

基于蒙特卡洛树搜索的移动储能车优化调度

摘要

随着城市建设和经济的高速发展，城区变电所供电容量常常无法满足负荷高峰期的用电需求。而移动储能车可以配合电网进行削峰填谷完成重要单位的保电任务。但移动储能车的行驶路径及充放电策略需要根据用电需求交通流量和电网电价等进行灵活规划。本文通过建立移动储能车的规划调度模型并利用蒙特卡洛搜索树对模型进行了求解。

针对问题一：对题目所给信息进行分析建模：移动储能车的利润分为两部分——一部分为行驶里程的消耗，另一部分为充放电的收益。约束条件为移动储能车仅能充放电一次、移动储能车放电功率不能超过台区总用电需求、移动储能车需满电返回且需要在晚 22 点前返回车辆存放点、放电服务需要按用户类型按顺序满足以及台区变的负载率。针对所建立的规划调度模型，利用蒙特卡洛搜索树分两阶段进行求解。第一阶段，建立移动储能车分配、充放电时间方案，并对其进行优化。第二阶段，对储能车的充放电路径进行优化。在具体的算法中将利润作为奖励函数的主体，并将约束条件作为罚函数加入奖励函数中。将非线性约束问题转化为无约束问题。

针对问题二：在问题一的基础上，移动储能车更改为可以多次充放电。并且增加了光伏发电装置。在保障三类用户的用户需求下，要求用到的移动储能车数量最少。所以本问题的目标函数更改为移动储能车的数量。约束条件更改为移动储能车可以进行多次充放电。此时，蒙特卡洛树搜索的目标为动用最少数量储能车的方案。由于移动储能车的类型和光伏发电费用的不同，可以基于利润高低对搜索到的调度方案进行再次优化。

针对问题三：在问题一的基础上，移动储能车的移动出现拥塞现象，充电出现排队现象，负载率服从正态分布。我们在建模过程中，对充电站类型的实体增加一个随时间变化而更新的变量排队时间 `TimeQueue`。每时间点新增的排队车数和预计排队时间从相应的随机分布中获得。当储能车准备在对应充电站充电时，会根据充电站实体的排队时间变量确定自身的预计充满电时间。基于问题一的数学建模与仿真框架，我们将未能在 22 点前返回出发点的调度车数量作为一个惩罚项纳入算法的约束条件。利用蒙特卡洛树搜索进行求解。

关键词：电力负载率 蒙特卡洛树搜索 移动储能车 调度

目录

基于蒙特卡洛树搜索的移动储能车优化调度	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	1
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	2
3.1 负载率计算	2
3.2 最短路径计算	3
4 问题分析与模型建立	4
4.1 问题分析	4
4.2 蒙特卡洛树搜索	5
4.3 问题一	6
4.3.1 数学模型	6
4.3.2 算法设计	7
4.3.3 结果与分析	9
4.4 问题二	11
4.4.1 数学模型	11
4.4.2 算法设计	11
4.4.3 结果与分析	12
4.5 问题三	14
4.5.1 数学模型	14
4.5.2 算法设计	15
4.5.3 结果与分析	17
5 模型评价	18
5.1 模型的优点	18
5.2 模型的不足	19
参考文献	20
附 录	21
附录 A: 可停靠点之间的最短路径	21
附录 B: 关键数据结构与程序伪代码	21

1 问题综述

1.1 问题背景

随着城市建设和经济的高速发展，城市人口不断增加，用电负荷不断增强，原有的城区变电所供电容量无法满足负荷高峰期的用电需求。移动储能车的应用可以有效协助解决这个问题。移动储能车通过装载一定容量的储能电池，可以配合电网实现削峰填谷，完成保电任务，有效提升配网末端用户的满意度。

储能车的移动、充电都需要一定的成本，而放电除能降低相应台区变的负载率外，还能根据放电时的电力价格获得收益。因此，合理规划移动储能车的调度方案能够在满足用户的需求情况下，尽可能的提升电量充放利润。在以往的研究中，分别着重于车辆的路线调度[1]或者移动储能车的充放电调度[2]，且所研究的储能车数量、台区变数量和快速充电站数量较少，求解难度较低且很多干扰因素未能充分考虑。因此我们需要面向交通网络，优化移动储能车的调度路径的同时，优化充放电方案。

1.2 问题提出

移动储能车的优化调度问题是一个典型的序贯决策问题，其具有解空间大、决策变量多的特点。移动储能车的调度方案主要由行驶路径，充放电时间构成。为了得到满足要求的方案需要在下列四点约束下：

- 一天为一个调度周期，每小时为一个时间间隔。
- 移动储能车满电情况下在集合存放点发车，一天结束后需充满电回到车辆集合存放点。
- 每台台区变在同一时段内只能接入一辆移动储能车，每个快速充电站同一时段只能给一辆移动储能车充电。
- 按照一二三优先级保障三类用户。

主要解决以下 3 个问题：

- (1) 问题 1：合理调度移动储能车使其利润最优。
- (2) 问题 2：合理调度移动储能车使调用数量最少。
- (3) 问题 3：在存在多种干扰的情况下，合理调度移动储能车使其利润最优。

1.3 资料条件

附件提供了负荷、电价数据，光伏出力数据以及 29 个节点数据，各个文件的详细说明如下：

- 附件 1：负荷、电价数据.xlsx 该文件详细介绍了各个时间点的四个台区变变负荷和各个时间点充放电价格，分别用于计算台区变的负载率和充放电利润。
- 附件 2：光伏出力数据.xlsx 该文件提供了光伏发电板在各个时间点的发电功率，主要用于问题二中计算移动储能车的充电费用。由于光伏发电价格较为便宜，有助于降低移动储能车的充电消耗。
- 附件 3：29 个节点数据.xlsx 该文件主要介绍了城市交通网络节点之间的连接关系，以及各个节点之间的距离，用于计算节点之间的最短路径以降低行程消耗。

2 模型假设与符号说明

2.1 模型基本假设

- (1) 假设储能车只在整点时刻进行状态改变。既仅在整点时刻决定是否前往目标地点，是否进行充放电操作。
- (2) 假设充电站在一个整时间间隔内可以并且仅可以对两辆 MV#1 型车辆充电或者仅可以对一辆 MV#2 型车辆充电。
- (3) 假设充电站的充电功率恒定为 400kW。
- (4) 假设光伏发电板的存在可以代替一部分电网输出，使得充电费用降低。
- (5) 移动储能车可以以不超过其额定功率的任意功率工作。

2.2 符号说明

本文定义了如下 9 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
t	时刻	h
k	移动储能车编号	无
s_k	储能车集合	无
$dis_k(t)$	第 k 辆车在 t 时刻的位置	无
$power_k(t)$	第 k 辆车在 t 时刻的电量	kWh
n	台区变编号	无
s_n	台区变集合	无
β_n	第 n 台台区变负载率	1
P_{kn}	第 k 辆车为第 n 个台区变的输出功率	kw
payoff	总服务利润	无

3 数据预处理

3.1 负载率计算

利用附件 1，计算各个台区变的实际负载率 $\beta = \frac{\text{台区变变荷数}}{0.8 \times \text{台区变额定容量} S}$ 。

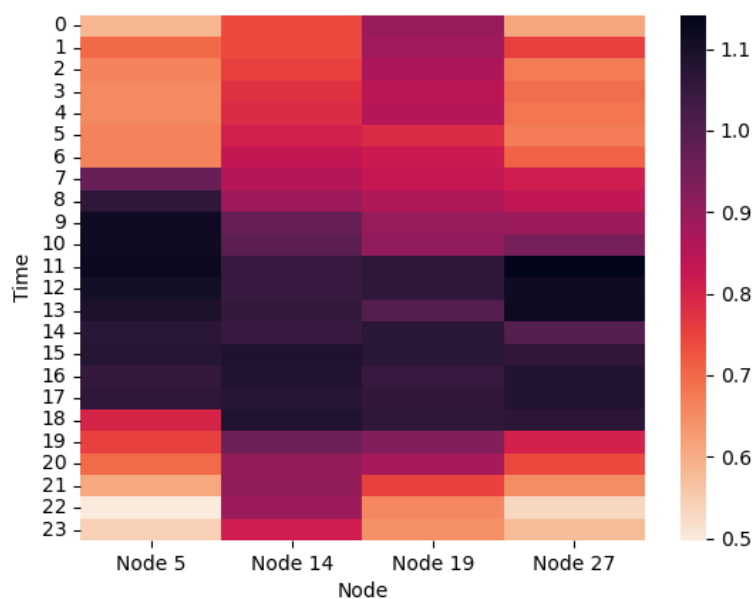


图1 各个台区变在不同时间的负载率

3.2 最短路径计算

由于除出发节点、台区变、快速充电站外，移动储能车不能停靠在其余位置，同时为了降低行驶里程带来的资源消耗，增加利润，仅需要考虑各个可停靠点之间的最短路径，使得移动储能车能够以最短路径从任意停靠点到达另一停靠点。可停靠点之间的最短路径示意图如下，其中线条越粗表示路径越近。

