DW分解求解最优潮流问题

Code

电网模型

构建基于Power Transfer Distribution Factor(PTDF)的直流潮流模型，对Pandapower自带的case 9与case 118求解最优潮流问题。case 9包含3个发电机，位于节点1、2、3，1为slack节点；case 118包含54个发电机。优化目标为发电机发电成本之和最小。

- 节点集合 $B = \{1, \ldots, |B|\}$，机组集合 $G = \{1, \ldots, |G|\}$。
- 决策变量为每台机组 g 的出力 x_g，可行域：

$$X_g = \{\, x_g \mid P_g^{\min} \le x_g \le P_g^{\max} \,\}.$$

- 目标函数为总发电成本：

$$\min \sum_{g \in G} c_g \, x_g,$$

- 由PTDF矩阵构造的系统耦合约束。令负荷为向量 Ld，机组—母线映射为 C，线路上限为 $f^{\max}$。则线路潮流：

$$f = \mathrm{PTDF}\,(C\,x - Ld) \in [-f^{\max}, f^{\max}],$$

- 能量平衡约束：

$$\sum_{g \in G} x_g = \sum_{b \in B} Ld_b,$$

把所有耦合约束整理成标准线性形式：

$$A_{\mathrm{eq}}\, x = b_{\mathrm{eq}}, \qquad A_{\mathrm{le}}\, x \le b_{\mathrm{le}},$$

其中 $x = [x_g]_{g \in G}$，矩阵每一列对应一台机组对各耦合约束的系数，由 PTDF 与机组-母线映射计算得到。

DW分解

> 每个子块对应一个发电机及其约束。主问题约束包括Power balance constraints和Power flow constraints，子问题约束为generation constraints。

原问题

- 目标：

$$\min \sum_{g \in G} c_g \, x_g.$$

- 局部约束：

$$x_g \in X_g,\ \forall g \in G.$$

- 全局耦合：

$$A_{\text{eq}}\, x = b_{\text{eq}}, \qquad A_{\text{le}}\, x \le b_{\text{le}}.$$

这是经典的块对角可分 + 少量耦合约束的结构，适合用Dantzig–Wolfe（DW）分解。

顶点表示与列变量

由于每个 X_g 有界且为多面体，可写成其极点的凸组合
令 X_g 的极点集合为 $\{v_g^k\}_{k\in\mathcal{K}_g}$，则

$$x_g = \sum_{k\in\mathcal{K}_g} \lambda_{gk}\, v_g^k, \qquad \sum_{k\in\mathcal{K}_g} \lambda_{gk} = 1, \qquad \lambda_{gk} \ge 0.$$

> 在本例（仅有机组出力上下限）中，X_g 的极点只有两个：$v_g^1 = P_g^{\min}$ 与 $v_g^2 = P_g^{\max}$；若以后加入爬坡、启停等局部线性约束，极点会增多，但模型形式不变。

限制主问题（RMP)

把 $x = \sum_{k\in\mathcal{K}_g} \lambda_{gk}$ 代入耦合约束，得到RMP：

$$\min \ \sum_{g\in G}\sum_{k\in\mathcal{K}_g} c_g\, v_g^k\, \lambda_{gk}$$

$$\text{s.t.} \quad A_{\text{eq}}\Big(\sum_{g\in G}\sum_{k\in\mathcal{K}_g} e_g\, v_g^k\, \lambda_{gk}\Big) = b_{\text{eq}}$$

$$A_{\text{le}}\Big(\sum_{g\in G}\sum_{k\in\mathcal{K}_g} e_g\, v_g^k\, \lambda_{gk}\Big) \le b_{\text{le}}$$

$$\sum_{k\in\mathcal{K}_g} \lambda_{gk} = 1, \quad \forall g \in G$$

$$\lambda_{gk} \ge 0, \quad \forall g \in G,\ \forall k \in \mathcal{K}_g.$$

其中 e_g 表示在机组 g 所在列位置取 1 的标准基向量。

RMP 的对偶与对偶变量

- 对 A_{eq} 的等式约束设对偶为 π_{eq} （自由号）。
- 对 A_{le} 的不等式约束设对偶为 π_{le}（$\pi_{\text{le}} \ge 0$）。
- 对 λ_{gk} 的等式约束设对偶为 μ_g（自由号）。

把它们合并成一组向量 π ，并令 a_g 表示耦合矩阵对应机组 g 的整列系数。则列 v_g^k 的约化成本为

$$\bar{c}_g(v_g^k) = c_g\, v_g^k - a_g^\top \pi\, v_g^k - \mu_g.$$

子问题

对每个机组 g，在当前对偶 (π, μ) 下，解

$$\begin{aligned} \min\ \bar{c}_g(x_g) &= (c_g - a_g^\top \pi)\, x_g - \mu_g \\ \text{s.t.} \quad & x_g \in X_g. \end{aligned}$$

- 若最优值 < 0 ，则 $x_g^\star$ 对应的新列 $v_g^{\text{new}} = x_g^\star$ 加入 RMP。
- 若所有机组的最优值均 ≥ 0 ，则当前 RMP 已为原问题的最优解，算法停止。

