

报名序号：002093

论文题目：A 题 园区微电网风光储协调优化配置

园区微电网风光储协调优化配置

摘要

随着全球能源结构的转型和可持续发展战略的推进，园区微电网作为新型能源利用模式，日益受到重视。本文旨在通过数学建模和优化方法，提出园区微电网风光储协调优化配置方案，以提高可再生能源的利用率，减少对传统能源的依赖，实现能源的清洁、高效利用。针对三个独立接入主电网的园区微电网，本文首先分析了在不同运营模式下的风光储协调优化配置方案及其经济性。研究内容包括：未配置储能时的经济效益评估、储能配置与运行策略制定、储能配置优化探讨，以及联合运营储能配置方案及其经济性分析。

在独立运营模式下，本文评估了未配置储能系统时各园区的购电量、弃风弃光电量、总供电成本以及单位电量的平均供电成本，并分析了影响经济性的关键因素。随后，为各园区配置了 50kW/100kWh 储能设备，并制定了储能的最优运行策略及购电计划，评估了储能配置对提升经济性的影响。此外，证明 50kW/100kWh 储能方案并非最优，提出了各园区的最优方案分别为 194kW/1129kWh，213kW/626kWh，261kW/1270kWh，并论证了所制定方案的优越性。

在联合运营模式下，本文分析了联合园区运行的经济性，包括总购电量、总弃风弃光电量、总供电成本和单位电量平均供电成本。进一步制定了联合园区的总储能最优配置方案，为 194kW/1129kWh，还包括储能运行策略和购电计划，并分析了其经济性。最后，比较了联合运营与各园区独立运营的经济收益差异，并分析了导致经济收益变化的主要因素。

此外，本文还考虑了负荷增长下的配置方案，分别为独立运营时的 A，1981kW 光伏装机，储能 1084kW/6611kWh，B，1806kW 风电装机，储能 486kW/1732kWh，C，401kW 光伏装机，1608kW 风电装机，储能 622kW/1257kWh，联合运营时的，1556kW 风电装机，4471kW 风电装机，储能 1044kW/3126kWh。

本文采用的主要建模工具包括 MATLAB 编程和遗传算法等非线性优化方法。通过模型的建立与求解，本文为园区微电网提供了一个经济高效、可持续的能源管理方案，并对模型的有效性进行了验证。研究成果对于促进园区微电网的优化发展具有重要的理论和实践意义。

关键词：数学建模；MATLAB；非线性优化；遗传算法；园区微电网；风光储协调优化配置

一、问题提出

1.1 问题背景

随着全球能源结构的转型和可持续发展战略的推进，园区微电网作为新型能源利用模式，日益受到重视。园区微电网通常由风电、光伏发电等可再生能源与主电网联合供电，旨在提高可再生能源的利用率，减少对传统能源的依赖，实现能源的清洁、高效利用。然而，风电和光伏发电的间歇性和不稳定性，以及园区负荷的波动性，给微电网的稳定运行和经济性带来了挑战。特别是风光发电与负荷需求的时序不匹配问题，可能导致弃风弃光现象，影响能源的充分利用。因此，合理配置储能系统，优化风光储协调运行策略，对于提高园区微电网的经济性和可靠性具有重要意义。

1.2 问题重述

设有如图 1-1 所示的三个园区微电网各自独立接入主电网。

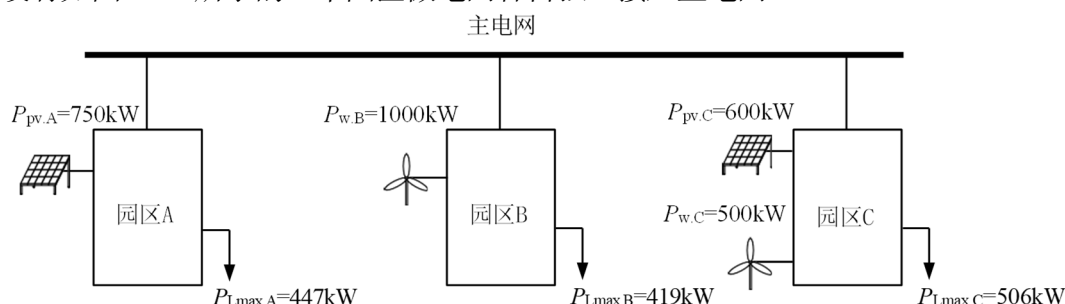


图 1-1 三个园区微电网各自独立接入主电网结构示意图

其中， $P_{pv,A}$ 、 $P_{pv,C}$ 分别为园区 A、C 的光伏装机容量， $P_{w,B}$ 、 $P_{w,C}$ 为园区 B、C 的风电装机容量， $P_{Lmax,A}$ 、 $P_{Lmax,B}$ 、 $P_{Lmax,C}$ 为园区 A、B、C 的负荷最大值，而各园区典型日负荷曲线如图 1-2 所示。

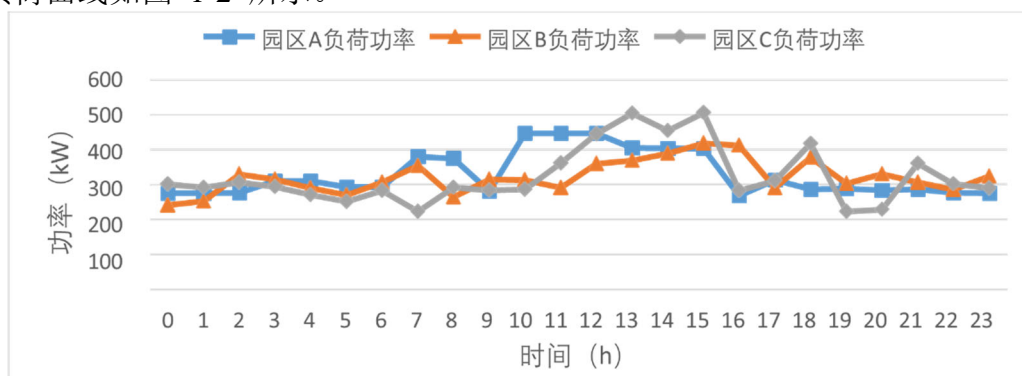


图 1-2 三个园区典型日负荷功率

本题目要求对三个独立接入主电网的园区微电网，在不同运营模式下的风光储协调优化配置方案及其经济性进行深入分析。具体问题如下：

1. 独立运营储能配置方案及经济性分析：

- (1) **未配置储能的经济效益评估：**在未配置储能系统的情况下，基于典型日风光发电功率和负荷曲线，计算各园区的购电量、弃风弃光电量、总供电成本以及单位电量的平均供电成本。分析影响各园区经济性的关键因素。
- (2) **储能配置与运行策略制定：**为各园区配置 50kW/100kWh 储能设备，制定储能的最优运行策略及购电计划，评估储能配置对提升经济性的影响，并解释原因。
- (3) **储能配置优化探讨：**在给定条件下，探讨 50kW/100kWh 储能方案是否为最优。若非最优，需提出各园区的最优储能功率和容量配置方案，并论证所制定方案

的优越性。

2. 联合运营储能配置方案（结构如图 1-3 所示）及经济性分析：

- (1) **联合园区运行经济性分析：**在未配置储能的情况下，分析联合园区的总购电量、总弃风弃光电量、总供电成本和单位电量平均供电成本。
- (2) **联合园区储能配置与运行策略：**制定联合园区的总储能最优配置方案，包括储能运行策略和购电计划，并分析其经济性。
- (3) **经济收益比较与因素分析：**比较联合运营与各园区独立运营的经济收益差异，并分析导致经济收益变化的主要因素。

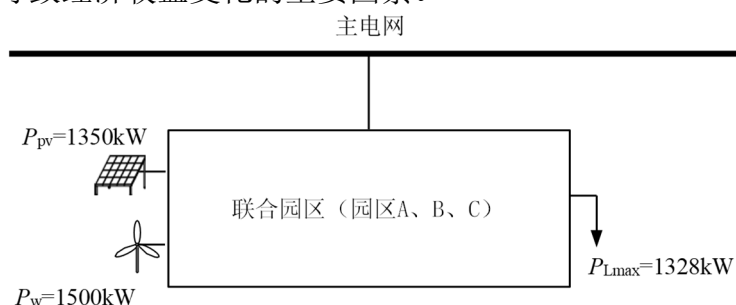


图 1-3 三个园区微电网联合接入主电网结构示意图

3. 风光储协调配置方案及经济性分析：

- (1) **负荷增长下的配置方案：**考虑三个园区的最大负荷增长 50%，且负荷波动特性不变，制定独立运营和联合运营的风光储协调配置方案。
- (2) **全年风光储协调配置方案：**根据附件 3 提供的全年 12 个月典型日风光发电功率数据，以及表 1 所示的分时电价，制定各园区独立运营的风光储协调配置方案。

表 1-1 分时电价表

时段	电价（元/kWh）
7:00-22:00	1
其余时段	0.4

二、基本假设

1. 风光荷功率波动特性在分析期间保持不变。
2. 储能系统充/放电效率和运行寿命参数具有普遍性和稳定性，不受外界因素和使用情况影响（如：温度、放电深度、电流大小等因素）。
3. 经济性分析仅考虑购电成本、弃风弃光电量、储能成本和运行策略，不考虑其他潜在成本，如维护、管理费用等。
4. 联合运营场景中，三个园区可以无障碍地共享和协调发电与负荷，不存在损耗。
5. 经费无限制，即电池容量和功率的上限无穷大。
6. 风电、光伏系统运营期间不可暂停，即若负载小于发电量，也不能停止发电。
7. 假设一年为 365 天，即不区分闰年平年。
8. 风光发电装置寿命无穷长，即无需将建造风光发电装置的费用平均分配至日购电成本。
9. 储能系统的参数（储能容量、功率）取整数值而非实数值（考虑到整数限制更贴近实际建造）。

三、符号说明

符号	说明	单位
t	时间	h
$P_L(P_{Lph})$	(分时) 负荷需求	kWh
$P_{pv}(P_{pvph})$	(分时) 光伏发电功率	kWh
$P_w(P_{wph})$	(分时) 风电发电功率	kWh
$P_t(P_{tph})$	(分时) 风光发电总功率	kW
$W_L(W_{Lph})$	(分时) 负荷需求电量	kWh
$W_{pv}(W_{pvph})$	(分时) 光伏发电量	kWh
$W_w(W_{wph})$	(分时) 风电发电电量	kWh
$W_t(W_{tph})$	(分时) 风光发电总电量	kWh
$W_{pur}(W_{purph})$	(分时) 购电量	kWh
$W_{aban}(W_{abanph})$	(分时) 弃风弃光电量	kWh
$C_{pur}(C_{purph})$	(分时) 购电成本	元
P_{rnet}	主电网购电单价	元/kWh
P_{rw}	风电购电单价	元/kWh
P_{rpv}	光伏购电单价	元/kWh
C_{ave}	单位电量平均供电成本	元/kWh
$S_{batt}(S_{battph})$	(分时) 电池储电量	kWh
S_m	电池容量	kWh
P_m	电池功率	kW
P_{mpv}	光伏电源装机量	kW

P_{mw}	风电电源装机量	kW
----------	---------	----

四、问题分析

在本题中，我们面临的是如何优化园区微电网的储能配置，以提高风光发电的利用率并降低运营成本。考虑到储能系统采用的是磷酸铁锂电池，其 SOC 允许范围被设定在 10%-90%，这表明电池的充放电深度受到限制，以延长电池的使用寿命。鉴于电池的运行寿命为 10 年，本题聚焦于典型日的运行情况，我们可以合理假设电池在初始时刻并非完全未充电状态，即电池的前 10%电量已经被充入，无需在模型中特别考虑。

为了简化模型并减少约束条件，我们进一步假设电池的容量可以按照装机电池容量的 80%来等效处理，同时将 SOC 的允许范围扩展至 0%-100%，这在数学建模中是一种常见的简化策略，它有助于我们更清晰地构建和求解模型。此外，电池的充放电效率为 95%，这意味着充电电量的 95%将转化为电池的电量增量，而在放电时，实际释放的电量应为放电电量除以 95%，以此来计算电池的电量减量。

