基础上,假设台区变节点 6、13、18、22 集成了光伏发电板,可以自发自用,直接通过自发的电弥补台区变在负荷高峰期的不足。此时,根据光伏发电板的每 15 分钟分布式光伏出力的预测数据,重新建立移动储能车的优化配置和调度模型并求解,使得在调动移动储能车最少的情况下,按照客户优先级满足各类用户的用电需求。

- (3) 问题 3: 在问题 1 的基础上, 我们发现移动储能车在实际的应用场景中存在着多辆储能车同时到达某一地点产生车辆拥堵的现象。此时就需要将交通路况的不确定性、充电站排队等候时间的不确定性考虑其中, 进一步制定调度方案的数学模型并求解。

1.3 资料条件

B 题的附件提供了相关的电负载和交通网系统的信息, 各文件的详细说明如下:

- 负荷、电价数据.xlsx 该文件提供了 24 小时中每个台区变的电负荷变化, 以及储能车在不同时间段的充放电的成本和收益。
- 光伏出力数据.xlsx 该文件详细提供了每 15 分钟光伏出电板的出力电量。
- 29 个节点数据.xlsx 该文件提供研究地区的道路交通网络, 以及每两个地点之间的距离。

2 模型假设与符号说明

2.1 模型基本假设

- (1) 假设移动储能车只在 4 个台区变和 5 个充电站共九个节点处停靠
- (2) 假设同种类型的储能车在每个时刻充电量和放电量都是相同的
- (3)
- (4)

2.2 符号说明

本文定义了如下 13 个使用次数较多的符号, 其余符号在使用时注明.

表1 符号说明

符号	含义	单位
i	储能车	无
J	充放电的地点	无
t	时间点	时
$P_i^c(t)$	车 i 在 t 时的充电量	KW
$P_i^d(t)$	车 i 在 t 时的放电量	KW
$x_{i,j}^c(t)$	车 i 是否放电的 0-1 变量	无
$C^c(t)$	t 时刻充电电价	元
$C^d(t)$	t 时刻放电电价	元
FC_i	车 i 的单公里成本	元
$X_{i,l}^m(t)$	表示车辆是否在路 l 上行驶的 0-1 变量	无
Di_l	路 l 的长度	KM
W_i	初始车电容量	KW
$W_j(t)$	t 时刻 j 处需要的下降的功率	KW

3 数据预处理

3.1.1 附件 1：负荷、电价数据

我们对附件 1 进行处理，根据其中每个时刻的实际运行功率和额定功率的比值，求出每个台区变压器在每个时刻的负载率。经过这样的处理和操作，我们可以直观地筛选出有哪些台区变压器在哪些时刻需要进行充电处理，使其负载率降为 1 以下。

进一步，我们又计算了每一台台区变压器想要使得负载率降为 1 以下需要的最少充电量。

3.1.2 附件 3：29 个节点数据

我们将附件 3 中的表示交通网系统的 29 个节点数据，首先导入成为一个表示距离的权重矩阵。为了节省成本使得收益最大，储能汽车在满足各种约束下，优先选择走两点之间路径最短的路程。因此，我们采用迪杰斯特拉算法（Dijkstra 算法）由代表距离的权重矩阵求得从道路网络中的一个地点出发去另一个地点距离最小的路径。

4 问题分析与模型建立

4.1 问题分析

题目以电网调度为背景，介绍了移动储能车在提高供电质的重要作用，问题一要求我们利用储能车将各个地区的台区变负载率超过 1 的地方降到稳定线以下，保证供电质量满足客户需求。

问题二要求我们在问题一的基础上进一步考虑新加入的光伏电板，此时对于台区变压器的充电由两部分构成。一是来自于移动储能车的充电，另一部分就是来自于光伏发电板的供给。结合问题一与光伏出力的每十五分钟分布式预测数据修改模型中的约束数据，重新对移动储能车进行配置。