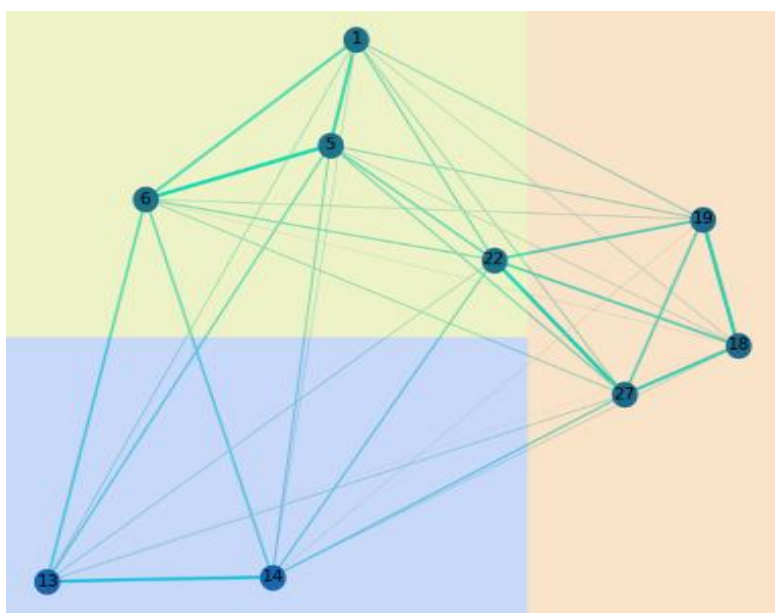


图2 可停靠节点之间的最短路径示意图

可停靠点之间的具体最短路径如附录 A 所示。

4 问题分析与模型建立

4.1 问题分析

通过对移动储能车的任务调度能够有效缓解变电所供电量不足的问题。题目以此为背景，介绍了移动储能车的工作流程，问题一要求我们在移动储能车仅完成一次充放电的情况下，合理安排调度方案使得充放电带来的收益最高，问题二要求我们在问题一的基础上，在移动储能车可以多次充放电的情况下，安排调度方案使得在按优先级满足各类用户供电的情况下，车的调用数量最少同时尽可能的提升利润。问题三在问题一的基础上，进一步考虑可能的交通拥塞以及充电站使用冲突的情况，合理安排方案使得充放电利润达到最大。

基于题中给出的信息，由于移动储能车的运营规则，我们假设其状态仅在整点改变；由于充电功率的原因，我们认为对于快速充电站每小时可以充满两辆 MV#1 型移动储能车或者充满一辆 MV#2 型移动储能车；此外，由于光伏发电板的增加，我们假设此时光伏发电可以为另一台进入充电站的车辆充电。

在整个题目中，围绕移动储能车的调度优化。将移动调度车的收益、数量作为目标函数，同时考虑优先满足、最短路径、条件不确定性，建立移动储能车的移动路径，充放电时间、地点的调度方案。

4.1.1 问题一:

移动储能车分为两种规格 MV#1 和 MV#2，两种车的属性如下：

表2 移动储能车的属性

车辆类型	额定功率(kW)	额定容量(kWh)	时速(km/h)	成本(¥/km)
MV#1	50	200	30	1
MV#2	100	400	30	2

两种类型的车辆共十辆，最终目标为移动储能车总充放电服务收益最大，既使得储能车在尽量少的驾驶里程下，尽可能赚取较多的电费差价。所给出的优化方案应优先满足 I 类用户（对应台区变#19、#27）、其次满足 II 类用户（对应台区变#5）、最后满足 III 类用户（对应台区变#14）。移动储能车出发（7：00 后）和回收（22：00 前）时都应处于满电状态，且只能充放电一次。此外，移动储能车的放电功率不能超过总用电需求。

移动储能车只能在出发地（#1）、快速充电站（#6、#13、#18、#22）和台区变（#5，#14，#19，#27）节点停留，所以为了简化移动储能车位置状态以及初步优化驾驶里程，需要计算图中各个可停靠点之间的最短路径，并忽略非最短路径上的不可停靠点。然后，基于最短路径图，建立非线性优化模型求解。

4.1.2 问题二:

在问题一的基础上，问题二的目标为利用尽可能少的移动储能车，但是考虑到大小车之间的搭配以及利润问题，本题的总目标应该为在调用最少数量车辆情况下，合理安排使得利润最高。此时，移动储能车可以进行多次充放电，所以可以通过交替放电、空歇充电、优先供给等手段尽可能提高移动储能车的使用率。并且由于光伏发电板的存在，移动储能车的充电费用较全部利用快速充电有所降低，但光伏发电板的发电时间和功率有限，且仅能在部分时段使用，也不能代替快速充电站。

4.1.3 问题三:

在问题一的基础上，道路出现拥堵情况，快速充电站出现排队情况且台区变负荷变为服从正态分布的随机变量。本题主要考察问题一所建模型以及求解方法在面对多不确定性时的韧性。需要结合排队论解决移动储能车的充电等待时长等问题，结合拥堵情况完善模型以及求解方法。

4.2 蒙特卡洛树搜索

4.2.1 蒙特卡洛树搜索简介

蒙特卡洛树搜索是一种经典的树搜索算法，能够真实的模拟序贯决策的实际物理过程，有效解决实际问题。名震一时的 **alphago** 技术就结合了蒙特卡洛树搜索从而击败了当时的围棋世界冠军。通过随机模拟和价值评估，蒙特卡洛树搜索通过将算力集中在更有价值的地方有效解决了大规模搜索空间的组合优化问题。

4.2.2 算法主要流程

蒙特卡洛树搜索算法是在序贯决策任务中采用的有效搜索算法，其采用选择、拓展、模拟、回溯 4 个流程实现高效的剪枝与搜索。

如果我们把序贯决策中特定的**状态**看作一个节点，由该状态可采取不同的**动作**。根据采取的动作，状态也将随之转移。由于同一个状态下可采取的动作不唯一，由一个状态可以转移到多个新的状态。由此，序贯决策实际上可以看作一个树状空间。具体流程图如下：

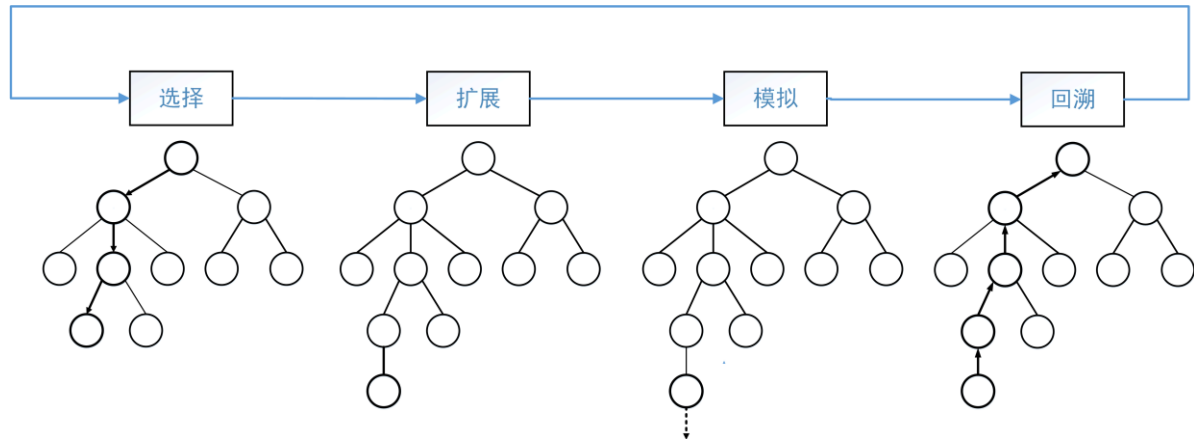


图3 蒙特卡洛树搜索流程

从特定状态出发，开始执行蒙特卡洛树搜索。蒙特卡洛树搜索方法以该状态为根节点，维护维护一个树状结构。并反复执行选择、拓展、模拟、回溯 4 个步骤，从而对序贯决策的搜索空间进行探索，其维护的树状结构即可看作被探索过的状态的集合。

树中的各节点记录该节点被访问的次数以及该点的价值大小，并以此为依据计算体现节点优劣的置信上限 (Upper Confidence Bounds, UCB):

$$UCB(S_i) = \bar{V}_i + c \sqrt{\frac{\log N}{n_i}}, c = 2 \quad (1)$$

其中 $\bar{V}_i$ 表示节点的平均价值， N 、 n_i 分别表示总探索次数以及该节点的探索次数，在选择阶段，该方法将沿着树状结构选择 UCB 取值最大的节点，直至到达所维护树状结构的叶节点。