在运营规则方面，无论是独立运营还是联合运营，各园区的可再生能源发电首先满足本区域负荷和储能的需求。在独立运营模式下，各园区的多余电量不能出售给主电网，即出现弃风弃光情况；不足的电量则需要通过储能系统放电或从主电网购入。而在联合运营模式下，三个园区被视为一个整体，同样遵循上述的运营限制。

对于储能策略，由于题目没有区分使用光伏或风电充电的成本差异，也没有区分不同时间段的放电成本差异，我们在设计储能运行方案时，将采取一种效率优先的策略：即在可能的情况下，电池将在第一时间充电，并在需求出现时第一时间放电。

最后，由于本题考虑的是典型日的情况，我们需要确保每天的运营周期结束时，系统的初始状态与次日的初始状态相同。这意味着，我们需要对模型施加一个约束，确保每天结束时电池的 SOC 与次日开始时的 SOC 一致，以满足典型日的循环条件。

通过上述分析，我们可以构建一个综合考虑储能配置、运营模式和充放电策略的优化模型，旨在为园区微电网提供一个经济高效、可持续的能源管理方案。

五、模型的建立与求解

5.1 问题一的模型建立与求解

5.1.1 问题一(1)的模型建立与求解

5.1.1.1 问题一(1)的分析

在未引入储能系统的情况下，园区微电网的电力供应直接依赖于风光发电与负荷需求之间的即时平衡。若负荷需求 (P_L) 超出风光发电总功率 ($P_t = P_{pv} + P_w$)，则需从主电网购电，购电量可通过负荷需求与风光发电总功率之差确定。相应地，购电成本直接与购电量和主电网购电价格相关。若风光发电总功率大于负荷需求，则产生弃风弃光现象，弃风弃光电量的计算方式与购电量相反，即由风光发电总功率与负荷需求之差决定。由于风光发电成本已产生，这部分电量虽未被利用，其成本也必须计入总成本中。

总供电成本由风光发电成本、主电网购电成本以及弃风弃电量对应的成本组成。具体地，风光发电成本由风光发电功率与各自购电成本的乘积确定；主电网购电成本则是购电量与购电价格的乘积；弃风弃光电量的成本则由弃风弃电量与风光发电购电成本的乘积得出。单位电量平均供电成本计算为总供电成本除以总负荷量。

5.1.1.2 问题一(1)模型的建立

典型日总发电功率

由发电功率计算发电电量，有

$$P_{tph} = p_{wph} + P_{pvph}$$

由负载功率计算负载电量，有

$$W_{tph} = P_{tph} \times t$$

由负载功率计算负载电量，有

$$W_{pvph} = P_{pvph} \times t$$

$$W_{wph} = P_{wph} \times t$$

$$W_{Lph} = P_{Lph} \times t$$

于是，分时购电量

$$W_{purph} = \max(0, W_{Lph} - W_{tph})$$

弃风弃光分时电量

$$W_{abanph} = \max(0, W_{tph} - W_{Lph})$$

从而有分时供电成本

$$C_{purph} = P_{rnet} \times W_{purph} + P_{rw} \times W_{wph} + P_{rpv} \times W_{pvph}$$

对于分时变量 X_{ph} 和日总变量 X 的关系，我们令 X 为 X_{ph} 对时间求和，即

$$X = \sum_{t=0}^{23} X_{ph}(t)$$

其中， X 可以为具有形如 X_{ph} 的变量（下标中有ph）。

从而我们可以得到单位电量平均供电成本

$$C_{ave} = \frac{C_{pur}}{W_L}$$

综上所述我们有购电量（ C_{pur} ）、弃风弃光电量（ W_{aban} ）、总供电成本（ C_{pur} ）和单位电量平均供电成本（ C_{ave} ）。据此可以分析各变量如何影响经济性。

5.1.1.3 问题一(1)模型的求解

从附件 1 中获取各园区典型日负荷数据（ P_{Lph} ），附件 2 中获取各园区典型日风光发电数据（ P_{wph} 、 P_{pvph} ），若无相应设备装机，则记为 0。

对于 $W = P \times t$ ，从附件获知其数据为分时数据，则有 $t \equiv 1$ ，即 W 与 P 在数值上相等。

使用 MATLAB 编程求解，程序源代码见附录 1，得到未配置储能时各园区独立运营结果，如表 5-1 所示。

表 5-1 未配置储能时各园区独立运营结果

项目 \ 园区	园区 A	园区 B	园区 C
购电量（ C_{pur} ）(kWh)	4874.12	2432.30	2699.39
弃风弃光电量（ W_{aban} ）(kWh)	951.20	897.50	1128.02
总供电成本（ C_{pur} ）(元)	6465.35	5519.90	5472.08
单位电量平均供电成本（ C_{ave} ）(元/kWh)	0.818296	0.715940	0.703714

分析各园区分时数据，原始数据见附件，我们有各园区购电量（图 5-1）、弃风弃光电量（图 5-2）、购电成本（图 5-3）、单位电量平均供电成本（图 5-4）的时间变化图。

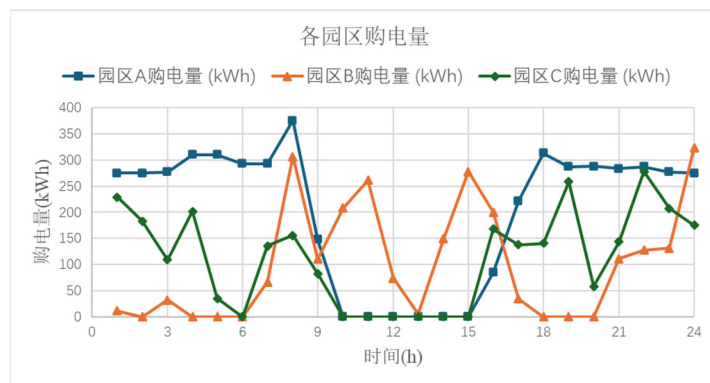


图 5-1 未配置储能各园区购电量

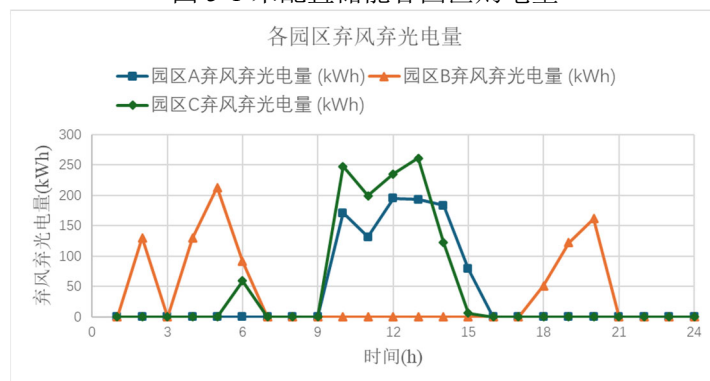


图 5-2 未配置储能各园区弃风弃光电量

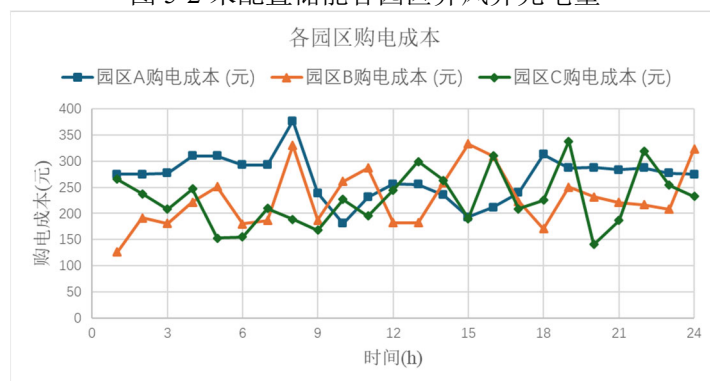


图 5-3 未配置储能各园区购电成本

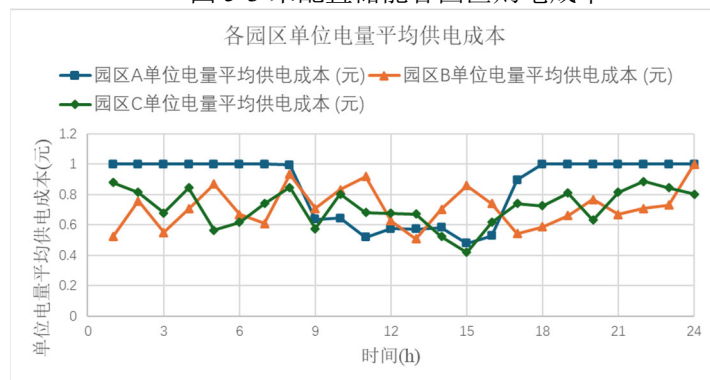


图 5-4 未配置储能各园区单位电量平均供电成本

可见，从电网购电的电量比例将显著影响单位电量平均供电成本，我们应当尽可能保证使用风光发电满足负载需求。

5.1.2 问题一(2)的模型建立与求解

5.1.2.1 问题一(2)的分析

配置储能系统后，园区微电网的电力供应策略得到优化。储能系统能够在风光发电量超过负荷需求时储存多余电量，并在负荷需求超过风光发电量时释放能量，从而减少从主电网购电的需求和弃风弃光现象。储能系统的引入增加了总成本，其中包括电池成本除以电池寿命得到的平均成本。

在给定的储能策略(50kW/100kWh)下，最优运行策略是确保电池在风光发电量超过负荷需求时尽可能多地储存能量，并在负荷需求超过风光发电量时及时释放能量。购电计划需要根据储能系统的充放电情况和分时电价进行调整，以最小化购电成本。

5.1.2.2 问题一(2)模型的建立

典型日总发电功率

$$P_{tph} = p_{wph} + P_{pvph}$$

由发电功率计算发电电量，有

$$W_{tph} = P_{tph} \times t$$

由负载功率计算负载电量，有

$$W_{pvph} = P_{pvph} \times t$$

$$W_{wph} = P_{wph} \times t$$

$$W_{Lph} = P_{Lph} \times t$$

由于已经配置储能，所以需要逐小时考虑电池是否能充电、是否需要放电。逻辑见以下伪代码：

```
If 负载小于发电量
    If 充电功率足够
        If 不能充满
            充电
        Elseif 能充满
            电池充满，剩余部分计入弃风弃光电量
        End
    Elseif 充电功率较小
        If 不能充满
            按最大功率充电，剩余部分计入弃风弃光电量
        Elseif 能充满
            电池充满，剩余部分计入弃风弃光电量
        End
    End
Elseif 负载量大于发电量
    If 放电功率足够
        If 储能足够
            放电
        Elseif 储能不足
            电池放空，，不足部分向主电网购电
        End
    Elseif 放电功率不足
        If 储能足够
            全功率放电，不足部分向主电网购电
        Elseif 储能不足
            电池放空，不足部分向主电网购电
```

End

End

End

需要注意的是，我们根据前述分析已经对电池模型 SOC 进行简化，即模型电池容量为建设容量的 80%，而 SOC 允许范围为 0%-100%。同时，由于充放电效率并非 100%，在处理上述逻辑时，充电时计入电池充电的电量需要乘以系数 0.95，而放电时需要放出需求电量的 1/0.95 倍。

从而，得到购电量、弃风弃光电量、储电量的分时信息。

在不考虑储电成本时，有分时供电成本

$$C_{purph} = P_{rnet} \times W_{purph} + P_{rw} \times W_{wph} + P_{rpv} \times W_{pvph}$$

由于电池寿命为 10 年，可以将电池的成本平均到 10 年内的每个小时，该平均成本即可计入分时供电成本，即

$$C_{purph} = P_{rnet} \times W_{purph} + P_{rw} \times W_{wph} + P_{rpv} \times W_{pvph} + 1800S_m + 800P_m$$

同样，对于分时变量 X_{ph} 和日总变量 X 的关系，我们令 X 为 X_{ph} 对时间求和，即

$$X = \sum_{t=0}^{23} X_{ph}(t)$$

其中， X 可以为具有形如 X_{ph} 的变量（下标中有ph）。

从而我们可以得到单位电量平均供电成本

$$C_{ave} = \frac{C_{pur}}{W_L}$$

综上我们得到考虑储能时购电量(C_{pur})、弃风弃光电量(W_{aban})、总供电成本(C_{pur})和单位电量平均供电成本 (C_{ave})。

5.1.2.3 问题一(2)模型的求解

按照上述储能模型，可以在问题一(1)的基础上，将原模型嵌入储能模型后用同样方法求解，设定 $P_m = 50\text{kW}$ 和 $S_m = 100\text{kWh}$ ，使用 MATLAB 编程进行求解，源代码见附录 1，得到储能方案为50kW/100kWh时的各参数如表 5-2 所示。