对于问题三，也是在之前的基础上在模型中增加新的限制，即考虑进实际情况中道路拥挤等不可预测的不确定性行为。根据城市不同时间段的拥挤系数重新规划调配模型，进一步求解。

4.2 调度模型建立

4.2.1 问题一

由于车辆 7:00 出发，22:00 前返回出发地，期间共 15 个小时，因此将一天分为 $T=15$ 个时段，令 $i = \{1, \dots, 10\}$ 代表了十台储能车，其中 1-5 代表 MV#1, 6-10 代表 MV#2。令 j 代表储能车所处的位置， $P_i^c(t)$ 和 $P_i^d(t)$ 分别表示车 i 在 t 时段充电和放电电量， $x_{i,j}^c(t)$ 为车在 j 点是否充电的 0-1 变量，0 表示不充电 1 表示充电， $x_{i,j}^d(t)$ 为车在 j 点是否放电的 0-1 变量，0 表示不放电 1 表示放电， $C^c(t)$ 为 t 时刻充电电价， $C^d(t)$ 为 t 时刻放电电价， FC_i 为车 i 的单公里油耗， $X_{i,l}^m(t)$ 表示车辆是否在路 l 上行驶的 0-1 变量，0 表示不在 1 表示在， Di_l 代表了路 l 的长度。根据上述假设一天之内所有车辆的总收益为：

$$\sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I x_{i,j}^d(t) \times P_i^d(t) \times C^d(t) \quad (1)$$

因充电造成的成本为：

$$\sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I x_{i,j}^c(t) \times P_i^c(t) \times C^c(t) \quad (2)$$

因移动位置造成的成本为：

$$\sum_{i=1}^I \sum_{t=1}^T \sum_{l=1}^L FC_i \times X_{i,l}^m(t) \times Di_l \quad (3)$$

综上可得所有车辆一天的总利润为：

$$\begin{aligned} & \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I \left[x_{i,j}^d(t) \times P_i^d(t) \times C^d(t) - x_{i,j}^c(t) \times P_i^c(t) \times C^c(t) \right] \\ & - \sum_{i=1}^I \sum_{t=1}^T \sum_{l=1}^L FC_i \times X_{i,l}^m(t) \times Di_l \end{aligned} \quad (4)$$

因此只需最小化上述利润的负值便可得到最优路径规划，即：

$$\begin{aligned} \min F = & \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I \left[x_{i,j}^c(t) \times P_i^c(t) \times C^c(t) - x_{i,j}^d(t) \times P_i^d(t) \times C^d(t) \right] \\ & + \sum_{i=1}^I \sum_{t=1}^T \sum_{l=1}^L FC_i \times X_{i,l}^m(t) \times Di_l \end{aligned} \quad (5)$$

下面考虑各项约束。

首先是状态约束，即每辆车应有一个状态，因此应满足如下约束条件：

$$\sum_{j=1}^J (X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t)) + \sum_{l=1}^L X_{i,l}^m(t) = 1 \quad (6)$$

然后每个时刻，充放电点只有一辆车，因此对于任意地点 和任意 t 都有：

$$0 \leq \sum_{i=1}^{10} (X_{i,j}^c(t)) \leq 1 \quad (7)$$

$$0 \leq \sum_{i=1}^{10} (X_{i,j}^d(t)) \leq 1 \quad (8)$$

进一步考虑负载约束。假设 时刻 处的需要的下降的功率为 ,则对于任意 t 和任意 j 都有

$$\sum_{i=1}^{10} (X_{i,j}^d(t) \times P_i^d(t)) \geq W_j(t) \quad (9)$$

同时每辆车的电量不能为复数，因此对于任意 i , 和任意 t 有

$$W_i - \sum_{t=1}^T (X_{i,j}^c(t) \times P_i^c(t)) + \sum_{t=1}^T (X_{i,j}^d(t) \times P_i^d(t)) \geq 0 \quad (10)$$