在选择一个较优的叶节点后，蒙特卡洛树将从该叶节点所代表的状态开始，选取一个可行的动作，以此拓展到下一个状态，并将该状态加入到自身维护的树状结构中。

进一步，在拓展节点的基础上，蒙特卡洛树进行一次模拟：不断随机执行新的动作，直到该序贯决策问题到达终点，并对此时的终点状态进行判定，计算该终点的价值。

最后，从拓展出的节点开始，蒙特卡洛树将模拟所得价值反向传播，更新拓展节点祖先路径中的所有节点。由于模拟过程中，节点采取的一系列动作是完全随机确定的，概率上，其值的大小反应了拓展所得状态的优劣。

随着搜索时间的延长，我们对根节点可采取的各项操作的优劣将会愈发准确，并得以以此为依据实现当前状态下的动作优选。以此往复，即可逐步确定整个序贯决策过程。

4.3 问题一

4.3.1 数学模型

通过对问题一的分析建立的规划调度模型如下：

$$\begin{aligned}
 \max \text{Payoff} &= \sum_{k=1}^{10} \sum_{t=8}^{22} -|dis_k(t) - dis_k(t-1)| C_k \\
 &+ I_{(power_k(t) - power_k(t-1) \geq 0)} (power_k(t) - power_k(t-1)) E_C(t-1) \\
 &+ I_{(power_k(t) - power_k(t-1) < 0)} (power_k(t) - power_k(t-1)) E_D(t-1) \\
 s.t. \left\{ \begin{aligned}
 &dis_k(7) = dis_k(22) \dots\dots\dots ① \\
 &power_k(7) = power_k(22) \dots\dots\dots ② \\
 &power_k(t_2) \leq power_k(t_1) \quad \forall 7 \leq t_1 \leq t_2 \leq t_3 \leq 22, \forall k \in s_k \dots\dots\dots ③ \\
 &power_k(t_2) \leq power_k(t_3) \quad \forall 7 \leq t_1 \leq t_2 \leq t_3 \leq 22, \forall k \in s_k \dots\dots\dots ④ \\
 &\Delta power_k \times \Delta power_{k'} = 0 \text{ if } dis_k(t) = dis_{k'}(t), \forall k, k' \in s_k \dots\dots\dots ⑤ \\
 &P_{kn} + P_n = P_{n_{需}} \dots\dots\dots ⑥ \\
 &P_{kn} \leq P_{k_{额}} \dots\dots\dots ⑦ \\
 &\overline{P_{kn}} \leq P_{n_{需}} \dots\dots\dots ⑧ \\
 &\beta_n \leq 1 \dots\dots\dots ⑨ \\
 &\beta_{n'} \leq 1 \text{ if } \beta_{n''} \leq 1 \dots\dots\dots ⑩ \\
 &\beta_{n''} \leq 1 \text{ if } \beta_{n'''} \leq 1 \dots\dots\dots ⑪ \\
 &7 \leq t \leq 22 \dots\dots\dots ⑫ \\
 &t \in \mathbb{Z} \dots\dots\dots ⑬
 \end{aligned} \right. \quad (3)
 \end{aligned}$$

其中， $s_k = \{1, 2, \dots, 10 \mid \text{type}(1, 2, \dots, 5) = \text{MV\#1}, \text{type}(6, 7, \dots, 10) = \text{MV\#2}\}$ 表示储能车集合，其中编号 1-5 的车辆类型为 MV#1，编号 6-10 的车辆类型为 MV#2。 $dis_k(t)$ 表示 t 时刻第 k 个移动储能车的位置， $|dis_k(t) - dis_k(t-1)|$ 表示第 k 个储能车在 $t-1$ 时刻的位置到 t 时刻位置之间的距离。 C_k 表示第 k 辆移动储能车的单位里程耗费。 $E_C(t)$ 表示 t 时刻的充电电价， $E_D(t)$ 表示 t 时刻的放电电价。 $I_{(\sim)}$ 表示示性函数，当括号内的条件被满足时其等于 1，否则等于 0。 $\Delta power_k = power_k(t-0.5) - power_k(t)$ 。 $\tilde{P}_{kn}$ 表示第 k 量车对第 n 号台区变的输出功率。 $P_{n_{需}}$ 表示第 n 号台区变的用电需求功率。

目标函数由行驶路程带来的资源消耗： $-|dis_k(t) - dis_k(t-1)|C_k$ 、充电电费： $I_{(power_k(t)-power_k(t-1))\geq 0}(power_k(t) - power_k(t-1))E_C(t-1)$ 、放电利润： $I_{(power_k(t)-power_k(t-1))< 0}(power_k(t) - power_k(t-1))E_D(t-1)$ 构成。

约束条件 1 和 2 表示车辆在 7 时与 22 时都在规定地点且电量都为满电，约束条件 3 表示车辆仅完整充放电一次，约束条件 4 表示充电站或台区变只能服务一台移动储能车或被一台移动储能车服务。约束条件 5 表示移动储能车和台区变满足用电需求。约束条件 6 表示车辆输出功率不能超过车的额定功率。约束条件 7 表示车辆的输出功率应小于车的额定功率。约束 8 表示为满足各类用户需求，各个台区变负载率都应小于 1。约束 9 表示若二类用户被满足则一类用户一定被满足，三类用户被满足则二类用户一定被满足，以此约束满足的优先级。约束 10 表示移动储能车的活动时间为 7-22 时，所以也仅考虑这个期间的调度情况，并且车辆的调度间隔为 1 小时所以我们仅考虑车辆在整点时的状态改变。

4.3.2 算法设计

针对上述数学模型，我们拟采用蒙特卡洛树搜索算法实现求解。蒙特卡洛树搜索算法拟解决序贯决策问题，然而在本问题中，对移动储能车的调度实际上是时间对储能车状态的映射。我们可以用蒙特卡洛树中的节点表示移动储能车自身的状态，并将其在每个整点时刻的调度作为动作。然而，这会导致储能车无法有效捕获全局状态。例如：当储能车在特定台区变节点放电时，其他储能车无法同时在此工作，若对单一储能车独立求解，会导致其在规划调度时忽视此类情况，从而出现无意义的排队等情况。

为解决上述对单一储能车独立求解带来的窘境，如图 4 左，我们采用全局状态求解。保留每一辆储能车之间的状态、动作变化带来的相互影响。为实现全局求解，我们将同一时刻的模型状态共享，并将同一时刻不同储能车的动作关联，从而将不同的储能车连在一起。

如图 4 右，序贯决策过程拓展到不同储能车之间。每辆储能车的动作为整个模型生成一个新的状态，并由下一辆储能车执行接下来的动作。当所有储能车完成一轮动作，调度迎来新的整点时刻，进入下一阶段。由此，我们得以将全局信息连成一片，在搜索中捕获整个模型的影响。同时，这种处理方案也存在一定的问题，由于决策过程完全串联，将导致搜索深度过大，实际搜索空间随之扩张，容易陷入局部最优解。

进一步，我们结合独立建模，实现垂直探索求解，从而提升搜索效率。该内容将在 4.3.3 节有所体现

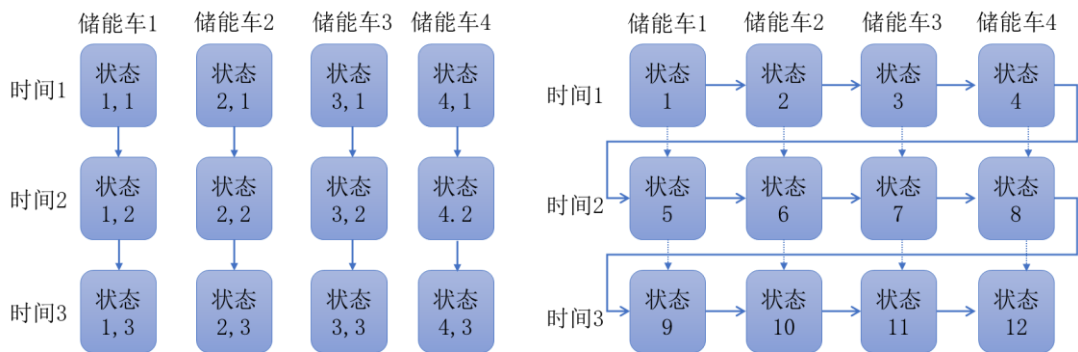


图4 状态转移流程

为实现蒙特卡洛树搜索，现对其关键要素**状态**、**动作**、**价值函数**进行定义：

状态

本文拟实现问题的全局状态求解，为此，状态所记录的信息为问题通用的全局信息，是所有储能车的共处系统环境。为便于说明调度情况，本文的状态表示整点时刻的系统信息。其数据如表 3 所示：

表3 状态描述

数据	描述
储能车位置	以 10 个整数描述每个储能车所在节点
储能车电量	以 10 个整数描述每个储能车剩余电量
充电站排队情况	以 4 个队列结构描述充电站的占用情况
台区变排队情况	以 4 个队列结构描述台区变的接入情况
时间	以整数形式记录当前所处整点时刻
待动作储能车	当前状态需进行动作的储能车序号

动作

为推动状态转移，对动作的定义必不可少。在特定状态下，本文采用的方案将同一时刻的不同储能车细化，故动作的进行也将驱使状态中的“待动作储能车”变化，即下一个状态应按顺序由下一辆储能车进行动作。当所有储能车完成一轮动作后，时间刻向前推动一个单位。