表 5-2 配置储能50kW/100kWh时各园区独立运营结果

项目 \ 园区	园区 A	园区 B	园区 C
购电量 (C_{pur}) (kWh)	4798.12	2248.10	2578.26
弃风弃光电量 (W_{aban}) (kWh)	866.99	693.40	993.81
总供电成本 (C_{pur}) (元)	6439.77	5386.11	5401.36
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.815057	0.698588	0.694620

对于优化目标单位电量平均供电成本而言，显然各园区运行经济性均有不同程度的改善。

分析各园区分时数据，原始数据见附件，我们有各园区购电量（图 5-5）、弃风弃光电量（图 5-6）、购电成本（图 5-7）、单位电量平均供电成本（图 5-8）、储电量（图 5-9）的时间变化图。

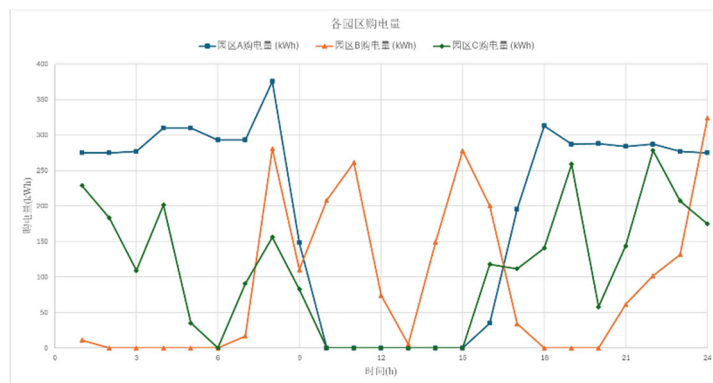


图 5-5 配置储能各园区购电量

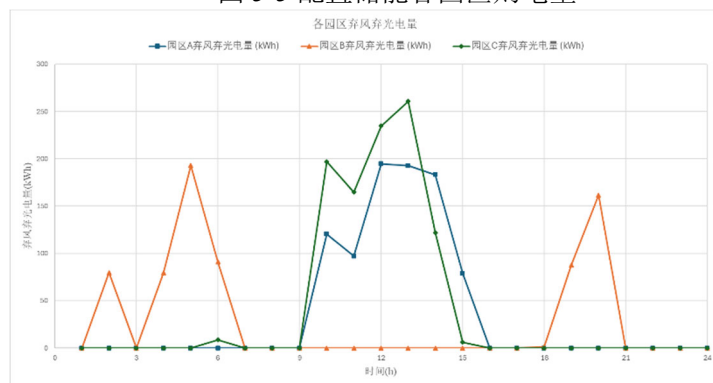


图 5-6 配置储能各园区弃风弃光电量

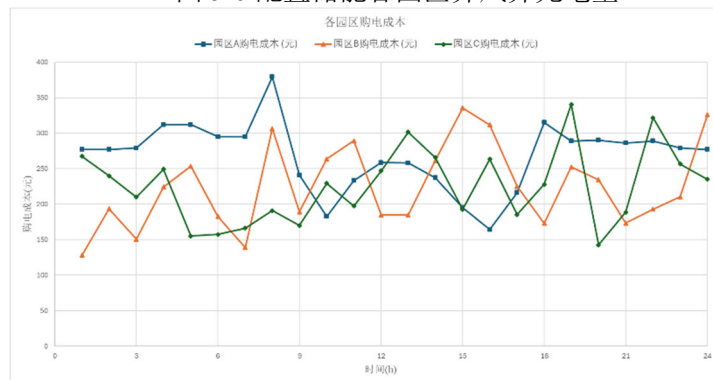


图 5-7 配置储能各园区购电成本

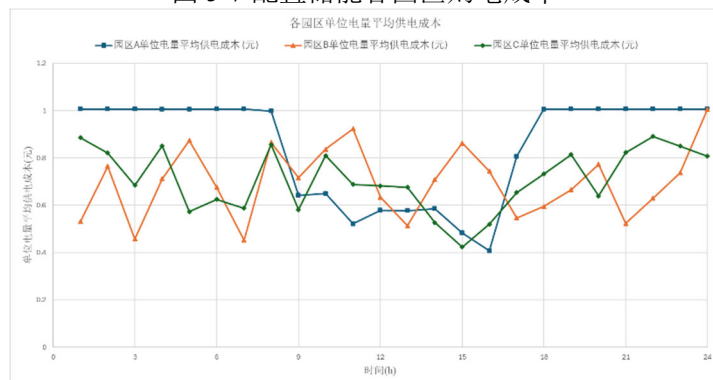


图 5-8 配置储能各园区单位电量平均供电成本

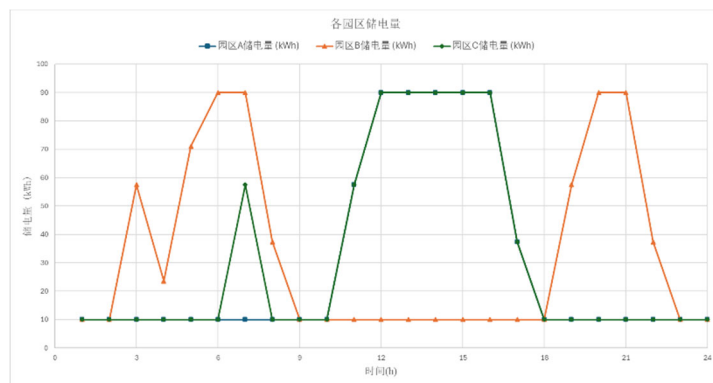


图 5-9 配置储能各园区储电量

可以看到，由于储能装置的存在，图 5-5 和图 5-7 的高点均有不同程度的下降，显然，储能装置将原弃风弃光电量存储下来，在用电高峰期放电，能够有效的降低单位电量平均供电成本。

购电计划即为购电量，如表 5-3 所示。

表 5-3 各园区购电计划

时间 (h)	园区 A 购电量 (kWh)	园区 B 购电量 (kWh)	园区 C 购电量 (kWh)
00:00	275	10.9	228.8
01:00	275	0	183.25
02:00	277	0	109.05
03:00	310	0	201.45
04:00	310	0	35.2
05:00	293	0	0
06:00	293	16.8	90.575
07:00	375.65	280.7	156
08:00	148.05	110.2	82.7
09:00	0	208.2	0
10:00	0	261.2	0
11:00	0	74.1	0
12:00	0	5.4	0
13:00	0	149.6	0
14:00	0	278	0
15:00	34.85	200.4	117.85
16:00	195.575	34.1	111.79
17:00	313	0	140.6
18:00	287	0	258.85
19:00	288	0	58.05
20:00	284	61.3	143.85
21:00	287	101.7	278.25
22:00	277	131.5	207.15
23:00	275	324	174.85

5.1.3 问题一(3)的模型建立与求解

5.1.3.1 问题一(3)的分析

针对给定的储能策略(50kW/100kWh)，我们将分析其在减少弃风弃光电量、降低总供电成本方面的有效性，并以单位电量平均供电成本作为评价指标。若该策略非最优，我们将通过调整储能功率和容量作为决策变量，进行单目标优化，寻找更优的储能配置方案。优化的目标是 minimized 单位电量平均供电成本（该成本已经考虑了储能系统的经济性和技术参数，如功率单价、能量单价、SOC 允许范围、充放电效率等），以论证所提出的储能配置方案的优越性。

5.1.3.2 问题一(3)模型的建立

沿用问题一(2)，将此模型封装为一个非线性目标函数，我们的目标即为

$$\min Z = f(P_m, S_m)$$

其中 P_m 和 S_m 作为该非线性优化的决策变量。

下面考虑约束。

由于 SOC 允许范围已在目标函数中考虑，无需额外添加该约束。

由于 P_m 和 S_m 已经计入成本，无需额外添加诸如充电功率小于等于电池容量等约束。

考虑到实际生活，我们将 P_m 和 S_m 约束为整数。

考虑到问题为典型日数据，我们需要保证该日开始时电池电量应当与该日结束即下一日开始时相同，否则无法形成完整的电池循环，因此需要对此进行约束，即

$$S_{battph}(0) \equiv C$$

C 为常数。

此为非线性约束。

5.1.3.3 问题一(3)模型的求解

首先建立上述非线性约束条件，仍使用问题一(2)模型中的电池充放电逻辑对电池电量进行递推，可以得到次日 0 点的电池电量数据，则等式

$$S_{battph}(0) = S_{battph}(24)$$

即可对每日初始电量进行完备约束，则现在的决策变量为 $P_m, S_m, S_{battph}(0)$ 。

对于整数约束，采用 MATLAB 工具 ga（遗传算法）即可自动约束，由于 MATLAB ga 不支持整数约束与等式约束共存，修改上述初始电量约束为

$$|S_{battph}(0) - S_{battph}(24)| \leq 0$$

即可。

使用 MATLAB 编程求解该非线性优化，源代码见附录 1，得到最优储能方案下的各园区参数如表 5-4 所示。

表 5-4 配置最优储能方案时各园区独立运营结果

项目 \ 园区	园区 A	园区 B	园区 C
容量(S_m) (kWh)	1129	626	1270
功率(P_m) (kW)	194	213	261
日初始电量($S_{battph}(0)$) (kWh)	112.90	62.60	127.00
购电量 (C_{pur}) (kWh)	4016.23	1622.45	1681.35
弃风弃光电量 (W_{aban}) (kWh)	0.62	0.16	0.00

总供电成本 (C_{pur}) (元)	6095.40	5003.71	5012.29
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.771471	0.648989	0.644584

显然，此最优储能方案并非50kW/100kWh，而此方案下任意园区单位电量平均供电成本均低于50kW/100kWh方案下成本。且此方案充分利用了峰值发电，极大程度降低了弃风弃光电量，符合环保意识。

此方案下各园区分时数据有各园区购电量（图 5-10）、弃风弃光电量（图 5-11）、购电成本（图 5-12）、单位电量平均供电成本（图 5-13）、储电量（图 5-14）的时间变化图，原始数据见附件。