其中 W_i 为初始车电容量。根据题目要求，每天完成任务后每辆车应充满电，因此对于任意 i 需要满足：

$$-\sum_{t=1}^{15} (X_{i,j}^c(t) \times P_i^c(t)) + \sum_{t=1}^{15} (X_{i,j}^d(t) \times P_i^d(t)) = 0 \quad (11)$$

车辆在不同点移动时还需满足状态转移约束，首先作为终点驶入的站点，下一时刻如果车要移动，必走以该站点为起点的路径，因此对于任意 t 和 j 有

$$\begin{aligned} & \sum_{l:D=j}^L X_{i,l}^m(t) + X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \\ &= \sum_{l:o=j}^L X_{i,l}^m(t+1) + X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1) \end{aligned} \quad (12)$$

同时 t 时刻进入站点加停留的车辆数和 $t+1$ 时刻驶出站点加停留车辆数应相等，因此对于任意 t 和 j 有

$$\begin{aligned} & \sum_{l:D=j}^L \sum_{i=1}^l X_{i,l}^m(t) + \sum_{i=1}^l (X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t)) \\ &= \sum_{l:0=j=1}^L \sum_{i,1}^l X_{i,l}^m(t+1) + \sum_{i=1}^l (X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1)) \end{aligned} \quad (13)$$

综合目标函数和约束条件，最优路径规划问题可表示为如下模型：

$$\begin{aligned}
\min F &= \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I \left[x_{i,j}^c(t) \times P_i^c(t) \times C^c(t) - x_{i,j}^d(t) \times P_i^d(t) \times C^d(t) \right] \\
&+ \sum_{i=1}^I \sum_{t=1}^T \sum_{l=1}^L FC_i \times X_{i,l}^m(t) \times Di_l \\
st. & \\
&\sum_{j=1}^J \left(X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \right) + \sum_{l=1}^L X_{i,l}^m(t) = 1; \\
&0 \leq \sum_{i=1}^{10} \left(X_{i,j}^c(t) \right) \leq 1; \\
&0 \leq \sum_{i=1}^{10} \left(X_{i,j}^d(t) \right) \leq 1; \\
&\sum_{i=1}^{10} \left(X_{i,j}^d(t) \times P_i^d(t) \right) \geq W_j(t); \\
&W_i - \sum_{t=1}^T \left(X_{i,j}^c(t) \times P_i^c(t) \right) + \sum_{t=1}^T \left(X_{i,j}^d(t) \times P_i^d(t) \right) \geq 0; \\
&-\sum_{t=1}^{15} \left(X_{i,j}^c(t) \times P_i^c(t) \right) + \sum_{t=1}^{15} \left(X_{i,j}^d(t) \times P_i^d(t) \right) = 0; \\
&\sum_{l:D=j}^L X_{i,i}^m(t) + X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \\
&= \sum_{l:o=j}^L X_{i,i}^m(t+1) + X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1); \\
&\sum_{l:D=j}^L \sum_{i=1}^I X_{i,l}^m(t) + \sum_{i=1}^I \left(X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \right) \\
&= \sum_{l:0=j=1}^L \sum_{i=1}^I X_{i,l}^m(t+1) + \sum_{i=1}^I \left(X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1) \right).
\end{aligned}$$

最终求得移动储能车的收益为 551 元。以一号移动储能车为例分析其调度：调度周期起始时在 1 节点，因 5 节点台区变压器距离 1 比较近，我们先将其通过节点 4 到达台区变进行充电，以 200KW 的功率放电 1h；此时 5 节点的用电压力得到缓解。在第一号车被调配回 1 节点直接去到 6 节点充电至满电状态而后通过第 3 节点和第 2 节点回到节点 1，为下一调度周期的运行做好准备。