此外，动作包含了原地等待、放电（台区变）、充电（充电站）、移动。其参数描述和可行场景情况如表 4 所示：

表4 动作描述

动作	场景	参数
原地等待	所有状态	待动作储能车序号
放电	待动作储能车位于台区变节点且满足放电条件	待动作储能车序号、放电功率
充电	待动作储能车位于充电站节点且满足充电条件	待动作储能车序号、充电功率
移动	所有状态	待动作储能车序号、目标节点

上述动作表格给出基础的适用场景，能够初步实现模型优化求解。然而，在后续实际搜索过程中，出现搜索空间过大，搜索结果难以收敛等情况。为进一步缩小搜索空间，本文将在 4.3.3 节中通过垂直探索求解给出高效的剪枝方案。

价值函数

$$reward = \omega \cdot payoff \quad (4)$$

这里， ω 表示惩罚系数，代表调度方案满足约束条件的程度。对惩罚系数的确定，我们给出如下方案：

在本问题中，要求(1)所有储能车于每天 22 时前返回车辆集合存放点；(2)返回车辆集合存放点的储能车应为满电状态；(3)按照优先级保证三类供电地区台区变负载率不超

过 1。(4)储能车仅能完全充放电一次；(5)台区变及充电站同一时间段只能服务同一台储能车。

在上述约束中，约束(4)(5)可通过限制不同状态下的可取动作来实现。针对约束(1)(2)(3)，本求解方法分别记录 22 时未能返回集合存放点的储能车数目、22 时未能充满电的储能车数目、各区域未保证负载率的小时数，记上述数目为 r 。

并根据优先级计算对应部分惩罚系数，如图 5 所示。其中颜色越倾向于绿色表示惩罚的优先级越高，越倾向于黄色表示优先级越低。由图可知，针对高优先级的惩罚系数，对于相同的记录数值 r ，更高优先级的惩罚项（一类地区不满足负载率的小时数、22 时无法回到存放点、22 时不是满电状态）将返回相对更低的取值，给予相对更强的惩罚，中（二类地区不满足负载率的小时数）、低优先级（三类地区不满足负载率的小时数）取值相对较高。

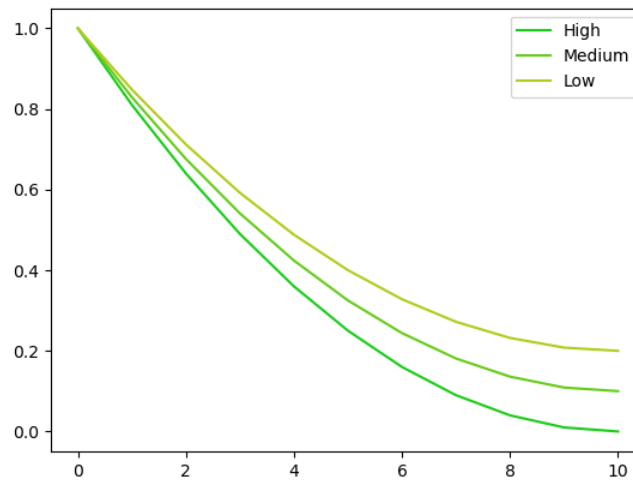


图5 惩罚系数

此外，惩罚项函数采用 $p_i = 0.01 \cdot \alpha_i (r_i - 10)^2 + (1 - \alpha_i)$ 形式，其中 α 在高、中、低优先级情况下分别取 1, 0.9, 0.8，针对上述五类不满足约束情况，求得 $\omega = \prod p_i$ 。

4.3.3 结果与分析

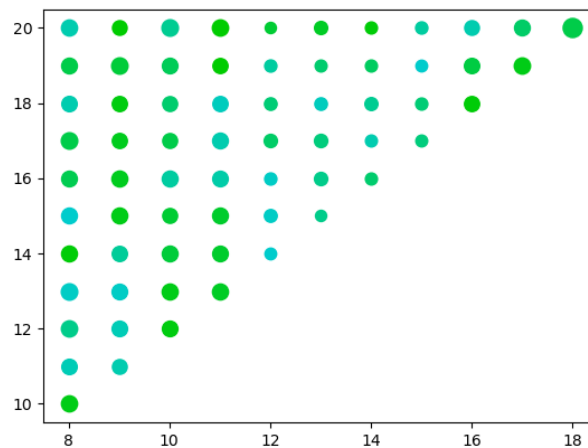


图6 个体储能车建模与优化

本文采用垂直求解的方式实现全局状态搜索下的快速剪枝。

首先我们设定单独的一辆储能车求解，车辆状态为自身位置、剩余电量和当前时间；车辆可选动作为移动、充电、放电、静止；其价值函数为车辆自身带来的服务利润。经过优化，我们记录优化后储能车倾向的充放电时间，如图 6 所示。其中，横坐标表示开始放电的时间点，纵坐标表示开始充电的时间点。散点图中，散点大小代表优化后的储能车选择在横坐标时刻放电、在纵坐标时刻充电的次数。可见随着优化的进行，储能车倾向于在 8-12 时、17-20 时放电，在 13-16 时、18-20 时充电。另外，散点颜色对应这些时刻充放电的平均收益，绿色散点收益较高，蓝色散点收益较低。收益值的大小与优化后储能车在不同状态的比例大小相符。

此外，由于储能车只能完全充放电 1 次，故充电后再次前往台区变节点是无意义的，且会带来额外路程开销。同理，若未充满电，中途返回集合存放点也是额外开销的来源。

由此，可对动作选择条件增添如下限制以缩减动作取值范围，进而实现搜索空间剪枝：

- 若当前时间处于高电价区间（8-12 时、17-20 时），对于位于台区变节点的储能车，若其仍有超过 10% 的电量，不可以向充电站或集合存放点移动；
- 位于集合存放点，且未充电（充电代表当天任务结束）的储能车，仅可执行向台区变节点移动、进行放电或原地静止的动作；
- 未充电的储能车不能执行前往集合存放点移动的动作；
- 位于充电站的储能车仅可执行充电、向集合存放点移动或原地静止的动作。

这系列约束将储能车的一日任务流程限制为出发、放电、充电、返程，且在用电高峰期会出现向台区变聚集的趋势。

最终优化结果如表 5 所示，实现最大利润时，十辆移动储能车的调度方案如下表所示：

表5 问题一调度方案

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
7 点	1,200,M	1,200,M	1,200,M	1,200,M	1,200,M	1,400,M	1,400,M	1,400,M	1,400,M	1,400,M
8 点	27,200,O	19,200,S	19,200,O	14,200,O	5,200,O	27,400,S	27,400,S	5,400,S	14,400,S	5,400,S
9 点	27,175,S	19,200,S	19,150,S	14,150,S	5,150,M	27,400,S	27,400,O	5,400,S	14,400,S	5,400,O
10 点	27,175,S	19,200,S	19,150,S	14,150,S	14,150,S	27,400,O	27,350,S	5,400,S	14,400,S	5,300,O
11 点	27,175,S	19,200,O	19,150,S	14,150,O	14,150,S	27,350,S	27,350,O	5,400,S	14,400,S	5,200,O
12 点	27,175,S	19,150,O	19,150,S	14,100,S	14,150,O	27,350,S	27,250,O	5,400,S	14,400,S	5,100,O
13 点	27,175,S	19,100,S	19,150,S	14,100,O	14,100,S	27,350,S	27,150,O	5,400,O	14,400,S	5,0,S
14 点	27,175,S	19,100,O	19,150,S	14,50,O	14,100,S	27,350,O	27,50,S	5,300,O	14,400,S	5,0,S
15 点	27,175,S	19,50,S	19,150,O	14,0,S	14,100,S	27,250,S	27,50,O	5,250,O	14,400,O	5,0,S
16 点	27,175,S	19,50,O	19,100,S	14,0,S	14,100,S	27,250,O	27,0,S	5,200,O	14,300,O	5,0,S
17 点	27,175,S	19,0,M	19,100,O	14,0,S	14,100,S	27,150,O	27,0,S	5,100,O	14,200,O	5,0,S
18 点	27,175,O	13,0,I	19,50,O	14,0,M	14,100,M	27,50,O	27,0,M	5,0,M	14,100,O	5,0,M
19 点	27,125,M	13,200,S	19,0,M	13,0,I	19,100,I	27,0,M	6,0,I	6,0,S	14,0,M	22,0,I
20 点	22,125,I	13,200,S	22,0,I	13,200,S	19,200,S	18,0,I	6,400,S	6,0,I	13,0,I	22,400,S
21 点	22,200,M	13,200,M	22,200,M	13,200,M	19,200,M	18,400,M	6,400,M	6,400,M	13,400,M	22,400,M
22 点	1,200,S	1,200,S	1,200,S	1,200,S	1,200,S	1,400,S	1,400,S	1,400,S	1,400,S	1,400,S

