

# 关于在大电网坍塌情况下，通过运输蓄电池解决节点负荷问题的几点思考

Zihan Zhang

July 2019

## Contents

|               |          |
|---------------|----------|
| <b>1 问题重述</b> | <b>1</b> |
| 1.1 基础问题      | 1        |
| 1.2 问题延伸      | 1        |
| <b>2 参数梳理</b> | <b>2</b> |
| <b>3 模型思路</b> | <b>2</b> |
| <b>4 模型反思</b> | <b>3</b> |

## 1 问题重述

### 1.1 基础问题

假设在维度为  $l \times l$  的网格图中（边长权重未知）存在一供电站，其单位时间供电量为  $v_{refill} kWh$ 。该供电站原储存有总容量为  $x kWh$  的电池，可被分割成任意尺寸。忽然大电网坍塌，该地区有  $k$  个不同节点上发生停电，负荷 ( $V$ ) 的规模可由随机数生成，需要派遣送电车进行供能。假设在正常供电过程中单位时间收益值为  $\$p$ ，而在停电过程中单位时间损失值为  $\$l$ 。若该地区共有  $a$  辆供电车且供电车的承载量为  $ckWh$  的电池。如一地区收到  $x_0 kWh$  的电池且负荷量为  $V_0$  时，则该地区可维持  $\frac{x_0}{V_0}$  小时的供电状态后又再次陷入停电，直到下一次供电车来回收并替换新电池。当供电车到达时，如该地区剩余电池容量仅为负荷的  $0\% \sim 10\%$  范围内，则进行替换。供电车可一次前往一地或多地。由于受灾原因，该地区一部分公路 ( $r$  条) 无法通行，只能选择绕路的方式尽快完成抢修。在各参数已知的情况下，计算该地区应最优存储有多少电池可满足该地区的供电需求？

### 1.2 问题延伸

- 如果送电车为电动车，在送电的过程中消耗电池的储电量（消耗量与运送距离线性相关），会对结果造成什么影响？

- 如果考虑电池在拆卸过程中所耗费的时间（与电池储电量线性相关），会对结果造成什么影响？
- 如果改变该地区的总用车量，会对结果造成什么影响？
- 如何平衡设置各参数之间的大小关系，使其更符合真实情况？

## 2 参数梳理

|                       |                                      |
|-----------------------|--------------------------------------|
| $l$                   | 网格图边长 ( $km$ )                       |
| $x$                   | 总电池储存量 ( $kWh$ )                     |
| $x_{ij}$              | 给第 $j$ 目的地第 $i$ 次运输时提供的电池量 ( $kWh$ ) |
| $t_{xy}$              | 从 $x$ 点到 $y$ 点所需要的最短时间 ( $h$ )       |
| $k$                   | 发生停电的地点数 (个)                         |
| $a$                   | 单位时间收益值（供电过程中）( $\$$ )               |
| $c$                   | 单位时间损失值（停电过程中）( $\$$ )               |
| $a$                   | 供电车的数量 (辆)                           |
| $C$                   | 供电车的电池承载量 ( $kWh$ )                  |
| $V$                   | 负荷值 ( $kWh$ )                        |
| $[V_{low}, V_{high}]$ | 负荷随机数生成的范围 ( $kWh$ )                 |
| $r$                   | 由于受灾原因无法通行的道路的数量 (条)                 |
| $v_{refill}$          | 单位时间供电站供电量 ( $kWh$ )                 |