4.2.2 问题二

目标函数与问题 1 一致，因此只需调整约束条件。考虑光伏发电，可以减少不同节点 j 需要的下降的功率为 $W_j(t)$ ，因此只需根据发电数据核减并更新问题 1 模型中的 $W_j(t)$ 便可得到问题 2 的优化模型，记核减后的 $W_j(t)$ 为 $W_j^\dagger(t)$ ，则问题 2 的优化模型为：

$$\begin{aligned}
\min F &= \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I \left[x_{i,j}^c(t) \times P_i^c(t) \times C^c(t) - x_{i,j}^d(t) \times P_i^d(t) \times C^d(t) \right] \\
&+ \sum_{i=1}^I \sum_{t=1}^T \sum_{l=1}^L FC_i \times X_{i,l}^m(t) \times Di_l \\
st. & \\
&\sum_{j=1}^J \left(X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \right) + \sum_{l=1}^L X_{i,l}^m(t) = 1; \\
&0 \leq \sum_{i=1}^{10} \left(X_{i,j}^c(t) \right) \leq 1; \\
&0 \leq \sum_{i=1}^{10} \left(X_{i,j}^d(t) \right) \leq 1; \\
&\sum_{i=1}^{10} \left(X_{i,j}^d(t) \times P_i^d(t) \right) \geq W_j^\dagger(t); \\
&W_i - \sum_{t=1}^T \left(X_{i,j}^c(t) \times P_i^c(t) \right) + \sum_{t=1}^T \left(X_{i,j}^d(t) \times P_i^d(t) \right) \geq 0; \\
&-\sum_{t=1}^{15} \left(X_{i,j}^c(t) \times P_i^c(t) \right) + \sum_{t=1}^{15} \left(X_{i,j}^d(t) \times P_i^d(t) \right) = 0; \\
&\sum_{l:D=j}^L X_{i,l}^m(t) + X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \\
&= \sum_{l:o=j}^L X_{i,l}^m(t+1) + X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1); \\
&\sum_{l:D=j}^L \sum_{i=1}^I X_{i,l}^m(t) + \sum_{i=1}^I \left(X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \right) \\
&= \sum_{l:0=j=1}^L \sum_{i=1}^I X_{i,l}^m(t+1) + \sum_{i=1}^I \left(X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1) \right).
\end{aligned}$$

最终求得收益为 611 元。

4.2.3 问题三

需要考虑排队充电，以及由于拥堵造成的速度降低，首先排队只需根据先后时间确定状态值，因此需要增加约束：

$$x_{i1,j}^c(t+1) - x_{i2,j}^c(t) = 1, i1 \neq i2 \quad (14)$$

而由于阻塞导致有充电站，台区变和初始的节点不能直接可达，因此只需将其余节点考虑进去便能满足约束，即 j 可取 1-29。综上问题 3 的优化模型可写为：

$$\begin{aligned}
\min F &= \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I \left[x_{i,j}^c(t) \times P_i^c(t) \times C^c(t) - x_{i,j}^d(t) \times P_i^d(t) \times C^d(t) \right] \\
&+ \sum_{i=1}^I \sum_{t=1}^T \sum_{l=1}^L FC_i \times X_{i,l}^m(t) \times Di_l \\
st. & \\
&\sum_{j=1}^J \left(X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \right) + \sum_{l=1}^L X_{i,l}^m(t) = 1; \\
&0 \leq \sum_{i=1}^{10} \left(X_{i,j}^c(t) \right) \leq 1; \\
&0 \leq \sum_{i=1}^{10} \left(X_{i,j}^d(t) \right) \leq 1; \\
&\sum_{i=1}^{10} \left(X_{i,j}^d(t) \times P_i^d(t) \right) \geq W_j^+(t); \\
&W_i - \sum_{t=1}^T \left(X_{i,j}^c(t) \times P_i^c(t) \right) + \sum_{t=1}^T \left(X_{i,j}^d(t) \times P_i^d(t) \right) \geq 0; \\
&-\sum_{t=1}^{15} \left(X_{i,j}^c(t) \times P_i^c(t) \right) + \sum_{t=1}^{15} \left(X_{i,j}^d(t) \times P_i^d(t) \right) = 0; \\
&\sum_{l:D=j}^L X_{i,l}^m(t) + X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \\
&= \sum_{l:o=j}^L X_{i,l}^m(t+1) + X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1); \\
&\sum_{l:D=j}^L \sum_{i=1}^I X_{i,l}^m(t) + \sum_{i=1}^I \left(X_{i,j}^c(t) + X_{i,j}^d(t) + X_{i,j}^h(t) \right) \\
&= \sum_{t:0=j=1}^L \sum_{i,1}^I X_{i,l}^m(t+1) + \sum_{i=1}^I \left(X_{i,j}^c(t+1) + X_{i,j}^d(t+1) + X_{i,j}^h(t+1) \right); \\
&x_{i1,j}^c(t+1) - x_{i2,j}^c(t) = 1, i1 \neq i2.
\end{aligned}$$