车辆在每个时刻的方案由三元组构成（位置，剩余电量，状态），表格中蓝色(S)表示车辆停靠，红色(O)表示车辆放电中，绿色(I)表示车辆充电，黄色(M)表示车辆移动。车辆在各个停靠点之间的移动路径遵循附录 A 中的路径方案。

根据表中的方案，计算得利润为 2847.07，且所有台区变均未出现超载情况，三类用户用电情况均得到有效保证。

4.4 问题二

4.4.1 数学模型

通过分析问题二，建立数学模型如下：

$$\max \text{payoff} \quad (5)$$

$$\min |set_k| \quad (6)$$

$$\left\{ \begin{array}{l} dis_k(7) = dis_k(22) \dots\dots\dots ① \\ power_k(7) = power_k(22) \dots\dots\dots ② \\ \Delta power_k \times \Delta power_{k'} = 0 \text{ if } dis_k(t) = dis_{k'}(t), \forall k, k' \in s_k \dots\dots\dots ③ \\ P_{kn} + P_n = P_{n\text{需}} \dots\dots\dots ④ \\ P_{kn} \leq P_{k\text{额}} \dots\dots\dots ⑤ \\ P_{kn} \leq P_{n\text{需}} \dots\dots\dots ⑥ \\ \beta_n \leq 1 \dots\dots\dots ⑦ \\ \beta_{n'} \leq 1 \text{ if } \beta_{n''} \leq 1 \dots\dots\dots ⑧ \\ \beta_{n''} \leq 1 \text{ if } \beta_{n'''} \leq 1 \\ 7 \leq t \leq 22 \dots\dots\dots ⑨ \\ t \in \mathbb{Z} \end{array} \right. \quad (7)$$

其中, $s_k = \{1, 2, \dots, 10 | \text{type}(1, 2, \dots, 5) = \text{MV\#1}, \text{type}(6, 7, \dots, 10) = \text{MV\#2}\}$ 表示储能车集合，其中编号 1-5 的车辆类型为#MV1，编号 6-10 的车辆类型为#MV2。 $dis_k(t)$ 表示 t 时刻第 k 个移动储能车的位置， $|dis_k(t) - dis_k(t-1)|$ 表示第 k 个储能车在 $t-1$ 时刻的位置到 t 时刻位置之间的距离。 C_k 表示第 k 辆移动储能车的单位里程耗费。 $E_C(t)$ 表示 t 时刻的充电电价， $E_D(t)$ 表示 t 时刻的放电电价。 $I_{(\sim)}$ 表示示性函数，当括号内的条件被满足时其等于 1，否则等于 0。 $\Delta power_{k_i} = power_{k_i}(t-0.5) - power_{k_i}(t)$ $\tilde{P}_{kn}$ 表示第 k 量车对第 n 号台区变的输出功率。 $P_{n\text{需}}$ 表示第 n 号台区变的用电需求功率。

题目中对车辆调度的目标为使用最少数量移动储能车，但由于移动储能车的类型以及光伏发电的存在，在最少数量的车辆供给下，不同的车辆类型组合也会导致方案不同，将利润作为目标函数，再次筛选具有相同数量不同组合类型的方案，以达到动用最少移动储能车获取相对较高利益的情况。

由于第二题允许车辆进行多次充放电操作，所以约束条件相较于第一题有所降低。关于车辆仅允许充放电一次的约束条件删除其余条件与第一题保持不变。**需要注意的是：**此时， E_C 由两部分组成：一部分为快速充电的电价，另一部分为光伏发电的电价。

4.4.2 算法设计

针对本问题，对蒙特卡洛树搜索过程以及仿真算法进行优化与修正。

在问题 1 动作选择限制的基础上，问题 2 取消了对充电次数的限制。在问题 1 中，经过完成充电后的储能车垂直探索求解，动作选择条件更新为 4.3.4 所示，当储能车完

成自身充电，其只能选择返回车辆集合存放点，或原地待命，择机返回集合存放点。在本问题中，充电完成的储能车仍能前往台区变进行放电，故本问题中动作选择条件放宽了对位于充电站的节点的限制，更新后的动作选择条件为：

- 若当前时间处于高电价区间（8-12 时、17-20 时），对于位于台区变节点的储能车，若其仍有超过 10% 的电量，不可以向充电站或集合存放点移动；
- 位于集合存放点，且未放电的储能车，仅可执行向台区变节点移动、进行放电或原地静止的动作；
- 未放电的储能车不能执行前往集合存放点移动的动作；

除此之外，根据增设光伏发电装置，储能车充电时能量来源由单一的充电站提供，变为充电站与光伏设备共同提供。廉价的光伏发电设备为储能车运行成本的进一步缩减提供了可能：

$$P_{\text{充}} = P_{m\text{实}} + P_{\text{光}}$$

具体开销相关信息由仿真过程中实体自身记录，并于仿真结束汇总，详见附录 B。

为此，本问题中，我们首先将惩罚系数作为价值函数进行优化，以尽可能减少负载率不满足要求的情况。进一步，在确定最少所需的储能车数目后，我们将针对服务利润进行优化，以求得储能车最优调度方案。

4.4.3 结果与分析

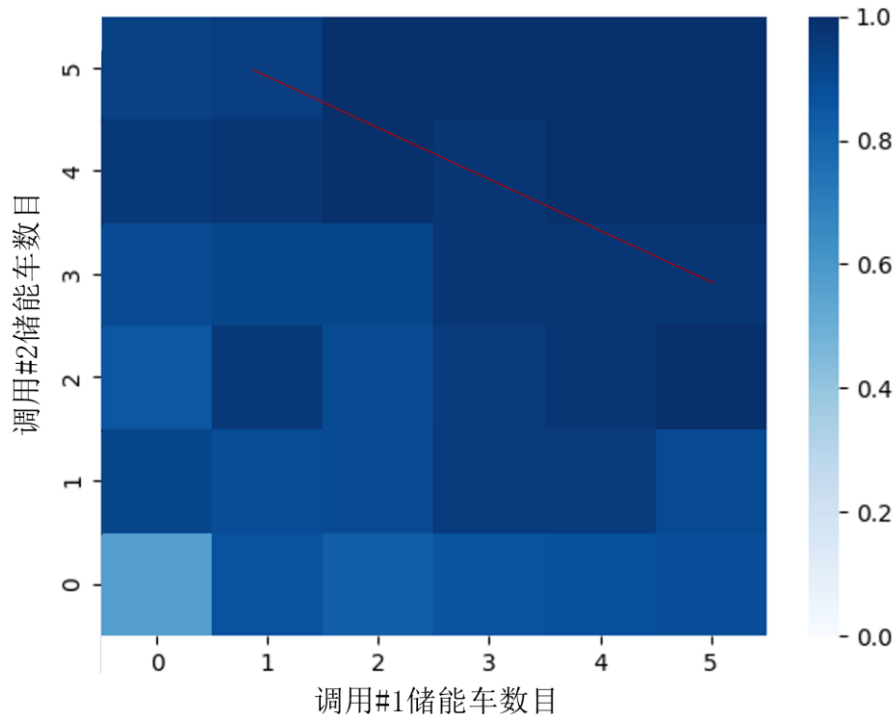


图7 惩罚系数变化图

我们首先计算了调用不同数目的 MV#1、MV#2 类储能车的情况下粗劣优化得到的惩罚系数，可见在直线 $n_2 = 6 - 0.5n_1$ 附近时，惩罚系数能够达到 0.9 水平，说明这部分储能车分配方案有较大潜力实现满足各区域负载的情况。

由此，我们进一步对如下分配方案进行优化求解：{<1, 5>, <2, 5>, <3, 4>, <4, 4>, <5, 3>}，并最终得到当调用 1 辆 MV#1 类储能车、5 辆 MV#2 类储能车时，能够完全满

足三个区域的负载率要求，如表 6 所示调度方案，此时服务利润 480.8。故对储能车的最少需求量为 6 辆。

表6 问题二调度方案

	#6	#7	#8	#9	#10	#1
7 点	1,400,M	1,400,M	1,400,M	1,400,M	1,400,M	1,200,M
8 点	5,400,O	5,400,S	27,400,S	14,400,S	14,400,S	19,200,S
9 点	5,300,O	5,400,S	27,400,S	14,400,S	14,400,S	19,200,S
10 点	5,200,O	5,400,S	27,400,S	14,400,S	14,400,S	19,200,S
11 点	5,100,O	5,400,S	27,400,O	14,400,O	14,400,S	19,200,O
12 点	5,0,M	5,400,O	27,300,O	14,350,O	14,400,S	19,150,O
13 点	6,0,I	5,300,O	27,200,O	14,300,O	14,400,S	19,100,S
14 点	6,400,M	5,200,O	27,100,M	14,250,M	14,400,O	19,100,O
15 点	27,400,O	5,150,O	14,100,O	19,250,O	14,350,S	19,50,S
16 点	27,350,O	5,100,O	14,0,M	19,200,O	14,350,O	19,50,S
17 点	27,300,O	5,50,O	18,0,I	19,150,O	14,250,O	19,50,S
18 点	27,250,O	5,0,M	18,400,S	19,100,M	14,150,O	19,50,O
19 点	27,200,M	13,0,I	18,400,S	22,100,I	14,50,M	19,0,M
20 点	6,200,I	13,400,S	18,400,S	22,400,S	13,50,I	18,0,I
21 点	6,400,M	13,400,M	18,400,M	22,400,M	13,400,M	18,400,M
22 点	1,400,S	1,400,S	1,400,S	1,400,S	1,400,S	1,200,S