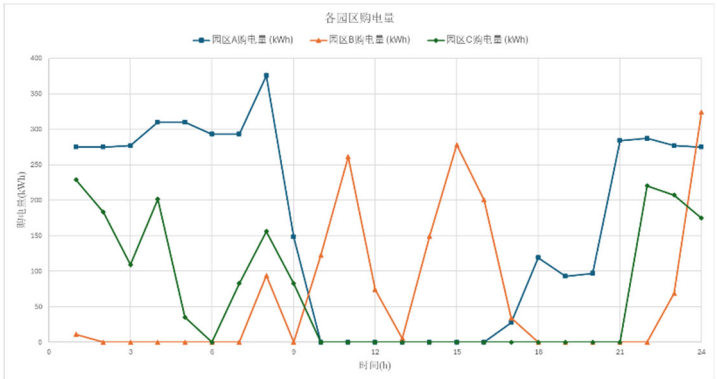


图 5-10 最优储能各园区购电量

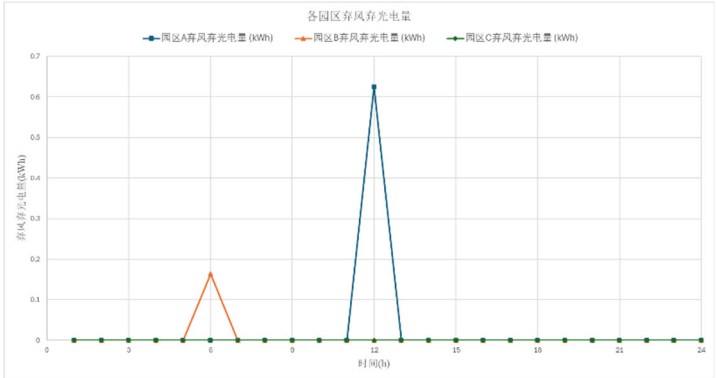


图 5-11 最优储能各园区弃风弃光电量

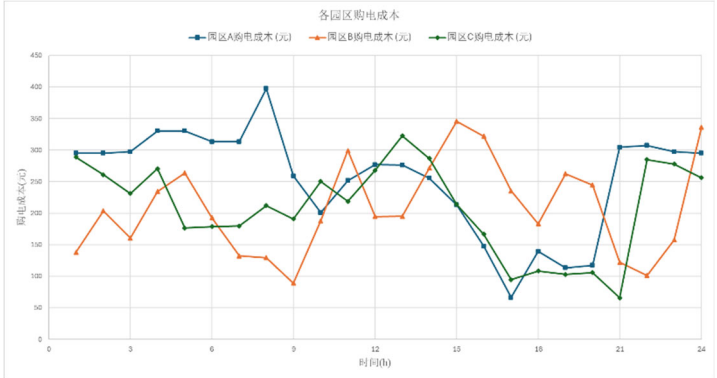


图 5-12 最优储能各园区购电成本

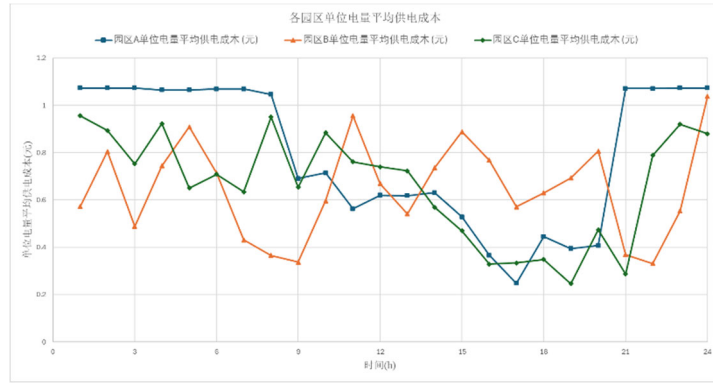


图 5-13 最优储能各园区单位电量平均供电成本

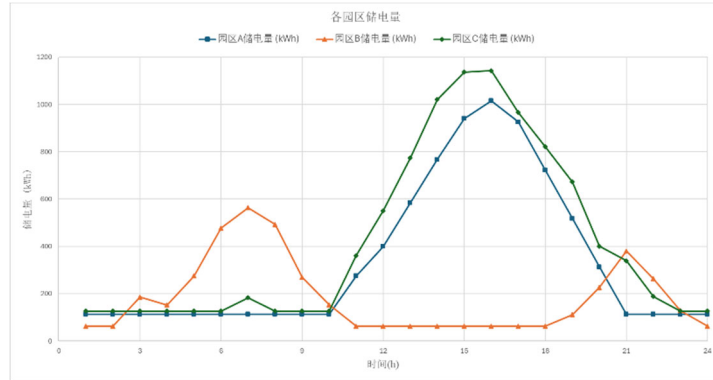


图 5-14 最优储能各园区储电量

可见最优方案下，过剩风光发电几乎被利用。

由于本文直接调用 MATLAB 函数，因此将不花费篇幅针对遗传算法进行优化，也不计划提供收敛性曲线等遗传算法相关的图表。

5.2 问题二的模型建立与求解

5.2.1 问题二(1)的模型建立与求解

5.2.1.1 问题二(1)的分析

在未配置储能系统的情况下，联合园区的电力供应策略相对简单，但同时也面临较大的经济风险。当联合园区的总负荷需求(P_L)超出风光发电总功率($P_t = P_{pv} + P_w$)时，必须从主电网购电，总购电量即为负荷需求与风光发电总功率之差。相应地，总购电成本为购电量乘以主电网购电价格。相反，若风光发电总功率超过总负荷需求，则产生弃风弃光现象，这部分电量无法利用，但其生产成本已经产生，也必须计入经济性考量。

总供电成本由风光发电成本、主电网购电成本以及弃风弃光电量对应的成本组成。单位电量平均供电成本计算为总供电成本除以总负荷量。此部分分析将明确联合园区在未配置储能时的经济性状况，为后续优化提供基准。

5.2.1.2 问题二(1)模型的建立

将各园区数据直接求和后即可得到联合园区数据，以联合园区数据作为问题一(1)中的单园区数据，模型与问题一(1)完全一致，不再过多赘述。

5.2.1.3 问题二(1)模型的求解

从附件 1 中获取各园区典型日负荷数据，并对所有园区求和，得到联合园区典型日负荷数据 (P_{Lph})，从附件 2 中获取各园区典型日风光发电数据 (取消归一化)，并对所有园区求和，得到联合园区典型日风光发电数据 (P_{wph} 、 P_{pvph})，若无相应设备装机，则记为 0。

使用 MATLAB 求解，源代码见附录 1。得到未配置储能时联合园区结果，如表 5-5 所示。

表 5-5 未配置储能时联合园区结果

项目 \ 园区	联合园区
购电量 (C_{pur}) (kWh)	8266.27
弃风弃光电量 (W_{aban}) (kWh)	1237.17
总供电成本 (C_{pur}) (元)	15717.79
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.672074

联合园区分时结果如表 5-6 所示。

表 5-6 未配置储能时联合园区分时结果

时间 (h)	购电量 (kWh)	弃风弃光电量 (kWh)	购电成本 (元)	单位电量平均供电成本 (元/kWh)
00:00	514.7	0	666.35	0.814608802
01:00	328.45	0	574.225	0.70027439
02:00	418.25	0	665.625	0.729052574
03:00	382.05	0	650.025	0.708088235
04:00	132.3	0	501.65	0.575947187
05:00	143.35	0	479.175	0.587944785
06:00	495.5	0	689.25	0.780577576
07:00	838.35	0	896.61	0.936896552
08:00	340.95	0	593.6	0.637593985
09:00	0	209.29	461.111	0.524585893
10:00	0	69.085	452.594	0.432276982
11:00	0	355.225	609.9	0.554454545
12:00	0	448.115	732.581	0.584661612
13:00	0	155.46	608.589	0.476204225
14:00	193.205	0	632.323	0.506669071
15:00	453.1	0	833.085	0.627323042
16:00	393.465	0	672.294	0.698124611
17:00	402.5	0	658.75	0.719945355
18:00	424.05	0	754.025	0.695595018
19:00	184.45	0	499.225	0.613298526
20:00	539.15	0	691.575	0.819401659
21:00	692.95	0	823.475	0.863181342
22:00	615.65	0	739.825	0.856278935
23:00	773.85	0	831.925	0.934747191
00:00				

联合园区经济性显然高于各园区独立运营，这可以从联合园区未配置储能时的弃风弃光电量小于各园区独立运营理解。

5.2.2 问题二(2)的模型建立与求解

5.2.2.1 问题二(2)的分析

配置储能系统后，联合园区的电力供应策略将更加灵活和经济高效。储能系统能够在风光发电量超过负荷需求时储存多余电量，并在负荷需求超过风光发电量时释放能量，从而减少从主电网购电的需求和弃风弃光现象。制定储能的最优配置方案需要考虑储能系统的功率和容量，以及与风光发电和负荷需求的匹配程度。

储能运行策略的制定将基于风光发电和负荷的实时数据，确保在任何给定时间点，联合园区的电力供应都能达到最优平衡。购电计划将根据储能系统的充放电情况和分时电价进行调整，以最小化购电成本。此部分分析将探讨储能配置对联合园区经济性的影响，并提出储能运行的最优策略。

5.2.2.2 问题二(2)模型的建立

将各园区数据直接求和后即可得到联合园区数据，以联合园区数据作为问题一(3)中的单园区数据，模型与问题一(3)完全一致，不再过多赘述。

5.2.2.3 问题二(2)模型的求解

由于风光荷功率波动保持上述条件不变，而总负荷最大值为 1328kW正好为各园区负载分时和的最大值，因此，风光荷功率直接从附件读取并求和即可。

求和得到联合园区数据后，仍使用 MATLAB 的 ga 解决该非线性单目标优化问题。源代码见附录 1。得到未配置储能时联合园区结果，如表 5-7 所示。

表 5-7 配置最优储能时联合园区结果

项目	联合园区
容量(S_m) (kWh)	1129
功率(P_m) (kW)	194
日初始电量($S_{battph}(0)$) (kWh)	112.90
购电量 (C_{pur}) (kWh)	7149.83
弃风弃光电量 (W_{aban}) (kWh)	0.12
总供电成本 (C_{pur}) (元)	15279.09
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.653315

分时结果如表 5-8 所示。

表 5-8 配置最优储能时联合园区分时结果

时间 (h)	储电量 (kWh)	购电量 (kWh)	弃风弃光电量 (kWh)	购电成本 (元)	单位电量平均供电成本 (元)
00:00	146.9	514.7	0	694.5892694	0.849131136
01:00	146.9	328.45	0	602.4642694	0.734712524
02:00	146.9	418.25	0	693.8642694	0.75998277
03:00	146.9	382.05	0	678.2642694	0.738849967
04:00	146.9	132.3	0	529.8892694	0.608368851
05:00	146.9	143.35	0	507.4142694	0.622594196

06:00	146.9	495.5	0	717.4892694	0.812558629
07:00	146.9	838.35	0	924.8492694	0.96640467
08:00	146.9	340.95	0	621.8392694	0.667926176
09:00	146.9	0	0	489.3502694	0.556712479
10:00	345.7255	0	0	480.8332694	0.459248586
11:00	411.35625	0	0	638.1392694	0.580126609
12:00	748.82	0	0.115	760.8202694	0.607198938
13:00	1174.42	0	0.007368421	636.8282694	0.49830068
14:00	1322.1	0	0	467.3572694	0.374484992
15:00	1118.726316	5.1	0	413.3242694	0.311238155
16:00	647.1473684	0	0	307.0682694	0.318866323
17:00	232.9736842	320.73	0	605.2192694	0.661441824
18:00	146.9	424.05	0	782.2642694	0.721646005
19:00	146.9	184.45	0	527.4642694	0.647990503
20:00	146.9	539.15	0	719.8142694	0.852860509
21:00	146.9	692.95	0	851.7142694	0.892782253
22:00	146.9	615.65	0	768.0642694	0.888963275
23:00	146.9	773.85	0	860.1642694	0.966476707
00:00	146.9				

购电策略即为购电量。储能运行策略即为储电量。此储能策略下，单位电量平均供电成本低于无配置的情况，且弃风弃光电量极低，充分利用了储能优势。

5.2.3 问题二(3)的模型建立与求解

5.2.3.1 问题二(3)的分析

联合运营相比独立运营可能带来显著的经济收益。通过整合三个园区的风光发电资源和负荷需求，联合园区可以实现更高效的资源分配和能源利用，减少弃风弃光现象，降低对主电网购电的依赖。

此部分分析将比较联合运营与独立运营在总购电量、总弃风弃光电量、总供电成本和单位电量平均供电成本方面的差异，明确联合运营带来的经济收益，并探讨导致经济收益改变的主要因素。

5.2.3.2 问题二(3)模型的建立

比较联合运营与独立运营的总购电量、总弃风弃光电量、总供电成本和单位电量平均供电成本方面的差异。

独立运营时，总购电量、总弃风弃光电量、总供电成本对各园区直接求和即可，对于单位电量平均供电成本，有

$$C_{ave} = \frac{C_{pur}}{W_L}$$

联合运营时，各参数为问题二(1)的解。

两者比较经济性即可。

5.2.3.3 问题二(3)模型的求解

由于考虑储能时，各园区最优策略不一致，不具有可比性，因此采用不储能时的策略进行经济性分析。

对独立运营直接求和得到购电量、总弃风弃光电量、总供电成本，并计算单位电量平均供电成本。

联合运营直接使用问题二(1)的解数据。

表 5-9 联合独立运营参数

项目 \ 园区	联合运营	独立运营
购电量 (C_{pur}) (kWh)	8266.27	10005.815
弃风弃光电量 (W_{aban}) (kWh)	1237.175	2976.72
总供电成本 (C_{pur}) (元)	15717.787	17457.332
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.672073673	0.746454526