## 3 模型思路

- 本模型的目标是令在后续的配送过程中不会出现再出现断电情况，以保证该地区自始至终满足供电需求，自给自足。
- 假设  $k = 6, a = 100, c = 700, C = 100, v_{refill} = 200, [V_{low}, V_{high}] = [30, 50]$
- 通过随机数生成供电站和停电地点的坐标。
- 利用随机数生成节点间边的权重大小，同时随机生成多条断路（权重为无限大）。
- 利用 Dijkstra 算法和邻接矩阵将所有目标地点归类（如  $t_{02} = t_{01} + t_{12}$ ，则第 1 目的地和第 2 目的地可被划分在一类中），分别将其加入到不同列表中，计算出能覆盖该地区运输的最小用车量，即为列表的总数。
- 为各地区的电池分配设定初始值，即  $\frac{x_{1h}}{V_h} = 2t_{0m}$ ，其中  $h \in [1, k], h \in \mathbb{Z}$ ，而  $t_{0m}$  为特定  $k$  所处的地点列表中耗时最远的点所对应的时间。
- 将计算得出的  $x_{1h}$  加入到 pandas.DataFrame 中，并设置为第一行。
- 定义  $\Delta$  为派遣到  $h$  所处的地点列表的运输车返回到供电站时，供电站内剩余未充电电池的数量。在计算时，将每辆车回到原点的周期相应的倍数加入到 np.array

中。分别选取每两个相邻的时间点，假设其为  $t_0, t_1$ ，同时假设在  $t_0$  运回来的总待充电电池量为  $x_0$ （有可能来自一辆或多辆车）。如  $x_0 \leq (t_1 - t_0) \times v_{refill}$ ，则  $\Delta = 0$ ；如  $x_0 > (t_1 - t_0) \times v_{refill}$ ，则  $\Delta = x_0 - (t_1 - t_0) \times v_{refill}$ 。分别将  $\Delta$  按照其对应的时间点和地点坐标加入到 DataFrame 中。

- 求解方程  $\frac{x_{2h}}{V_h} = 2t_{0m} + \frac{\sum x_{1n} + \Delta}{v_{refill}}$ ，其中  $\sum x_{1n}$  为所有与  $h$  处于同一地点列表中的地点在第一次所收到的电池总量。
- 将以下方程迭代 1000 次，即  $\frac{x_{lh}}{V_h} = 2t_{0m} + \frac{\sum x_{(l-1)n} + \Delta}{v_{refill}}$ ， $l \in [2, 1000]$ ，分别将每次计算得到的  $x_{lh}$  的值加入到 pandas.DataFrame 中。
- 计算出每一行的和，即为每一次给各个目的地运输的电池量总和，定义为  $X_k$ ，并利用 matplotlib.pyplot 绘制出折线图。
- 给相邻两行的和求和，计算出所有结果的最大值即为该地区最少需求的电池总量。

## 4 模型反思

- 本模型仅针对最少车辆的环境下进行分析。如果调整参数，修改车辆的个数，代码需要进行相应变动，说明代码的灵活性和兼容性较低。
- Dijkstra 算法的复杂度为  $O(E + VlgV)$ ，相比较 BellmanFord 和 SPFA 的算法复杂度均为  $O(EV)$ ，结合本题的参数设定具有较低的复杂度。
- 在分析结果时，数据显示结果与迭代次数形成近似于振荡函数的关系。需要注意的是，随着迭代次数增加，电池量总和并未达到收敛，说明迭代 1000 次所计算出的最小需求电池量并不能保证能满足未来所有的需求。而如果一味调高迭代次数会使得时间成本过高，效率极低。这潜在的计算误差是本算法目前无法解决的一个问题。

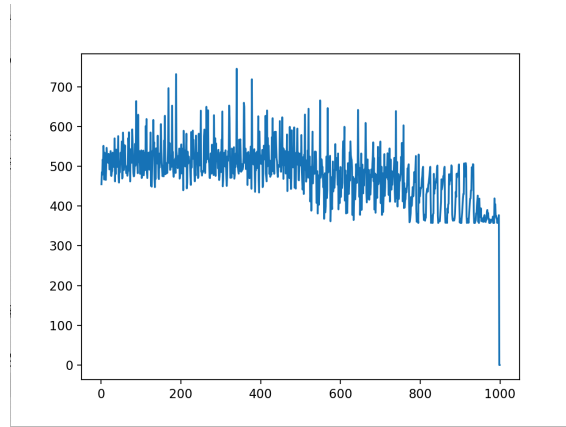


Figure 1: 电池量总和与迭代次数的关系