进一步，我们将服务利润作为价值函数，在仅派出 6 辆车的前提下，寻求利润最优，计算可得如表 7 所示调度方案：

表7 问题 2 利润优选调度方案

	#1	#2	#3	#4	#5	#6
7 点	1,400,M	1,400,M	1,400,M	1,400,M	1,400,M	1,200,M
8 点	5,400,O	5,400,S	27,400,S	14,400,S	14,400,S	19,200,S
9 点	5,300,O	5,400,S	27,400,S	14,400,S	14,400,S	19,200,S
10 点	5,200,O	5,400,S	27,400,S	14,400,S	14,400,S	19,200,S
11 点	5,100,O	5,400,S	27,400,O	14,400,O	14,400,S	19,200,O
12 点	5,0,M	5,400,O	27,300,O	14,350,O	14,400,S	19,150,O
13 点	6,0,I	5,300,O	27,200,O	14,300,O	14,400,S	19,100,S
14 点	6,400,M	5,200,O	27,100,M	14,250,M	14,400,O	19,100,O
15 点	27,400,O	5,150,O	14,100,O	19,250,O	14,350,S	19,50,S
16 点	27,350,O	5,100,O	14,0,M	19,200,O	14,350,O	19,50,S
17 点	27,300,O	5,50,O	18,0,S	19,150,O	14,250,O	19,50,S
18 点	27,250,O	5,0,M	18,0,S	19,100,M	14,150,O	19,50,O
19 点	27,200,M	13,0,I	18,0,I	22,100,I	14,50,M	19,0,M
20 点	6,200,I	13,400,S	18,400,S	22,400,S	13,50,I	18,0,I
21 点	6,400,M	13,400,M	18,400,M	22,400,M	13,400,M	18,400,M
22 点	1,400,S	1,400,S	1,400,S	1,400,S	1,400,S	1,200,S

该调度方案下利润为 533.47。

4.5 问题三

4.5.1 数学模型

问题三与问题一的优化模型相同：

$$\begin{aligned}
 \max \text{Payoff} = & \sum_{k=1}^{10} \sum_{t=8}^{22} -|dis_k(t) - dis_k(t-1)| C_k \\
 & + I_{(power_k(t) - power_k(t-1)) \geq 0} (power_k(t) - power_k(t-1)) E_c(t-1) \\
 & + I_{(power_k(t) - power_k(t-1)) < 0} (power_k(t) - power_k(t-1)) E_D(t-1) \\
 s.t. \left\{ \begin{aligned}
 & dis_k(7) = dis_k(22) \dots\dots\dots ① \\
 & power_k(7) = power_k(22) \dots\dots\dots ② \\
 & power_k(t_2) \leq power_k(t_1) \quad \forall 7 \leq t_1 \leq t_2 \leq 22, \forall k \in s_k \dots\dots\dots ③ \\
 & \Delta power_k \times \Delta power_{k'} = 0 \text{ if } dis_k(t) = dis_{k'}(t), \forall k, k' \in s_k \dots\dots\dots ④ \\
 & P_{kn} + P_n = P_{n\text{需}} \dots\dots\dots ⑤ \\
 & P_{kn} \leq P_{k\text{额}} \dots\dots\dots ⑥ \\
 & \overline{P_{kn}} \leq \overline{P_{n\text{需}}} \dots\dots\dots ⑦ \\
 & \beta_n \leq 1 \dots\dots\dots ⑧ \\
 & \beta_{n'} \leq 1 \text{ if } \beta_{n''} \leq 1 \dots\dots\dots ⑨ \\
 & \beta_{n''} \leq 1 \text{ if } \beta_{n'''} \leq 1 \\
 & 7 \leq t \leq 22 \dots\dots\dots ⑩ \\
 & t \in \mathbb{Z}
 \end{aligned} \right. \quad (9)
 \end{aligned}$$

其中， $s_k = \{1, 2, \dots, 10 | \text{type}(1, 2, \dots, 5) = MV \#1, \text{type}(6, 7, \dots, 10) = MV \#2\}$ 表示储能车集合，其中编号 1-5 的车辆类型为 MV#1，编号 6-10 的车辆类型为 MV#2。 $dis_k(t)$ 表示 t 时刻第 k 个移动储能车的位置， $|dis_k(t) - dis_k(t-1)|$ 表示第 k 个储能车在 $t-1$ 时刻的位置到 t 时刻位置之间的距离。 C_k 表示第 k 辆移动储能车的单位里程耗费。 $E_c(t)$ 表示 t 时刻的充电电价， $E_D(t)$ 表示 t 时刻的放电电价。 $I_{(\sim)}$ 表示示性函数，当括号内的条件被满足时其等于 1，否则等于 0。 $\Delta power_k = power_k(t-0.5) - power_k(t)$ 。 $\tilde{P}_{kn}$ 表示第 k 量车对第 n 号台区变的输出功率。 $P_{n\text{需}}$ 表示第 n 号台区变的用电需求功率。

目标函数由行驶路程带来的资源消耗： $-|dis_k(t) - dis_k(t-1)| C_k$ 、充电电费： $I_{(power_k(t) - power_k(t-1)) \geq 0} (power_k(t) - power_k(t-1)) E_c(t-1)$ 、放电利润： $I_{(power_k(t) - power_k(t-1)) < 0} (power_k(t) - power_k(t-1)) E_D(t-1)$ 构成。

上述模型在求解过程中，与第一问相比需要**增加条件**：需要考虑移动储能车移动时的拥堵情况、充电时的排队情况。

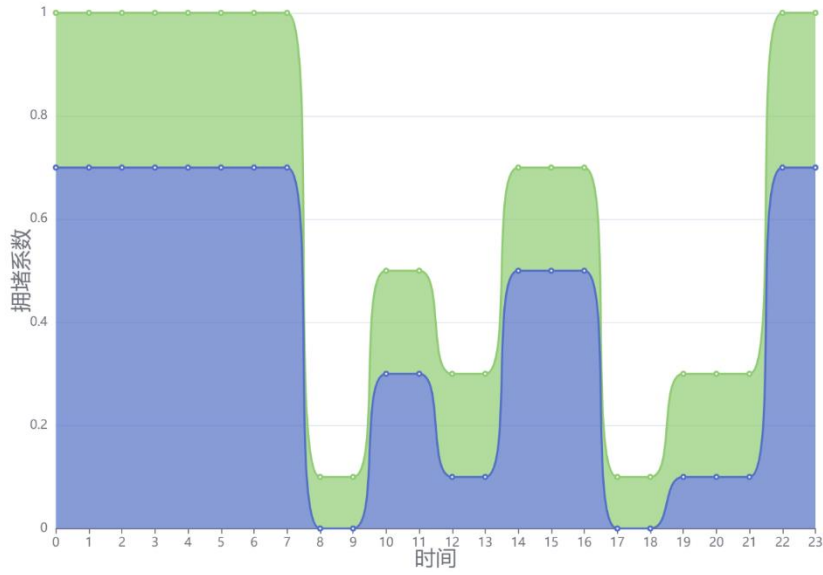


图8 拥堵情况可视化

我们将拥堵情况视为服从均匀分布随机变量。分布上下限为题中所给数据值。

4.5.2 算法设计

相比于第一问，移动储能车的充电出现排队现象，因此需要对排队逻辑进行设计：在该问中，我们给充电站对象新增了一个需要随时间遍历而维护的变量：`TimeQueue`，表示在当前时刻下，该充电站预计排队时长（单位为小时）。同时，仿真过程还需要输入四个充电站的泊松分布参数和一个负指数分布参数，具体的更新方式如下：

假设 7 点之前时，充电站没有排队情况。当当前时刻大于 7 点时，时间每遍历一个小时，所有充电站实体的 `TimeQueue-1`，表示已经过去一个小时时间。同时，根据充电站对应的泊松分布参数，获得接下来一小时内会到达充电站的车辆数量，并根据负指数分布得出这些车辆的排队等候时间 Δt ，则充电站实体的 `TimeQueue +=  $\Delta t$` 。

接下来，在每一时刻对充电车进行分析。对于某一台充电车 J ，若其在 i 时刻位于充电站实体 q ，且将在 q 处充电，则根据该实体维护的 `q.TimeQueue` 变量，充电车充满电所需电量 ΔE ，可以确定充电车 J 的开始充电时间 $Start$ 和结束充电时间 End

$$Start = i + q.TimeQueue$$

$$End = Start + \frac{\Delta E}{400}$$

这里， $Start$ 和 End 是精确的，但不一定是整数。因此，设在 $Leave$ 时刻 ($Leave$ 是一个整数)，充电车 J 还未完成充电，但在 $Leave + 1$ 时刻，充电车 J 已经充满电，同时注意到，充电站充电功率为 400KW，即充电车的充电时间不会多于一小时。因此，如图所示，在各种情况下，都有：