联合运营时分时数据如表 5-5 所示，独立运营时分时数据如表 5-10 所示。

表 5-10 独立运营时分时结果

时间 (h)	购电量 (kWh)	弃风弃光电量 (kWh)	购电成本 (元)	单位电量平均供电成本 (元)
00:00	514.7	0	666.35	0.814608802
01:00	458.25	129.8	704.025	0.858567073
02:00	418.25	0	665.625	0.729052574
03:00	511.45	129.4	779.425	0.849046841
04:00	345.2	212.9	714.55	0.820378875
05:00	293	149.65	628.825	0.771564417
06:00	495.5	0	689.25	0.780577576
07:00	838.35	0	896.61	0.936896552
08:00	340.95	0	593.6	0.637593985
09:00	208.2	417.49	669.311	0.761445961
10:00	261.2	330.285	713.794	0.681751671
11:00	74.1	429.325	684	0.621818182
12:00	5.4	453.515	737.981	0.588971269
13:00	149.6	305.06	758.189	0.593262128
14:00	278	84.795	717.118	0.574613782
15:00	453.1	0	833.085	0.627323042
16:00	393.465	0	672.294	0.698124611
17:00	453.6	51.1	709.85	0.77579235
18:00	545.85	121.8	875.825	0.807956642
19:00	346.05	161.6	660.825	0.811824324
20:00	539.15	0	691.575	0.819401659
21:00	692.95	0	823.475	0.863181342
22:00	615.65	0	739.825	0.856278935
23:00	773.85	0	831.925	0.934747191
00:00				

可以看到，联合运营时，每个园区的负载、发电通过联合运营进行均衡，使得弃风弃光电量得以降低，同时购电量也进一步减少。因此购电成本减少，而负载电量不变，

所以单位电量平均成本减少。

5.3 问题三的模型建立与求解

5.3.1 问题三(1)的模型建立与求解

5.3.1.1 问题三(1)的分析

在负荷增长 50%的前提下，首先需要重新评估各园区的负荷需求，并据此调整风光发电和储能的配置。独立运营时，每个园区需根据自身的负荷特性和风光发电条件来优化配置方案。联合运营时，则需要考虑三个园区作为一个整体，统一调配风光发电资源和储能设施，以实现资源的最优利用。

风光发电成本将根据风电、光伏电源的配置成本（分别为 3000 元/kW、2500 元/kW）来计算。储能配置则需考虑功率单价、能量单价、SOC 允许范围、充放电效率以及运行寿命等因素，通过单目标非线性优化来确定最优的储能功率和容量。

5.3.1.2 问题三(1)模型的建立

在问题一(3)的优化模型基础上，本小问需要将光伏装机容量和风电装机容量纳入决策范围内。除此之外，目标函数与前述相同。

考虑新增约束条件。

光伏装机容量和风电装机容量仍需要约束为整数。

同时，本题要求投资回报周期按照 5 年计算，即投入风光发电的经费需要在 5 年内通过项目节省的购电费用进行收回。因此需要增加投资回报期约束。

令 W 为投资金额，则有

$$W - 365 \times 5 \times (C_{pur_{ori}} - C_{pur_{now}}) \leq 0$$

其中，

$$\begin{aligned} C_{pur_{ori}} &= W_L \times P_{rnet} \\ C_{pur_{now}} &= C_{pur} \end{aligned}$$

此外约束与问题一(3)一致。

而对于联合园区，同问题二(2)，将各园区数据直接求和后即可得到联合园区数据，以联合园区数据作为问题一(3)中的单园区数据，模型即可与上述分析完全一致。

5.3.1.3 问题三(1)模型的求解

本问中，三个园区的最大符合增长 50%，但符合波动特性不变，可以简单的将附件 1 中各园区典型日负荷数据乘以系数 1.5 作为 P_L 进行分析。

增加的投资回报期约束可以直接添加。

使用 MATLAB 的 ga 函数求解，源代码见附录 1，得到最优风光储方案时各园区独立运营参数如表 5-11 所示。

表 5-11 最优风光储方案时各园区独立运营结果

项 目 \ 园 区	园区 A	园区 B	园区 C
容量(S_m) (kWh)	6611	1732	1257
功率(P_m) (kW)	1084	486	622
日初始电量($S_{battph}(0)$) (kWh)	2473.59	173.20	125.70
光伏电源装机量(P_{mpv}) (kW)	1981	0	401

风电电源装机量(P_{mw}) (kW)	0	1806	1608
购电量 (C_{pur}) (kWh)	1886.15	658.80	306.67
弃风弃光电量 (W_{aban}) (kWh)	0.00	0.00	0.00
总供电成本 (C_{pur}) (元)	8934.88	7024.84	6496.74
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.753903	0.607422	0.556991

将相关参数按园区求和后,使用 MATLAB 求解得到联合园区数据如表 5-12 所示。

表 5-12 最优风光储方案时联合运营结果

项 目 \ 园 区	联合园区
容量(S_m) (kWh)	3126
功率(P_m) (kW)	1044
日初始电量($S_{battph}(0)$) (kWh)	312.60
光伏电源装机量(P_{mpv}) (kW)	1556
风电电源装机量(P_{mw}) (kW)	4471
购电量 (C_{pur}) (kWh)	400.0
弃风弃光电量 (W_{aban}) (kWh)	0.00
总供电成本 (C_{pur}) (元)	18627.13
单位电量平均供电成本 (C_{ave}) (元/kWh)	0.530983

5.3.2 问题三(2)的模型建立与求解

5.3.2.1 问题三(2)的分析

结合全年的风光发电数据和分时电价,可以更精确地制定各园区独立运营时的风光储协调配置方案。这需要对每个月的风光发电功率数据进行详细分析,确定在不同季节和时间段内风光发电与负荷需求的匹配程度,以及储能设施在此过程中的调节作用。

在此基础上,制定的配置方案将包括风光发电装机容量的调整、储能系统的规模和充放电策略,以及与主电网互动的购电计划。这些方案将旨在最大化自给自足的能力,减少对主电网的依赖,同时考虑到经济效益和投资回报期,确保方案在 5 年内能够实现成本回收。

5.3.2.2 问题三(2)模型的建立

类似于第(1)小问，但需要考虑分时电价。且电量约束需考虑 12 个月。

5.3.2.3 问题三(2)模型的求解

使用 MATLAB 的 ga 求解。

六、模型的评价与推广

6.1 模型的评价

实用性：所提出的模型紧密结合园区微电网的实际运营情况，考虑了风光发电的间歇性和不稳定性，以及储能系统在其中的作用，具有较高的实用性和操作性。

优化效果：通过遗传算法等非线性优化方法，模型能够有效地寻找到储能配置的最优解，减少了弃风弃光现象，降低了运营成本，提高了能源利用效率。

灵活性：模型能够根据不同的运营模式（独立运营和联合运营）和不同的负荷需求，灵活调整风光储的配置方案，具有较强的适应性和灵活性。

经济性：模型考虑了储能系统的成本和收益，通过经济性分析，确保了方案的经济合理性，有助于实现投资回报。

可持续性：模型强调了可再生能源的利用和储能系统的优化配置，符合当前能源转型和可持续发展的趋势。

6.2 模型的推广

技术扩展：将模型推广到更广泛的应用场景，如不同规模的园区、不同类型的可再生能源（如水电、生物质能等），以及其他国家和地区的园区微电网。

多目标优化：在现有单目标优化的基础上，引入多目标优化，如同时考虑成本、可靠性、环境影响等多个目标，以实现更全面的优化效果。

政策因素：考虑政策变化对园区微电网运营的影响，如电价调整、补贴政策、碳交易等，使模型能够适应政策变动。

智能化管理：结合物联网、大数据和人工智能技术，开发智能化的能源管理系统，实现对园区微电网的实时监控和自动优化。

风险评估：引入风险评估机制，考虑风光发电不确定性和市场变化等因素，评估不同配置方案的风险，为决策提供更全面的支持。

社区参与：探索社区参与模式，鼓励园区内企业和居民参与能源管理，提高能源利用的民主性和公平性。

教育与培训：通过教育和培训，提高园区管理者和技术人员对微电网管理和优化技术的认识和能力，促进模型的实施和推广。

七、附录

7.1 附录 1 部分程序源码

7.1.1 cal.m

% 计算单个园区的购电量、弃风弃光电量、总供电成本和单位电量平均供电成本

```
function varargout = cal(park, varargin)
```

%% 输入参数处理

```
p = inputParser;
```

```
defaultDataPath = "../Data/";
```

```
addParameter(p, "DataPath", defaultDataPath, @isstring);
```

```

defaultResultsPath = "../Results/";
addParameter(p, "ResultsPath", defaultResultsPath, @isstring);

defaultResultsName = "results.xlsx";
addParameter(p, "ResultsName", defaultResultsName, @isstring);

defaultInitialize = false;
addParameter(p, "Initialize", defaultInitialize, @islogical);

defaultQuiet = false;
addParameter(p, "Quiet", defaultQuiet, @islogical);

defaultCapacity = 0;
addParameter(p, "Capacity", defaultCapacity, @isnumeric);

defaultPower = 0;
addParameter(p, "Power", defaultPower, @isnumeric);

defaultIniCapacity = 0;
addParameter(p, "IniCapacity", defaultIniCapacity, @isnumeric);

defaultPvIst = (park == "A") * 750 + (park == "C") * 600;
addParameter(p, "PvIst", defaultPvIst, @isnumeric);

defaultWIst = (park == "B") * 1000 + (park == "C") * 500;
addParameter(p, "WIst", defaultWIst, @isnumeric);

defaultLoadInc = 1;
addParameter(p, "LoadInc", defaultLoadInc, @isnumeric);

parse(p, varargin{:});

dataPath = p.Results.DataPath;
resultsPath = p.Results.ResultsPath;
resultsName = p.Results.ResultsName;
Initialize = p.Results.Initialize;
quiet = p.Results.Quiet;
capacity = p.Results.Capacity * 0.8;
power = p.Results.Power;
iniCapacity = p.Results.IniCapacity;
pvIst = p.Results.PvIst;
wiIst = p.Results.WIst;
loadInc = p.Results.LoadInc;

```

```

resultsFile = resultsPath + resultsName;
loadDataFile = dataPath + "附件 1: 各园区典型日负荷数据.xlsx";
generationDataFile = dataPath + "附件 2: 各园区典型日风光发电数据.xlsx";

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

%% 园区单独化设置
switch park
    case "A"
        resultsRange = "B2:B25";
        resultsRange2 = "B2:B26";
        pvGen = readmatrix(generationDataFile, "Range", "B4:B27") .* pvIst;
        wGen = zeros(24, 1);
    case "B"
        resultsRange = "C2:C25";
        resultsRange2 = "C2:C26";
        pvGen = zeros(24, 1);
        wGen = readmatrix(generationDataFile, "Range", "C4:C27") .* wIst;
    case "C"
        resultsRange = "D2:D25";
        resultsRange2 = "D2:D26";
        pvGen = readmatrix(generationDataFile, "Range", "D4:D27") .* pvIst;
        wGen = readmatrix(generationDataFile, "Range", "E4:E27") .* wIst;
    otherwise
        fprintf("[WARNING] 园区指定不正确! 将仅执行初始化功能! \n");
        Initialize = true;
end

%% 初始化非数据部分
if (~quiet && Initialize)
    % 时间
    time = strings(24, 1);
    for i = 0:23
        time(i + 1) = sprintf("%02d:00", i);
    end
    writematrix(time, resultsFile, "Sheet", "购电量", "Range", "A2:A25");
    writematrix(time, resultsFile, "Sheet", "弃风弃光电量", "Range", "A2:A25");
    writematrix(time, resultsFile, "Sheet", "购电成本", "Range", "A2:A25");
    writematrix(time, resultsFile, "Sheet", "单位电量平均供电成本", "Range",
"A2:A25");
    writematrix([time; "00:00"], resultsFile, "Sheet", "储电量", "Range",
"A2:A26");