由于道路状况存在拥堵的现象，对小车的行驶造成了影响，使得最终求得收益进一步减少为 445 元

5 模型评价与推广(或改进)

5.1 模型的优点

- (1) 本文在有逻辑地分析了问题的前提下，建立了科学的计算模型，为谋得最大收益准备了条件。
- (2) 有可靠的数学基础，具有不错的可解释性。
- (3) 模型的建立易于实现，言简意赅。
- (4) 本文采用了专门的混合线性规划的求解器 Yalmip，求解速度准确且快。
- (5) 本文首先恰当的处理了数据，使得问题思路脉络清晰。

5.2 模型的不足

- (6) 为了更好地求解模型，使用了一些模型假设，使得算出来的规划与真实情况仍然存在差异
- (7) 我们建立的模型引入的变量比较多，容易引起“维数灾难”，且不利于编程处理。

5.3 模型的推广【或改进】

本文实际上是在电网调配背景下的运筹规划问题，相同的思路同样可以运用到其他的场景中。本文中对时间采用了离散化的处理，以时间点为基准，时间间隔选为一小时。这是比较粗略地一种处理方法，下一步我们可以将时间间隔细化为 30 分钟，15 分钟，甚至时 1 分钟，逐渐逼近连续的真实情形。

参考文献

- [1] 邵宇峰. 主动配电网移动储能优化配置与调度策略研究[D]. 华北电力大学,2021.
- [2] 肖凯超,邱伟强,陶以彬,等. 储能应急车优化调度的模糊机会约束方法[J]. 高压电器,2021,57(2):116-124. DOI:10.13296/j.1001-1609.hva.2021.02.017
- [3] 李靖霞,纪陵,左建勋,等. 基于遗传算法的移动储能车调度方案优化及应用[J]. 浙江电力,2020,39(3):50-53. DOI:10.19585/j.zjdl.202003008
- [4] 王伟. 提升扰动发生后电网主动应对能力的储能资源时空灵活调度研究[D].重庆大学,2021.DOI:10.27670/d.cnki.gcqdu.2021.000055.