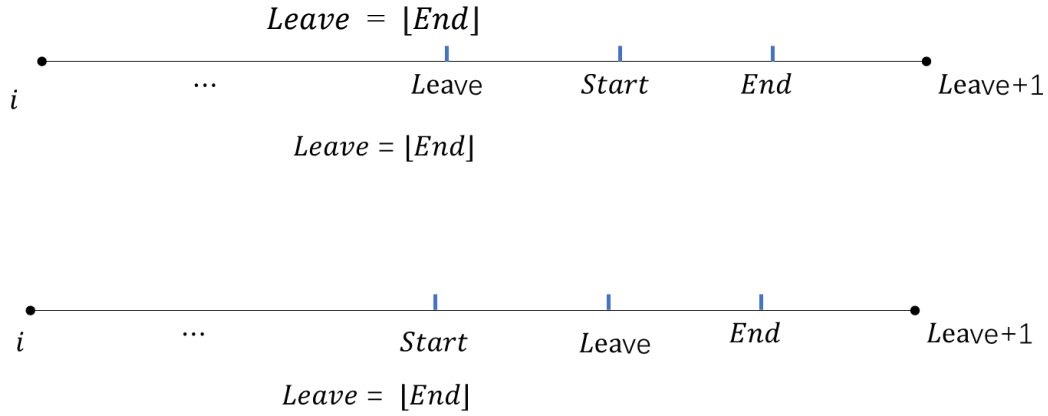


图9 储能车充电等待示意图

接下来，仿真程序需要考虑充电车 J 充完电后能否在晚上 22 点前返回终点。根据附件 3 中的数据，可知充电站 q 到节点 1 的最短距离 $Distance$

设充电车 J 充完电后立刻开始移动，在 22 点前，其最多可以移动的距离为 $AbleDistance$

同时，根据题设所给的道路拥堵情况，可知各时段内的充电车平均速度 $Speed$ 。因此， $AbleDistance$ 可根据下列伪代码所示逻辑表达：

```

AbleDistance = 0;
AbleDistance = Speed[leave] * (Leave + 1 - End)
For i ← Leave + 1 to 22
AbleDistance += Speed[i] * 1

```

如附录三中介绍的数据结构所示，针对问题三给出的场景，我们给充电站对象增加了变量 $ChargeList$ 和 $chargeCar1, chargeCar2$ 三个变量

$Chargelist$ 为充电站的全局充电车对象顺序 当有车辆到达充电站想要充电时，车辆的 ID 会被加入 $Chargelist$ 的最后，而当车辆完成充电时，则其 ID 讲从 $Chargelist$ 中取出。

在保证同时刻只为一个对象充电的前提下，充电站在一小时内可以为两台小车或一台大车充满电，因此，我们设 $chargeCar$ 表示充电站在前半小时的充电对象类型。 $chargeCa1$ 为 0 说明该时刻的前半小时充电站无充电对象，任意车都可以按照先到先服务的原则在该充电站充电；当 $chargeCar1$ 为 1 说明该时刻的前半小时充电站在为大车充电，则该时段（一小时内）充电站无法为其他车辆充电；当 $chargeCar1$ 为 2 说明该时刻的前半小时充电站在为小车充电，则该时段内充电站还能为一辆小车充电，但无法为大车充电。

$Is_chargeCar$ 表示该时刻内是否有车在台区变节点处放电， $carCharge$ 为该时刻车辆对台区变的放电量；若该时刻无车辆在台区变处放电，则 $carCgarfe$ 为 0。

在得到 $AbleDistance$ 后，与实际距离 $Distance$ 进行对比，从而获知在当前方案和随机数序列下，充电车 J 无法在 22 点前返回终点。

最终，返回值和各区域超载次数、到 22 点未满足电车辆数量、充电后无法暗示返回数量有关。考虑到随机数的影响，程序的返回值为相同输入情况下循环 100 次的 $Reward$ 平均值。

4.5.3 结果与分析

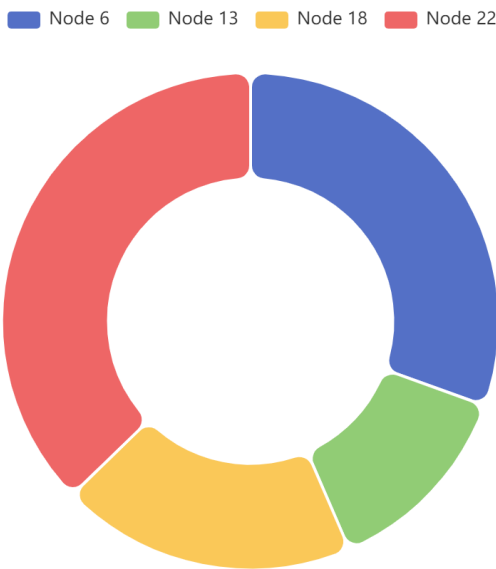


图10 充电占比图

根据第一问计算时产生的模拟数据，各充电站使用率占比约为图 9 所示。为此，我们特别选取其中适用占比最高的两个充电站节点作为本问题的研究对象。我们取节点 22、节点 6。假设所有车所需充电时长满足同一个负指数分布，并设置默认参数为 0.05min^{-1} ，假设每个台区变各时刻所需负载满足均值为第一题设置条件，方差为 0.02 的正态分布。

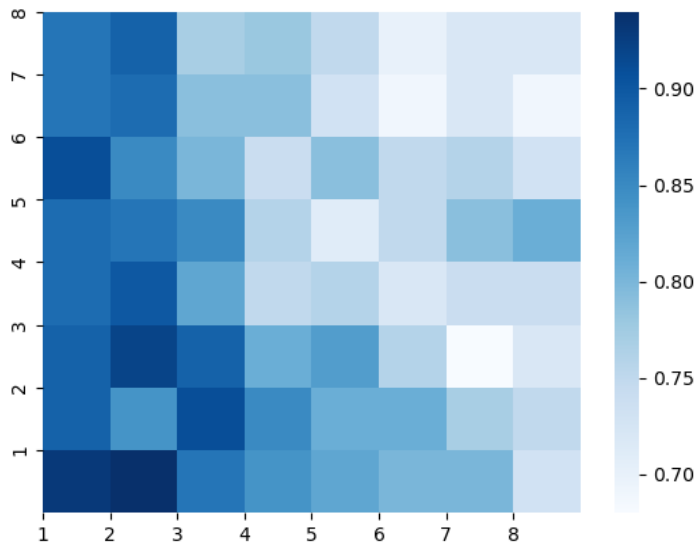


图11 惩罚系数分布

由此对节点 6、22 取不同的泊松分布参数，求解所得最优惩罚系数如图 11 所示，其中横轴节点 6 对应的参数取值，纵轴为节点 22 所对应的参数取值。该图中，随着节点 6、22 处的泊松分布参数增大，二者求解得到的惩罚系数均产生明显的降低，说明单个充电站排队情况加剧会导致全局约束更加难以实现。

同时提升相同的参数取值，6 号充电站对应的情况会产生更强烈的惩罚系数跌损情况，这也能够说明二者之间，6 号充电站起到更加重要的作用。

接下来，我们取特定参数下的情况进行求解，以展示本方法在随机条件下的优化水平。

这里，取节点 6 充电站的到达间隔服从均值为 8 的泊松分布，其他个节点的到达时间间隔服从均值为 2 的泊松分布，求解所得最有调度方案如表 8 所示。

表8 问题 3 最有调度方案

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
7 点	1,200,M	1,200,M	1,200,M	1,200,M	1,200,M	1,400,M	1,400,M	1,400,M	1,400,M	1,400,M
8 点	27,200,S	19,200,S	19,200,O	14,200,S	14,200,S	27,400,O	27,400,S	5,400,O	14,400,O	5,400,S
9 点	27,200,S	19,200,O	19,150,S	14,200,S	14,200,S	27,300,O	27,400,S	5,300,O	14,350,O	5,400,S
10 点	27,200,S	19,150,O	19,150,M	14,200,O	14,200,M	27,250,M	27,400,O	5,200,M	14,250,M	5,400,O
11 点	27,200,S	19,100,O	27,150,M	14,175,S	5,200,M	14,250,O	27,300,M	19,200,M	5,250,M	5,300,O
12 点	27,200,O	19,50,S	14,150,S	14,175,S	14,200,S	14,150,O	14,300,S	5,200,O	19,250,O	5,200,S
13 点	27,150,S	19,50,S	14,150,S	14,175,S	14,200,S	14,50,M	14,300,O	5,100,O	19,150,O	5,200,S
14 点	27,150,M	19,50,S	14,150,M	14,175,S	14,200,M	14,50,M	14,200,O	5,0,M	19,50,M	5,200,S
15 点	14,150,S	19,50,M	19,150,M	14,175,M	19,200,M	18,50,Q	14,100,O	13,0,Q	19,50,M	5,200,M
16 点	14,150,O	27,50,O	27,150,S	27,175,S	27,200,S		14,0,M		19,50,M	5,200,M
17 点	14,100,O	27,0,M	27,150,S	27,175,S	27,200,S		6,0,Q		13,50,Q	27,200,O
18 点	14,50,O	27,0,M	27,150,M	27,175,M	27,200,O					27,100,O
19 点	14,50,M	27,0,M	27,150,M	27,175,M	27,150,M					27,0,M
20 点	14,50,M	18,0,Q	22,150,Q	22,175,Q	27,150,M					6,0,Q
21 点	22,50,Q				22,150,Q					
22 点	1,50,U	1,400,H	1,400,H	1,400,H	1,150,U	1,200,H	1,200,H	1,200,H	1,200,H	1,200,H