```



```

    title = ["时间 (h)" "园区 A 购电量 (kWh)" "园区 B 购电量 (kWh)" "园区 C 购电量 (kWh)"];
    writematrix(title, resultsFile, "Sheet", "购电量", "Range", "A1:D1");
    title = ["时间 (h)" "园区 A 弃风弃光电量 (kWh)" "园区 B 弃风弃光电量 (kWh)" "园区 C 弃风弃光电量 (kWh)"];
    writematrix(title, resultsFile, "Sheet", "弃风弃光电量", "Range", "A1:D1");
    title = ["时间 (h)" "园区 A 购电成本 (元)" "园区 B 购电成本 (元)" "园区 C 购电成本 (元)"];
    writematrix(title, resultsFile, "Sheet", "购电成本", "Range", "A1:D1");
    title = ["时间 (h)" "园区 A 单位电量平均供电成本 (元)" "园区 B 单位电量平均供电成本 (元)" "园区 C 单位电量平均供电成本 (元)"];
    writematrix(title, resultsFile, "Sheet", "单位电量平均供电成本", "Range", "A1:D1");
    title = ["时间 (h)" "园区 A 储电量 (kWh)" "园区 B 储电量 (kWh)" "园区 C 储电量 (kWh)"];
    writematrix(title, resultsFile, "Sheet", "储电量", "Range", "A1:D1");
end

%% 读取数据
if ~exist("resultsRange", "var")
    return;
end
load = readmatrix(loadDataFile, "Range", resultsRange) * loadInc;

%% 购电量 & 弃风弃光电量
% ph -> per hour, 每小时
% t -> total, 总
% pur -> purchase, (从电网)购买(的电量)
% aban -> abandon, 放弃, 即弃(风弃光电量)
% diffLG -> difference between load and power generation, 负载发电差

storage = [iniCapacity; zeros(24, 1)];
phPur = zeros(24, 1);
phAban = zeros(24, 1);

diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电, 可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
                storage(i + 1) = capacity;
            end
        end
    end
end

```

```

else % 充电功率受限制
    if storage(i) + power * 0.95 < capacity % 不能充满
        phAban(i) = - diffLG(i) - power;
        storage(i + 1) = storage(i) + power * 0.95;
    else % 充满
        phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
        storage(i + 1) = capacity;
    end
end
else % 负载大于等于发电，可以放电
    if (power > diffLG(i)) % 放电功率足够
        if (storage(i) * 0.95 > diffLG(i)) % 储能足够
            storage(i + 1) = storage(i) - diffLG(i) / 0.95;
        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    else % 放电功率不足
        if (storage(i) * 0.95 > power) % 储能足够
            phPur(i) = diffLG(i) - power; % 购电
            storage(i + 1) = storage(i) - power / 0.95;
        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    end
end
end
storage = storage + 0.1 * p.Results.Capacity;

if ~quiet, writematrix(phPur, resultsFile, "Sheet", "购电量", "Range",
resultsRange); end
tPur = sum(phPur);

if ~quiet, writematrix(phAban, resultsFile, "Sheet", "弃风弃光电量", "Range",
resultsRange); end
tAban = sum(phAban);

if ~quiet, writematrix(storage, resultsFile, "Sheet", "储电量", "Range",
resultsRange2); end
aveStorage = sum(storage) / 24;

%% 购电成本
% ph -> per hour, 每小时

```

```

% t -> total, 总
% purCost -> the cost of purchasing, 购(电)成本
% 考虑到实际情况, 我们认为, 弃风弃光仍正常发电, 即仍计入成本

phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
if ~quiet, writematrix(phPurCost, resultsFile, "Sheet", "购电成本", "Range",
resultsRange); end
tPurCost = sum(phPurCost);

%% 单位电量平均供电成本
% ph -> per hour, 每小时
% t -> total, 总
% aveCostPkw 每度电平均成本
% 考虑到实际情况, 我们认为, 有效电量为此时的负载耗电量

phAveCostPkw = phPurCost ./ load;
if ~quiet, writematrix(phAveCostPkw, resultsFile, "Sheet", "单位电量平均供电成本",
"Range", resultsRange); end
tAveCostPkw = tPurCost / sum(load);

%% 打印结果
if ~quiet
    fprintf("园区%s (光伏装机容量%dkw, 风电装机容量%dkw) :\n", park, pvIst, wIst);
    fprintf(" 典型日购电量, %.2fkWh;\n", tPur);
    fprintf(" 典型日弃风弃光电量, %.2fkWh;\n", tAban);
    fprintf(" 典型日购电成本, %.2f 元;\n", tPurCost);
    fprintf(" 典型日单位电量平均供电成本, %f 元;\n", tAveCostPkw);
    fprintf(" 典型日日均储能, %.2fkWh, 结束时储能, %.2fkWh;\n", aveStorage,
storage(25));
end

%% 返回结果
switch nargout
    case 1
        varargout{1} = tAveCostPkw;
end

end

```

7.1.2 calU.m

```

% 计算单个园区的购电量、弃风弃光电量、总供电成本和单位电量平均供电成本
function varargout = calU(varargin)
%% 输入参数处理

```

```

p = inputParser;

defaultDataPath = "../Data/";
addParameter(p, "DataPath", defaultDataPath, @isstring);

defaultResultsPath = "../Results/";
addParameter(p, "ResultsPath", defaultResultsPath, @isstring);

defaultResultsName = "results.xlsx";
addParameter(p, "ResultsName", defaultResultsName, @isstring);

defaultQuiet = false;
addParameter(p, "Quiet", defaultQuiet, @islogical);

defaultCapacity = 0;
addParameter(p, "Capacity", defaultCapacity, @isnumeric);

defaultPower = 0;
addParameter(p, "Power", defaultPower, @isnumeric);

defaultIniCapacity = 0;
addParameter(p, "IniCapacity", defaultIniCapacity, @isnumeric);

defaultPvIst = [750 600];
addParameter(p, "PvIst", defaultPvIst, @isnumeric);

defaultWIst = [1000 500];
addParameter(p, "WIst", defaultWIst, @isnumeric);

defaultLoadInc = 1;
addParameter(p, "LoadInc", defaultLoadInc, @isnumeric);

parse(p, varargin{:});

dataPath = p.Results.DataPath;
resultsPath = p.Results.ResultsPath;
resultsName = p.Results.ResultsName;
quiet = p.Results.Quiet;
capacity = p.Results.Capacity * 0.8;
power = p.Results.Power;
iniCapacity = p.Results.IniCapacity;
pvIst = p.Results.PvIst;
wiIst = p.Results.WIst;
loadInc = p.Results.LoadInc;

```

```

resultsFile = resultsPath + resultsName;
loadDataFile = dataPath + "附件 1: 各园区典型日负荷数据.xlsx";
generationDataFile = dataPath + "附件 2: 各园区典型日风光发电数据.xlsx";

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

%% 读入发电量

% A 园区光伏 + C 园区光伏
pvGen = readmatrix(generationDataFile, "Range", "B4:B27") .* pvIst(1) + ...
    readmatrix(generationDataFile, "Range", "D4:D27") .* pvIst(2);

% B 园区风能 + C 园区风能
wGen = readmatrix(generationDataFile, "Range", "C4:C27") .* wIst(1) + ...
    readmatrix(generationDataFile, "Range", "E4:E27") .* wIst(2);

%% 初始化非数据部分
if (~quiet)
    % 时间
    time = strings(24, 1);
    for i = 0:23
        time(i + 1) = sprintf("%02d:00", i);
    end
    writematrix([time; "00:00"], resultsFile, "Range", "A2:A26");

    title = ["时间 (h)" "储电量 (kWh)" "购电量 (kWh)" "弃风弃光电量 (kWh)" "购电成本 (元)" "单位电量平均供电成本 (元)"];
    writematrix(title, resultsFile, "Range", "A1:F1");
end

%% 读取数据
% 按行求和得到联合园区负载

load = sum(readmatrix(loadDataFile, "Range", "B2:D25"), 2) * loadInc;

%% 购电量 & 弃风弃光电量
% ph -> per hour, 每小时
% t -> total, 总
% pur -> purchase, (从电网)购买(的电量)
% aban -> abandon, 放弃, 即弃(风弃光电量)
% diffLG -> difference between load and power generation, 负载发电差

storage = [iniCapacity; zeros(24, 1)];

```

```

phPur = zeros(24, 1);
phAban = zeros(24, 1);

diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电，可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
                storage(i + 1) = capacity;
            end
        else % 充电功率受限制
            if storage(i) + power * 0.95 < capacity % 不能充满
                phAban(i) = - diffLG(i) - power;
                storage(i + 1) = storage(i) + power * 0.95;
            else % 充满
                phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
                storage(i + 1) = capacity;
            end
        end
    else % 负载大于等于发电，可以放电
        if (power > diffLG(i)) % 放电功率足够
            if (storage(i) * 0.95 > diffLG(i)) % 储能足够
                storage(i + 1) = storage(i) - diffLG(i) / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            end
        else % 放电功率不足
            if (storage(i) * 0.95 > power) % 储能足够
                phPur(i) = diffLG(i) - power; % 购电
                storage(i + 1) = storage(i) - power / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            end
        end
    end
end
storage = storage + 0.1 * p.Results.Capacity;

if ~quiet, writematrix(storage, resultsFile, "Range", "B2:B26"); end
aveStorage = sum(storage) / 24;

if ~quiet, writematrix(phPur, resultsFile, "Range", "C2:C25"); end

```

```

tPur = sum(phPur);

if ~quiet, writematrix(phAban, resultsFile, "Range", "D2:D25"); end
tAban = sum(phAban);

%% 购电成本
% ph -> per hour, 每小时
% t -> total, 总
% purCost -> the cost of purchasing, 购(电)成本
% 考虑到实际情况, 我们认为, 弃风弃光仍正常发电, 即仍计入成本

phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
if ~quiet, writematrix(phPurCost, resultsFile, "Range", "E2:E25"); end
tPurCost = sum(phPurCost);

%% 单位电量平均供电成本
% ph -> per hour, 每小时
% t -> total, 总
% aveCostPkw 每度电平均成本
% 考虑到实际情况, 我们认为, 有效电量为此时的负载耗电量

phAveCostPkw = phPurCost ./ load;
if ~quiet, writematrix(phAveCostPkw, resultsFile, "Range", "F2:F25"); end
tAveCostPkw = tPurCost / sum(load);

%% 打印结果
if ~quiet
    fprintf("联合园区（总光伏装机容量%d = %d + %dkW, 总风电装机容量%d = %d\n", sum(pvIst), pvIst(1), pvIst(2), sum(wIst), wIst(1), wIst(2)));
    fprintf(" 典型日购电量, %.2fkWh;\n", tPur);
    fprintf(" 典型日弃风弃光电量, %.2fkWh;\n", tAban);
    fprintf(" 典型日购电成本, %.2f 元;\n", tPurCost);
    fprintf(" 典型日单位电量平均供电成本, %f 元;\n", tAveCostPkw);
    fprintf(" 典型日日均储能, %.2fkWh, 结束时储能, %.2fkWh;\n", aveStorage, storage(25));
end

%% 返回结果
switch nargout
    case 1
        varargout{1} = tAveCostPkw;
end

end
end

```

7.1.3 Q1_3.m

% 问题 1: 各园区独立运营储能配置方案及其经济性分析

% (3) 假设风光荷功率波动特性保持上述条件不变, 你认为 50kW/100kWh 的

% 方案是否最优? 若是, 请给出分析依据; 若不是, 请制定各园区最优的储能功率、

% 容量配置方案, 论证所制定方案的优越性。

% 考虑了日初始容量的问题, 不过由于遗传算法的特性, 这个限制可能更多的作用是使得

% 每日结束时的容量应该为 10% (最低允许值)

