Project step 1: deterministic optimisation

S. Drake Siard
DTU 31792, Spring 2021

10 Jan 2021

1 Problem definitions

The convex optimization problem selected in step 0 was the day-ahead power market-clearing problem with transport constraints, following the formulation in [1, pg. 11].

For a set of generators $\{g\}$ with production costs $\{C_g\}$, demands $\{d\}$ with utility $\{U_d\}$, buses $\{n\}$, and connections $\{t_{n,m}\}$, denote:

- maximum generation for g as $\overline{P}_g^G$, maximum demand for d as $\overline{P}_d^D$, and maximum flow through connection $t_{n,m}$ as $F_{n,m}$
- the difference in voltage angle between buses n and m as $\theta_n - \theta_m$, the susceptance of their connection $t_{n,m}$ as $B_{n,m}$, and the (rough) approximation of flow between them as $f_{n,m} = B_{n,m}(\theta_n - \theta_m)$
- the set of generators and demands in bus n as Ψ_n, and the set of buses connected to bus n via transmission lines as Ω_n
- θ_{n_0} as the reference voltage angle of some arbitrary bus n_0

2 Primal problem

The primal problem then becomes the maximisation of social welfare (the sum of the producer and consumer surplus):

$$\max_{p_g^G, p_d^D, \theta_n} \sum_d U_d p_d^D - \sum_g C_g p_g^G \tag{2.1}$$

subject to:

$$0 \leq p_d^D \leq \overline{P}_d^D \qquad \forall d \tag{2.2}$$

$$0 \leq p_g^G \leq \overline{P}_g^G \quad \forall g \tag{2.3}$$

$$-F_{n,m} \leq f_{n,m} \leq F_{n,m} \quad \forall t_{n,m} \tag{2.4}$$

$$\theta_{n_0} = 0 \tag{2.5}$$

$$f_{n,m} = B_{n,m}(\theta_n - \theta_m) \quad \forall t_{n,m} \tag{2.6}$$

$$\sum_{d \in \Psi_n} p_d^D + \sum_{m \in \Omega_n} f_{n,m} - \sum_{g \in \Psi_n} p_g^G = 0 \quad \forall n \tag{2.7}$$

Note that unlike the original formulation the flows are represented as additional variables with their own bounds (Eq. 2.4) and explicitly linked to the voltage angles (Eq. 2.6) for ease of retrieval in the implementation.

In order to expand the problem to a reasonable size, day-ahead bid and offer data for October 7, 2020 were downloaded from the Midcontinent ISO (MISO) Market Data site [2]. The data format was mostly interpreted according to the MISO Energy Markets Business Practices Manual [3]. However, some simplifications were made:

- The day-ahead power market was simplified to economic participants and active power only, removing ancilliary services.
- Non-monotonic offer curves and negative price offers were removed from the data set.
- Binary variables were avoided, as directed on [1, p. 14], and therefore:
  - All slope bids were considered as stepwise bids
  - Unit commitment constraints and costs were ignored
- The network was simplified by grouping all generators and demands into regional zones
- Fixed (price taking) demands were represented with demand prices set to the price cap
- Offer curves were represented by independent generator variables for each segment of the curve (as curve prices increase monotonically, there is no need to bind variables in the same curve)

This resulted in a problem with 3617 generators, 309 demands (of which 268 were price-taking), 3 nodes, and 3 transmission lines; the size of the problem became 3932 variables (with their bounds) and 6 additional linear constraints. The problem was coded in Python using the MOSEK Fusion API [4], and both run without transmission constraints and with a 5GW transmission limit[1]; a summary of the results can be found in Table 1 (with prices included, as MOSEK supplies dual variables to constraints by default).

As anticipated, congestion of inter-region transmission causes prices to differ between nodes; in this particular constrained case, the marginal generating unit in the North region is zero-cost (such as wind or solar). Transmission constraints, unsurprisingly, also reduce the objective value (the total social welfare). As this was a relatively straightforward linear optimisation problem, and MOSEK deals with high numbers of variables more quickly than it does high numbers of constraints, the solver completed in less than 20ms in both cases. Since problem setup with the Fusion API took longer than the optimisation itself, this suggests that when repeatedly solving large numbers of cases, it will be more effective to set up the problem once, and only change parameters, than set up the problem once per iteration.