当移动储能车开始排队后，其能否返回出发点、返回过程的路途成本等信息可以通过以用的充电站排队时间、剩余电量、节点间距、堵车稀疏求出，因此，当移动储能车进入排队序列后（浅绿色），仿真程序直接输出相应计算结果（体现在 22 点 U：未能返回，H：得以返回），不再对移动储能车进行仿真。

共有两台车来不及返回出发点，利润为 2475.46

可以发现，当存在排队、交通堵塞等情况时，会存在储能车来不及充满电返回出发地点的情况，且在调度过程中，存在部分车辆从 A 到达 B 之前，会利用当时堵塞程度较低的路况，先行抵达离 B 更近的 C 的情况，例如 3 号车在 11 点时刻的动作

5 模型评价

5.1 模型的优点

- (1) 模型充分结合实际，通过考虑移动储能车的调度周期，调度节点，光伏发电板的作用等因素，建立了合理的数学模型，便于应用求解。
- (2) 模型充分考虑了移动储能车的各项参数，以及快速充电站、台区变的功率。将各项约束用公式化的语言表达便于模型的泛化。
- (3) 本文使用的蒙特卡洛树搜索，把资源放在更值得搜索的分枝上，即算力集中在更有价值的地方，对于求解序贯决策、组合优化问题具有优良的性能。
- (4) 本文得到的移动储能车调度方案，具有在满足各类用户的需求下，达成最高利润或动用车辆数量最少的特点。除此之外，通过较高次数的模拟，本文应用求解算法能够很好的面对调度过程中的不确定性。

5.2 模型的不足

- (1) 由于外罚函数法可能导致求出的结果对于约束不能严格满足。此时求解出的方案虽然利润较高，但实际上不可用。
- (2) 由于时间限制，我们的算法效率还不够高，仿真过程中也存在一些冗余逻辑。同时，问题解空间的维度还有待进一步的优化。

参考文献

- [1] 陈燕, 于放, 田月,等. 基于 Hadoop 的车辆调度算法优化及应用[J]. 计算机系统应用, 2018.
- [2] 季辉, 丁泽军, 卞,等. 双人博弈问题中的蒙特卡洛树搜索算法的改进[J]. 计算机科学, 2018, 45(1):4.

附 录

附录 A: 可停靠点之间的最短路径

路 径 序 号	可 停 靠 点 对	具体路径	路 径 序 号	可 停 靠 点 对	具体路径
1	1-6	1-2-5-6	19	1-14	1-4-5-6-9-24-14
2	5-6	5-6	20	6-14	6-9-24-14
3	1-13	1-4-5-6-10-13	21	13-14	13-14
4	6-13	6-10-13	22	5-14	5-6-9-24-14
5	5-13	5-6-10-13	23	1-19	1-4-7-21-19
6	1-18	1-4-7-21-19-18	24	6-19	6-5-4-7-21-19
7	6-18	6-5-4-7-21-19-18	25	13-19	13-10-9-8-7-21-19
8	13-18	13-14-20-25-26-27-18	26	18-19	18-19
9	5-18	5-4-7-21-19-18	27	5-19	5-4-7-21-19
10	14-18	14-20-25-26-27-18	28	14-19	14-24-23-22-21-19
11	1-22	1-4-7-21-22	29	1-27	1-4-7-21-22-27
12	6-22	6-5-8-23-22	30	6-27	6-5-8-23-26-27
13	13-22	13-14-24-23-22	31	13-27	13-14-20-25-26-27
14	18-22	18-27-22	32	18-27	18-27
15	5-22	5-8-23-22	33	22-27	22-27
16	14-22	14-24-23-22	34	5-27	5-8-23-26-27
17	19-22	19-21-22	35	14-27	14-20-25-26-27
18	1-5	1-2-5	36	19-27	19-18-27

附录 B: 关键数据结构与程序伪代码

表9 问题一计算车辆收益伪代码

输入: cars[10], Station[4], ChargeStation[4], Map[29][29]

输出: rewards

```
1   For i←7 to 22 ▷ 按时间遍历
2       For j←1 to 10 ▷ 对每一台车分别遍历
3           If cars[j] output Energy at Station[q] ▷ 放电
4               If cars[j].Power[i] > cars[j].energy ▷放电多于余电
5                   Station[q].CarCharge ← Energy ▷ 向台区变放电
6                   Update cars[j].cost
7           Else
8               Station[q]. CarCharge ←Power[i] - cars[j].energy
9               Update cars[j].cost
10          EndIf
11      ElseIf cars[j] is charging at ChargeStation[q] ▷ 充电
12          Update cars[j].cost and cars[j].energy
13      ElseIf cars[j] is moving▷ 移动
14          Distance←Map[cars[j].currentPos][cars[j].NextPos]
15          Update cars[j].cost according to Distance
16      EndIf
17      If( i == 22)
18          If cars[j].energy != cars[j].initEnergy
19              Num_uncharged++
20          EndIf
21      EndIf
22  EndFor
23  For j← 1 to 4▷ 检查各台区变是否超载
24      If Station[j].LoadFactor > 1
25          If Station[j].priority == 1
26              Num_1++
27          ElseIf Station[j].priority == 2
28              Num_2++
29          ElseIf Station[j].priority == 3
```

```

20         Num_3++
31     EndIf
32 EndIf
33 EndFor
34 EndFor
35 Return Rewards(Cost,num_uncharged,Num_1,Num_2,Num_3)

```

表10 问题二计算车辆收益伪代码

输 入 :NumCar,cars[NumCar],Station[4],ChargeStation[4],Map[29][29],
CarType[NumCar]

输出: rewards

```

1  For i←7 to 22 ▷ 按时间遍历
2  For j←1 to NumCar ▷ 对每一台车分别遍历
3  If cars[j] output Energy at Station[q] ▷ 放电
4  If cars[j].Power[i] > cars[j].energy ▷放电多于余电
5  Station[q].CarCharge ← Energy ▷ 向台区变放电
6  Update cars[j].cost
7  Else
8  Station[q]. CarCharge ←Power[i] - cars[j].energy
9  Update cars[j].cost
10 EndIf
13 ElseIf cars[j] is charging at ChargeStation[q] ▷ 充电
14 If chargeStation[q].SunPower[i] > cars[j].InitEnergy – cars[j].setEnergy(cars[j].InitEn-
15 ergy)
16 else
17 cars[j].setEnergy(cars[j].InitEnergy)
18 EndIf
19 Update cars[j].cost and cars[j].energy
20 ElseIf cars[j] is moving▷ 移动
21 Distance←Map[cars[j].currentPos][cars[j].NextPos]
22 Update cars[j].cost according to Distance

```

```

23 EndIf
24 If( i == 22)
25 If cars[j].energy != cars[j].initEnergy
26 Num_uncharged++
27 EndIf
28 EndIf
29 EndFor
30 For j← 1 to 4▷ 检查各台区变是否超载
31 If Station[j].LoadFactor > 1
32 If Station[j].priority == 1
33 Num_1++
34 ElseIf Station[j].priority == 2
35 Num_2++
36 ElseIf Station[j].priority == 3
37 Num_3++
38 EndIf
39 EndIf
40 EndFor
41 EndFor
42 Return Rewards(num_uncharged,Num_1,Num_2,Num_3,NumCar)

```

在本文搭建的仿真环境中，我们结合题设给出的条件和限制，对充电车、充电站、台区变三类实体进行了建模。具体数据结构介绍如下：

表11 数据结构

Class Car
int ID
bool type
double cost
double energy
int location[16]
int action[16]
double power[16]

Class ChargeStation
<code>int</code> ID
<code>int</code> priority
<code>bool</code> is_chargeCar
<code>double</code> SunPowerRate[16]
<code>double</code> QueueTime
<code>int</code> chargeCar1
<code>int</code> chargeCar2
<code>vector<int></code> chargelist
Class DischargeStation
<code>int</code> ID
<code>int</code> priority
<code>double</code> MVA
<code>double</code> carCharge
<code>double</code> loadRate[16]