% 由于考虑的是典型日, 显然有每天的始末状态应当一致, 而实际运行中可能会出现由于

% 功率限制使得前一天晚上电量无法耗尽而在后一天凌晨使用的情况, 所以该问题需要考虑

% 由于遗传算法的特性, 可能需要多次运行才能得到论文提供的最优解

```
function Q1_3()
```

```
%% 清理
```

```
if (nargin == 0)
```

```
    clc, close all;
```

```
end
```

```
%% 初始化
```

```
solver = "g"; % 求解器类型, 调试用, 可选 g(a)和 f(mincon)
```

```
% 测试发现, fmincon 比 ga 更容易陷入局部最优解, 且前者无法限制为整数值, 于是使用 ga
```

```
park = "A"; % 园区选择, 由于只有 3 个园区, 不做 ALL 选项了
```

```
dataPath = "../Data/";
```

```
loadDataFile = dataPath + "附件 1: 各园区典型日负荷数据.xlsx";
```

```
generationDataFile = dataPath + "附件 2: 各园区典型日风光发电数据.xlsx";
```

```
resultsName = "问题 1(3): 配置储能时最优解各园区详细数据.xlsx";
```

```
[pvGen, wGen, load] = readData(park, generationDataFile, loadDataFile);
```

```
%% 作为非线性优化问题进行求解
```

```
% 对每个园区有三个决策变量, x(1)代表容量, x(2)代表功率, x(3)代表日初始容量
```

```
% 目标函数
```

```
fitnessfcn = @(x) calc(x, pvGen, wGen, load);
```

```
% 不等式约束  $A * x \leq b$ 
```

```
% capacity  $\geq$  power  $\rightarrow x(1) \geq x(2) \rightarrow -x(1) + x(2) \leq 0$ 
```

```
A = [];
```

```
b = [];
```

```
% 非线性约束 (初始容量约束)
```

```
nonlcon = @(x) initialCapacity(x, pvGen, wGen, load);
```



```

% 变量界限
lb = [1; 1; 0];
ub = Inf * ones(3, 1);

% 整数约束
intcon = [1 2];

if solver == "g" % 调用 ga 求解
    % 选项设置
    options = optimoptions("ga", "ConstraintTolerance", 1e-6);
    % options = optimoptions("ga", "Display", "iter", "ConstraintTolerance", 1e-6);
    [x, fval] = ga(fitnessfcn, 3, A, b, [], [], lb, ub, nonlcon, intcon, options);
    fprintf("容量: %dkWh; 功率: %dkW; 初始容量: %.2fkWh; \n 单位电量成本: %f 元\n",
x(1), x(2), x(3) + 0.1 * x(1), fval);
else % 调用 fmincon 函数求解
    x0 = [1; 1; 0];
    [x, fval] = fmincon(fitnessfcn, x0, A, b, [], [], lb, ub, nonlcon);
    fprintf("容量: %f(%d)kWh; 功率: %f(%d)kW; 初始容量: %fkWh; \n 单位电量成本: %f 元\n",
x(1), round(x(1)), x(2), round(x(2)), x(3), fval);
end

% 数据写入文件
cal(park, "ResultsName", resultsName, "Capacity", x(1), "Power", x(2));
cal("0", "ResultsName", resultsName);

end

%% 日初始容量约束
function [c, ceq] = initialCapacity(x, pvGen, wGen, load)
ceq = [];

capacity = x(1) * 0.8;
power = x(2);
storage = [x(3); zeros(24, 1)];
diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电, 可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        end
    end
end

```

```

else % 充电功率受限制
    if storage(i) + power * 0.95 < capacity % 不能充满
        storage(i + 1) = storage(i) + power * 0.95;
    else % 充满
        storage(i + 1) = capacity;
    end
end
else % 负载大于等于发电，可以放电
    if (power > diffLG(i)) % 放电功率足够
        if (storage(i) * 0.95 > diffLG(i)) % 储能足够
            storage(i + 1) = storage(i) - diffLG(i) / 0.95;
        end
    else % 放电功率不足
        if (storage(i) * 0.95 > power) % 储能足够
            storage(i + 1) = storage(i) - power / 0.95;
        end
    end
end
end

c = abs(storage(25) - storage(1));

end

%% 计算函数 (从 cal 函数精简而来，防止反复 load 降低效率)
function tAveCostPkw = calc(x, pvGen, wGen, load)

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

capacity = x(1) * 0.8;
power = x(2);

% 购电量 & 弃风弃光电量
storage = [x(3); zeros(24, 1)];
phPur = zeros(24, 1);
phAban = zeros(24, 1);

diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电，可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            end
        end
    end
end

```

```

        else % 充满
            phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
            storage(i + 1) = capacity;
        end
    else % 充电功率受限制
        if storage(i) + power * 0.95 < capacity % 不能充满
            phAban(i) = - diffLG(i) - power;
            storage(i + 1) = storage(i) + power * 0.95;
        else % 充满
            phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
            storage(i + 1) = capacity;
        end
    end
end
else % 负载大于等于发电，可以放电
    if (power > diffLG(i)) % 放电功率足够
        if (storage(i) * 0.95 > diffLG(i)) % 储能足够
            storage(i + 1) = storage(i) - diffLG(i) / 0.95;
        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    else % 放电功率不足
        if (storage(i) * 0.95 > power) % 储能足够
            phPur(i) = diffLG(i) - power; % 购电
            storage(i + 1) = storage(i) - power / 0.95;
        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    end
end
end
end

% 购电成本
phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
tPurCost = sum(phPurCost);

% 单位电量平均供电成本
tAveCostPkw = tPurCost / sum(load);

end

%% 数据读入函数（将数据读入函数独立）
function [pvGen, wGen, load] = readData(park, generationDataFile, loadDataFile)

```

```

switch park
    case "A"
        pvGen = readmatrix(generationDataFile, "Range", "B4:B27") .* 750;
        wGen = zeros(24, 1);
        load = readmatrix(loadDataFile, "Range", "B2:B25");
    case "B"
        pvGen = zeros(24, 1);
        wGen = readmatrix(generationDataFile, "Range", "C4:C27") .* 1000;
        load = readmatrix(loadDataFile, "Range", "C2:C25");
    case "C"
        pvGen = readmatrix(generationDataFile, "Range", "D4:D27") .* 600;
        wGen = readmatrix(generationDataFile, "Range", "E4:E27") .* 500;
        load = readmatrix(loadDataFile, "Range", "D2:D25");
    otherwise
        fprintf("[WARNING] 园区指定不正确! \n");
        pvGen = -1;
        wGen = -1;
        load = -1;
end

end

```

7.1.4 Q2_2.m

% 问题 2: 联合园区储能配置方案及其经济性分析
 % (2) 假设风光荷功率波动特性保持上述条件不变, 制定联合园区的总储能
 % 最优配置方案, 给出储能运行策略及购电计划, 分析其经济性;

% 由于遗传算法的特性, 可能需要多次运行才能得到论文提供的最优解

```

function Q2_2()
%% 清理
if (nargin == 0)
    clc, close all;
end

%% 初始化
dataPath = "../Data/";
loadDataFile = dataPath + "附件 1: 各园区典型日负荷数据.xlsx";
generationDataFile = dataPath + "附件 2: 各园区典型日风光发电数据.xlsx";
resultsName = "问题 2(2): 配置储能时最优解联合园区详细数据.xlsx";
[pvGen, wGen, load] = readData(generationDataFile, loadDataFile);

%% 作为非线性优化问题进行求解

```

```

% 对每个园区有三个决策变量，x(1)代表容量，x(2)代表功率，x(3)代表日初始容量

% 目标函数
fitnessfcn = @(x) calc(x, pvGen, wGen, load);

% 不等式约束  $A * x \leq b$ 
% capacity  $\geq$  power  $\rightarrow x(1) \geq x(2) \rightarrow -x(1) + x(2) \leq 0$ 
A = [-1 1 0];
b = 0;

% 非线性约束（初始容量约束）
nonlcon = @(x) initialCapacity(x, pvGen, wGen, load);

% 变量界限
lb = [1; 1; 0];
ub = Inf * ones(3, 1);

% 整数约束
intcon = [1 2];

% 选项设置
options = optimoptions("ga", "ConstraintTolerance", 1e-6);
% options = optimoptions("ga", "Display", "iter", "ConstraintTolerance", 1e-6);

% 调用 ga 求解
[x, fval] = ga(fitnessfcn, 3, A, b, [], [], lb, ub, nonlcon, intcon, options);
fprintf("容量: %dkWh; 功率: %dkw; 初始容量: %.2fkWh; \n 单位电量成本: %f 元\n", x(1),
x(2), x(3) + 0.1 * x(1), fval);

% 数据写入文件
calU("ResultsName", resultsName, "Capacity", x(1), "Power", x(2));

end

%% 日初始容量约束
function [c, ceq] = initialCapacity(x, pvGen, wGen, load)
ceq = [];

capacity = x(1) * 0.8;
power = x(2);
storage = [x(3); zeros(24, 1)];
diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电，可以充电

```

```

    if (power > -diffLG(i)) % 充电功率足够
        if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
            storage(i + 1) = storage(i) - diffLG(i) * 0.95;
        else % 充满
            storage(i + 1) = capacity;
        end
    else % 充电功率受限制
        if storage(i) + power * 0.95 < capacity % 不能充满
            storage(i + 1) = storage(i) + power * 0.95;
        else % 充满
            storage(i + 1) = capacity;
        end
    end
else % 负载大于等于发电，可以放电
    if (power > diffLG(i)) % 放电功率足够
        if (storage(i) * 0.95 > diffLG(i)) % 储能足够
            storage(i + 1) = storage(i) - diffLG(i) / 0.95;
        end
    else % 放电功率不足
        if (storage(i) * 0.95 > power) % 储能足够
            storage(i + 1) = storage(i) - power / 0.95;
        end
    end
end
end

c = abs(storage(25) - storage(1));

end

%% 计算函数 (从 cal 函数精简而来，防止反复 load 降低效率)
function tAveCostPkw = calc(x, pvGen, wGen, load)

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

capacity = x(1) * 0.8;
power = x(2);

% 购电量 & 弃风弃光电量
storage = [x(3); zeros(24, 1)];
phPur = zeros(24, 1);
phAban = zeros(24, 1);

```

```

diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电，可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
                storage(i + 1) = capacity;
            end
        else % 充电功率受限制
            if storage(i) + power * 0.95 < capacity % 不能充满
                phAban(i) = - diffLG(i) - power;
                storage(i + 1) = storage(i) + power * 0.95;
            else % 充满
                phAban(i) = - diffLG(i) - (capacity - storage(i)) / 0.95; % 弃电
                storage(i + 1) = capacity;
            end
        end
    else % 负载大于等于发电，可以放电
        if (power > diffLG(i)) % 放电功率足够
            if (storage(i) * 0.95 > diffLG(i)) % 储能足够
                storage(i + 1) = storage(i) - diffLG(i) / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        else % 放电功率不足
            if (storage(i) * 0.95 > power) % 储能足够
                phPur(i) = diffLG(i) - power; % 购电
                storage(i + 1) = storage(i) - power / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        end
    end
end

% 购电成本
phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
tPurCost = sum(phPurCost);

```

```

% 单位电量平均供电成本
tAveCostPkw = tPurCost / sum(load);

end

%% 数据读入函数（将数据读入函数独立）
function [pvGen, wGen, load] = readData(generationDataFile, loadDataFile)

% A 园区光伏 + C 园区光伏
pvGen = readmatrix(generationDataFile, "Range", "B4:B27") .* 750 + ...
        readmatrix(generationDataFile, "Range", "D4:D27") .* 600;

% B 园区风能 + C 园区风能
wGen = readmatrix(generationDataFile, "Range", "C4:C27") .* 1000 + ...
        readmatrix(generationDataFile, "Range", "E4:E27") .* 500;

% 按行求和得到联合园区负载
load = sum(readmatrix(loadDataFile, "Range", "B2:D25"), 2);

end

```

7.1.5 Q3_1_1.m

% 问题 3：园区风、光、储能的协调配置方案及其经济性分析

% （1）分别按各园区独立运营、联合运营制定风光储协调配置方案：

% 由于遗传算法的特性，可能需要多次运行才能得到论文提供的最优解

```

function Q3_1_1()
%% 清理
if (nargin == 0)
    clc, close all;
end

%% 初始化
park = "A"; % 园区选择，由于只有 3 个园区，不做 ALL 选项了
dataPath = "../Data/";
loadDataFile = dataPath + "附件 1：各园区典型日负荷数据.xlsx";
generationDataFile = dataPath + "附件 2：各园区典型日风光发电数据.xlsx";
resultsName = "问题 3(1)：最优风光储独立运营各园区详细数据.xlsx";
[pvNGen, wNGen, load] = readData(park, generationDataFile, loadDataFile);

%% 作为非线性优化问题进行求解
% 对每个园区有五个决策变量，x(1)代表储能容量，x(2)代表储能功率，x(3)代表日初始容量，x(4)
代表光伏装机容量，x(5)代表风电装机容量