附 录

附录 I: 表示调度方案的矩阵

见文件 A_Scheduling: 是尺度为 150*90 的数据, 表示 10 辆车在移动的 15 个小时中的状态, 以及行走的路径。

附录 II: 主要程序/关键代码

代	操作系统: Win 10
码	编程语言: Matlab 2017b
环	编辑器: PyCharm 2018.3.2 (Professional Edition)
境	代码详见: Scripts.m

```
clear;clc;close all;
T=150;
J=9;
load('distance.mat')
load('data_c.mat')
load('data_d.mat')
load('load_mitigation.mat')

% 创建决策变量
L=length(find(distance>0));

t=0;
for i=1:9
    for j=1:9
        if distance(i,j)>0
            t=t+1;
            D_l(t)=distance(i,j);
        end
    end
end

end
C = sdpvar(T,2*J+L); %i = 10: 车辆; j = 9: 车辆可以处的位置; 1 : 最短路径
% 添加约束条件
D = [];
for t = 0:14
    for i = 1:10
        [D;sum(C(10*t+i,1:9),2)+sum(C(10*t+i,10:18),2)+sum(C(10*t+i,19:18+L),2)==1];
    end
end
end
```

```

for t = 0:14
    for j = 1:9
        [D;sum(C(10*t+1:10*t+10,j),1)<=1];
        [D;sum(C(10*t+1:10*t+10,j),1)>=0];

        [D;sum(C(10*t+1:10*t+10,j+9),1)<=1];
        [D;sum(C(10*t+1:10*t+10,j+9),1)>=0];
    end
end

for c_i=1:5
    P_c(c_i)=200;
end
for c_i=6:10
    P_c(c_i)=400;
end
P_c=repmat(P_c',T/10,1);

for c_i=1:5
    P_d(c_i)=50;
end
for c_i=6:10
    P_d(c_i)=100;
end
P_d=repmat(P_d',T/10,1);

for t=0:14
    for j = 1:4
        [D;C(10*t+1:10*t+10,j)*P_d(10*t+1:10*t+10)>=load_mitigation(t+7,j)];
    end
end

for i=1:10
    for j = 1:9
        S1=0;
        S2=0;
        for t = 0:14
            S1 = S1 + C(10*t+i,j);
            S2 = S2 + C(10*t+i,j+9);
        end
    end
end

```

```

        end
        [D;P_c(i)-S1*P_c(i)+S2*P_d(i)>=0];
        [D;-S1*P_c(i)+S2*P_d(i)==0];
    end
end

% 状态转移约束
% 刚作为终点驶入的站点，下一时刻如果车要移动，必走以该站点为起点的路径。
% 对于任意 t 和 j 有, xm 为 9*9 的矩阵，对应路径

for t = 0:14
    for j = 1:9
        for i=1:10

[D;C(10*t+i,j)+C(10*t+i,j+9)+C(10*t+i,j+18)+sum(C(10*t+i,j+18:8:18+72))]==C(10*(t+1)+i,j)+C(10*(t+1)+i,j+9)+C(10*(t+1)+i,j+18)+sum(C(10*(t+1)+i,j+18+8*j-7:1:j+18+8*j))];
        end
    end
end

% 流量约束
% 保证车不会消失，t 时刻进入站点+停留和 t+1 时刻驶出站点+停留的车数一致
% 对于任意 t 和 j 有
for t = 0:14
    for j = 1:9

[D;sum(C(10*t+1:11*t,j)+C(10*t+1:11*t,j+9)+C(10*t+1:11*t,j+18))+sum(C(10*t+1:11*t,j+18:8:18+72))]==sum(C(10*(t+1)+1:11*(t+1),j)+C(10*(t+1)+1:11*(t+1),j+9)+C(10*(t+1)+1:11*(t+1),j+18))+sum(C(10*(t+1)+1:11*(t+1),j+18+8*j-7:8*j))];
    end
end

%一天调度满电

% 配置
ops = sdpsettings('verbose',0,'solver','lpsolve');
% 目标函数

```

```

C_c= repmat(data_c(1),10,1);
C_d= repmat(data_d(1),10,1);
for c_t=2:T/10
    C_c=[C_c; repmat(data_c(c_t),10,1)];
    C_d=[C_d; repmat(data_d(c_t),10,1)];
end

for c_i=1:5
    FC(c_i)=1;
end
for c_i=6:10
    FC(c_i)=2;
end
FC= repmat(FC',T/10,1);
D_1= rand(1,L);
D_1= repmat(D_1,T,1);

for c_j=1:J
    cost_c(:,c_j)=C(:,c_j).*P_c.*C_c;
    cost_d(:,c_j)=C(:,c_j+J).*P_d.*C_d;
end

for c_l=1:L
    cost_l(:,c_l)=C(:,c_l+2*J).*FC.*D_1(:,c_l);
end

z=sum(sum(cost_c))-sum(sum(cost_d))+sum(sum(cost_l));
% 注意这是求解最大值
% 求解

reuslt = optimize(D,z);
if reuslt.problem == 0 % problem =0 代表求解成功
    value(x)
    -value(z) % 反转
else
    disp('求解出错');

```

end