代码求解

case 9

电网模型

▶ 显示代码

直接求解

▶ 显示代码

```
Objective = 362.000000
Total time: 0.000 seconds
```

矩阵分解

▶ 显示代码

DW分解求解

▶ 显示代码

```
Iter 01 | RMP obj=562.000000 | x=[ 50. 235.  30.] | RMP_rt=0.000000s
   SP g=0: rc=-2.000000e+02, cand pt=10.000000, SP_rt=0.000000s
     + add column to block 0: v=10.000000
   SP g=1: rc=-2.700000e+02, cand pt=10.000000, SP_rt=0.000000s
     + add column to block 1: v=10.000000
   SP g=2: rc=-2.000000e+01, cand pt=10.000000, SP_rt=0.000000s
     + add column to block 2: v=10.000000
Iter 02 | RMP obj=562.000000 | x=[ 50. 235.  30.] | RMP_rt=0.000000s
   SP g=0: rc=0.000000e+00, cand pt=10.000000, SP_rt=0.000000s
   SP g=1: rc=-2.470000e+02, cand pt=300.000000, SP_rt=0.000000s
     + add column to block 1: v=300.000000
   SP g=2: rc=-9.600000e+02, cand pt=270.000000, SP_rt=0.000000s
```

```
    + add column to block 2: v=270.000000
Iter 03 | RMP obj=362.000000 | x=[ 10.  35. 270.] | RMP_rt=0.000000s
   SP g=0: rc=0.000000e+00, cand pt=10.000000, SP_rt=0.000000s
   SP g=1: rc=0.000000e+00, cand pt=10.000000, SP_rt=0.000000s
   SP g=2: rc=1.421085e-14, cand pt=270.000000, SP_rt=0.000000s
No improving columns. Converged.

=== Runtimes (Gurobi Runtime) ===
RMP total: 0.000000 s
SP  total: 0.000000 s
ALL total: 0.000000 s

Objective = 362.000000
Number of iterations = 3
```

DW分解与直接求解的结果一致,时间差异不大.

case 118

电网模型

▶ 显示代码

直接求解

▶ 显示代码

```
Objective = 84840.000000
Total time: 0.001 seconds
```

矩阵分解

▶ 显示代码

DW分解求解

▶ 显示代码

```
No improving columns. Converged.

=== Runtimes (Gurobi Runtime) ===
RMP total: 0.006000 s
SP  total: 0.005000 s
ALL total: 0.011000 s

Objective = 84840.000000
Number of iterations = 3
```

DW分解与直接求解的结果一致。时间上，直接求解稍微快于DW分解。

> 在case 118以及另一线性规划算例中,直接求解都快于DW分解.

> 也有其他文献使用DW分解求解DCOPF问题 (Bastianel, Ergun, and Hertem 2022) ,用的是DCOPF模型(不计算PTDF),对generator output建一个子问题，对branch flow建一个子问题,使用JuMP求解.也得到直接求解快于DW分解的结论.
>
> 这可能是因为DW分解更适合大规模的,列>>行的问题,而DCOPF的列数与行数接近,因此无法凸显DW分解的优势.

为什么更慢（结构与算法层面）

- **耦合稠密**
  PTDF 形式下，每台机组的注入几乎影响所有线路潮流 ⇒ 耦合矩阵 A 的每列都很“满”。DW 的限制主问题（RMP）中，每新增一列都会出现在大量约束里，单纯形更新成本高、迭代慢。
- **子问题太简单**
  单期 DCOPF 的子问题常是“单变量 + 上下界”。理论最优点在端点即可判断，但 DW 仍要每一轮做“解 RMP + 解全部子问题（定价）”，调度与通信开销远大于把完整 LP 一次性交给求解器的成本。
- **退化严重、对偶震荡**
  电网潮流/容量约束常高度退化。DW 的对偶迭代（由 RMP 对偶驱动）易震荡、步长保守，若不做稳定化，往往需要更多轮迭代。
- **错失稀疏线性代数优势**
  PTDF 把 $B\theta$ 的稀疏关系“乘积化”，失去原始（角度变量）模型的稀疏性；而直接解 $B\theta$ 形式，商业求解器的预处理/分解会更高效。
- **实现层面开销**
  逐轮在 Python 端“建模→求解→加列→再求解”，频繁模型修改、热启动不充分、I/O/解释器开销，都会堆高总耗时；而“直接法”一次建模、一次求解，预处理可最大化发挥。

什么时候 DW 反而合适

- **超大规模、强可分结构**：多时段（耦合在少量互联约束）、大规模情景/鲁棒优化（情景独立、少量耦合）、或 MILP 的 **branch-and-price**。DW/列生成能让“按需生成列”，避免一次性建出天量变量。

References

Bastianel, G., H. Ergun, and D. Van Hertem. 2022. “Linearised Optimal Power Flow Problem Solution Using Dantzig - Wolfe Decomposition.” In *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 1–5. Novi Sad, Serbia: IEEE. https://doi.org/10.1109/ISGT-Europe54678.2022.9960529.