```



```

% 目标函数
fitnessfcn = @(x) calc(x, pvNGen, wNGen, load);

% 不等式约束 A * x <= b
% capacity >= power --> x(1) >= x(2) --> -x(1) + x(2) <= 0
A = [-1 1 0 0 0];
b = 0;

% 非线性约束（初始容量约束、投资回报期约束）
nonlcon = @(x) nonLCon(x, pvNGen, wNGen, load);

% 变量界限
lb = [1; 1; 0; 0; 0];
ub = Inf * ones(5, 1);

if park == "A" % A 园区只能光伏装机
    ub(5) = 0;
elseif park == "B" % B 园区只能风电装机
    ub(4) = 0;
end

% 整数约束
intcon = [1 2 4 5];

% 选项设置
options = optimoptions("ga", "ConstraintTolerance", 1e-6);
% options = optimoptions("ga", "Display", "iter", "ConstraintTolerance", 1e-6);

% 调用 ga 求解
[x, fval] = ga(fitnessfcn, 5, A, b, [], [], lb, ub, nonlcon, intcon, options);
fprintf('容量: %dkWh; 功率: %dkW; 初始容量: %.2fkWh; 光伏装机容量: %dkW; 风电装机容量: %dkW; \n 单位电量成本: %f 元\n', x(1), x(2), x(3) + 0.1 * x(1), x(4), x(5), fval);

% 数据写入文件
cal(park, "ResultsName", resultsName, "Capacity", x(1), "Power", x(2), "IniCapacity", x(3), "PvIst", x(4), "wIst", x(5), "LoadInc", 1.5);
cal("0", "ResultsName", resultsName);

end

%% 非线性约束
function [c, ceq] = nonLCon(x, pvNGen, wNGen, load)
ceq = [];

```

```

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

capacity = x(1) * 0.8;
power = x(2);
storage = [x(3); zeros(24, 1)];
pvIst = x(4);
wIst = x(5);

pvGen = pvIst .* pvNGen;
wGen = wIst .* wNGen;

% 购电量 & 弃风弃光电量
phPur = zeros(24, 1);
diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电, 可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        else % 充电功率受限制
            if storage(i) + power * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) + power * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        end
    else % 负载大于等于发电, 可以放电
        if (power > diffLG(i)) % 放电功率足够
            if (storage(i) * 0.95 > diffLG(i)) % 储能足够
                storage(i + 1) = storage(i) - diffLG(i) / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        else % 放电功率不足
            if (storage(i) * 0.95 > power) % 储能足够
                phPur(i) = diffLG(i) - power; % 购电
                storage(i + 1) = storage(i) - power / 0.95;
            end
        end
    end
end

```

```

        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    end
end
end

% 购电成本
phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
tPurCost = sum(phPurCost);

c = [ ...
    abs(storage(25) - storage(1)); ... 日初始容量 --> abs(次日初始 - 首日初始) <= 0
    (2500 * pvIst + 3000 * wIst) - 365 * 5 * (sum(load) - tPurCost); ... 投资回报
    周期 --> 投资金额 - 5年 * (原购电成本 - 现购电成本) <= 0
    ];

end

%% 计算函数 (从 cal 函数精简而来, 防止反复 load 降低效率)
function tAveCostPkw = calc(x, pvNGen, wNGen, load)

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

capacity = x(1) * 0.8;
power = x(2);
storage = [x(3); zeros(24, 1)];
pvIst = x(4);
wIst = x(5);

pvGen = pvIst .* pvNGen;
wGen = wIst .* wNGen;

% 购电量 & 弃风弃光电量
phPur = zeros(24, 1);
diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电, 可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            end
        end
    end
end

```

```

        else % 充满
            storage(i + 1) = capacity;
        end
    else % 充电功率受限制
        if storage(i) + power * 0.95 < capacity % 不能充满
            storage(i + 1) = storage(i) + power * 0.95;
        else % 充满
            storage(i + 1) = capacity;
        end
    end
end
else % 负载大于等于发电，可以放电
    if (power > diffLG(i)) % 放电功率足够
        if (storage(i) * 0.95 > diffLG(i)) % 储能足够
            storage(i + 1) = storage(i) - diffLG(i) / 0.95;
        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    else % 放电功率不足
        if (storage(i) * 0.95 > power) % 储能足够
            phPur(i) = diffLG(i) - power; % 购电
            storage(i + 1) = storage(i) - power / 0.95;
        else % 储能不足
            phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
            storage(i + 1) = 0;
        end
    end
end
end

% 购电成本
phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
tPurCost = sum(phPurCost);

% 单位电量平均供电成本
tAveCostPkw = tPurCost / sum(load);

end

%% 数据读入函数（将数据读入函数独立）
function [pvNGen, wNGen, load] = readData(park, generationDataFile, loadDataFile)

switch park
    case "A"

```

```

        pvNGen = readmatrix(generationDataFile, "Range", "B4:B27");
        wNGen = zeros(24, 1);
        load = readmatrix(loadDataFile, "Range", "B2:B25") .* 1.5;
    case "B"
        pvNGen = zeros(24, 1);
        wNGen = readmatrix(generationDataFile, "Range", "C4:C27");
        load = readmatrix(loadDataFile, "Range", "C2:C25") .* 1.5;
    case "C"
        pvNGen = readmatrix(generationDataFile, "Range", "D4:D27");
        wNGen = readmatrix(generationDataFile, "Range", "E4:E27");
        load = readmatrix(loadDataFile, "Range", "D2:D25") .* 1.5;
    otherwise
        fprintf("[WARNING] 园区指定不正确! \n");
        pvNGen = -1;
        wNGen = -1;
        load = -1;
end

end

```

7.1.6 Q3_1_2.m

% 问题 3: 园区风、光、储能的协调配置方案及其经济性分析

% (1) 分别按各园区独立运营、联合运营制定风光储协调配置方案;

% 由于遗传算法的特性, 可能需要多次运行才能得到论文提供的最优解

```

function Q3_1_2()
%% 清理
if (nargin == 0)
    clc, close all;
end

%% 初始化
dataPath = "../Data/";
loadDataFile = dataPath + "附件 1: 各园区典型日负荷数据.xlsx";
generationDataFile = dataPath + "附件 2: 各园区典型日风光发电数据.xlsx";
resultsName = "问题 3(1): 最优风光储独立运营联合园区详细数据.xlsx";
[pvNGen, wNGen, load] = readData(generationDataFile, loadDataFile);

%% 作为非线性优化问题进行求解
% 对每个园区有七个决策变量, x(1)代表储能容量, x(2)代表储能功率, x(3)代表日初始容量, x(4)
% 代表 A 光伏装机容量, x(5)代表 C 光伏装机容量, x(6)代表 B 风电装机容量, x(7)代表 C 风电装机容量

```

```

% 目标函数
fitnessfcn = @(x) calc(x, pvNGen, wNGen, load);

% 不等式约束 A * x <= b
% capacity >= power --> x(1) >= x(2) --> -x(1) + x(2) <= 0
A = [-1 1 0 0 0 0 0];
b = 0;

% 非线性约束（初始容量约束、投资回报期约束）
nonlcon = @(x) nonLCon(x, pvNGen, wNGen, load);

% 变量界限
lb = [1; 1; 0; 0; 0; 0; 0];
ub = Inf * ones(7, 1);

% 整数约束
intcon = [1 2 4 5 6 7];

% 选项设置
options = optimoptions("ga", "ConstraintTolerance", 1e-6);
% options = optimoptions("ga", "Display", "iter", "ConstraintTolerance", 1e-6);

% 调用 ga 求解
[x, fval] = ga(fitnessfcn, 7, A, b, [], [], lb, ub, nonlcon, intcon, options);
fprintf("容量: %dkWh; 功率: %dkW; 初始容量: %.2fkWh; 光伏装机容量: %d/%dkW; 风电装机容量: %d/%dkW; \n 单位电量成本: %f 元\n", x(1), x(2), x(3) + 0.1 * x(1), x(4), x(5), x(6), x(7), fval);

% 数据写入文件
calU("ResultsName", resultsName, "Capacity", x(1), "Power", x(2), "IniCapacity", x(3), "PvIst", x(4:5), "wIst", x(6:7), "LoadInc", 1.5);

end

%% 非线性约束
function [c, ceq] = nonLCon(x, pvNGen, wNGen, load)
ceq = [];

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

capacity = x(1) * 0.8;
power = x(2);
storage = [x(3); zeros(24, 1)];

```

```

pvIst = x(4:5);
wIst = x(6:7);

pvGen = sum(pvIst .* pvNGen, 2);
wGen = sum(wIst .* wNGen, 2);

% 购电量 & 弃风弃光电量
phPur = zeros(24, 1);
diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电, 可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        else % 充电功率受限制
            if storage(i) + power * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) + power * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        end
    else % 负载大于等于发电, 可以放电
        if (power > diffLG(i)) % 放电功率足够
            if (storage(i) * 0.95 > diffLG(i)) % 储能足够
                storage(i + 1) = storage(i) - diffLG(i) / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        else % 放电功率不足
            if (storage(i) * 0.95 > power) % 储能足够
                phPur(i) = diffLG(i) - power; % 购电
                storage(i + 1) = storage(i) - power / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        end
    end
end
end
end

```

```

% 购电成本
phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
tPurCost = sum(phPurCost);

c = [ ...
    abs(storage(25) - storage(1)); ... 日初始容量 --> abs(次日初始 - 首日初始) <= 0
    (2500 * sum(pvIst) + 3000 * sum(wIst)) - 365 * 5 * (sum(load) - tPurCost); ...
    投资回报周期 --> 投资金额 - 5年 * (原购电成本 - 现购电成本) <= 0
    ];

end

%% 计算函数 (从 cal 函数精简而来, 防止反复 load 降低效率)
function tAveCostPkw = calc(x, pvNGen, wNGen, load)

netCost = 1;
pvCost = 0.4;
wCost = 0.5;

capacity = x(1) * 0.8;
power = x(2);
storage = [x(3); zeros(24, 1)];
pvIst = x(4:5);
wIst = x(6:7);

pvGen = sum(pvIst .* pvNGen, 2);
wGen = sum(wIst .* wNGen, 2);

% 购电量 & 弃风弃光电量
phPur = zeros(24, 1);
diffLG = load - (pvGen + wGen);
for i = 1:24
    if (diffLG(i) < 0) % 负载小于发电, 可以充电
        if (power > -diffLG(i)) % 充电功率足够
            if storage(i) - diffLG(i) * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) - diffLG(i) * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        else % 充电功率受限制
            if storage(i) + power * 0.95 < capacity % 不能充满
                storage(i + 1) = storage(i) + power * 0.95;
            else % 充满
                storage(i + 1) = capacity;
            end
        end
    end
end

```



```

        end
    end
    else % 负载大于等于发电，可以放电
        if (power > diffLG(i)) % 放电功率足够
            if (storage(i) * 0.95 > diffLG(i)) % 储能足够
                storage(i + 1) = storage(i) - diffLG(i) / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        else % 放电功率不足
            if (storage(i) * 0.95 > power) % 储能足够
                phPur(i) = diffLG(i) - power; % 购电
                storage(i + 1) = storage(i) - power / 0.95;
            else % 储能不足
                phPur(i) = diffLG(i) - storage(i) * 0.95; % 购电
                storage(i + 1) = 0;
            end
        end
    end
end

% 购电成本
phPurCost = phPur .* netCost + pvGen .* pvCost + wGen .* wCost + ...
    (capacity * 1800 + power * 800) / (365 * 10 * 24);
tPurCost = sum(phPurCost);

% 单位电量平均供电成本
tAveCostPkw = tPurCost / sum(load);

end

%% 数据读入函数（将数据读入函数独立）
function [pvNGen, wNGen, load] = readData(generationDataFile, loadDataFile)

% A 园区光伏 + C 园区光伏
pvNGen = [readmatrix(generationDataFile, "Range", "B4:B27")
    readmatrix(generationDataFile, "Range", "D4:D27")];

% B 园区风能 + C 园区风能
wNGen = [readmatrix(generationDataFile, "Range", "C4:C27")
    readmatrix(generationDataFile, "Range", "E4:E27")];

% 按行求和得到联合园区负载
load = sum(readmatrix(loadDataFile, "Range", "B2:D25"), 2) .* 1.5;

```

end