[1] In the accompanying IPython notebooks, `proj1-primal-tu.ipynb` is the problem without constraints and `proj1-primal-tc.ipynb` includes the transmission constraints; these have also been rendered as HTML for viewers without Juptyer Lab. Note that in both cases the MOSEK log lists 3933 variables; this is because by default MOSEK adds a variable, constrained to equal 1.0, that is used internally in the problem translation.

region	price ($/MW)	consumed (MW)	supplied (MW)
North	9.54	12854.2	31372.5
Central	9.54	28181.2	18817.1
South	9.54	16367.5	7213.3

(a) Unconstrained transmission (objective value: 1.13974e+08)

	price ($/MW)	consumed (MW)	supplied (MW)
North	0.00	12854.2	22854.2
Central	11.84	28178.1	23178.1
South	11.32	16367.5	11367.5

(b) Constrained transmission (objective value: 1.13911e+08)

Table 1: Market clearing summary

3 Dual problem

Consider the primal problem in Section 2. We can substitute out flows $f_{n,m}$ to put the problem in canonical form (and denote the Lagrange multipliers of the constraints) as follows:

$$\min_{p_g^G \geq 0, p_d^D \geq 0, \theta_n} -\sum_d U_d p_d^D + \sum_g C_g p_g^G \tag{3.1}$$

subject to:

$$p_d^D - \overline{P}_d^D \leq 0 \quad : \overline{\mu}_d^D \quad \forall d \tag{3.2}$$

$$-p_d^D \leq 0 \quad : \underline{\mu}_d^D \quad \forall d \tag{3.3}$$

$$p_g^G - \overline{P}_g^G \leq 0 \quad : \overline{\mu}_g^G \quad \forall g \tag{3.4}$$

$$-p_g^G \leq 0 \quad : \underline{\mu}_g^G \quad \forall g \tag{3.5}$$

$$B_{n,m}(\theta_n - \theta_m) - F_{n,m} \leq 0 \quad : \overline{\eta}_{n,m} \quad \forall t_{n,m} \tag{3.6}$$

$$-F_{n,m} - B_{n,m}(\theta_n - \theta_m) \leq 0 \quad : \underline{\eta}_{n,m} \quad \forall t_{n,m} \tag{3.7}$$

$$\sum_{d \in \Psi_n} p_d^D + \sum_{m \in \Omega_n} B_{n,m}(\theta_n - \theta_m) - \sum_{g \in \Psi_n} p_g^G = 0 \quad : \lambda_n \quad \forall n \tag{3.8}$$

$$\theta_{n_0} = 0 \quad : \gamma \tag{3.9}$$

The Lagrangian then becomes:

$$\begin{aligned}
&\mathcal{L}\left(p_g^G, p_d^D, \theta_n, \overline{\mu}_d^D, \underline{\mu}_d^D, \overline{\mu}_g^G, \underline{\mu}_g^G, \overline{\eta}_{n,m}, \underline{\eta}_{n,m}, \lambda_n, \gamma\right) = \\
&+ \sum_d \left(-U_d p_d^D + \overline{\mu}_d^D \left(p_d^D - \overline{P}_d^D\right) - \underline{\mu}_d^D p_d^D\right) + \sum_g \left(C_g p_g^G + \overline{\mu}_g^G \left(p_g^G - \overline{P}_g^G\right) - \underline{\mu}_g^G p_g^G\right) \\
&+ \sum_n \sum_{m \in \Omega_n} \left(\overline{\eta}_{n,m}\left(B_{n,m}(\theta_n - \theta_m) - F_{n,m}\right) + \underline{\eta}_{n,m}\left(-F_{n,m} - B_{n,m}(\theta_n - \theta_m)\right)\right) \\
&+ \sum_n \lambda_n \left(\sum_{d \in \Psi_n} p_d^D + \sum_{m \in \Omega_n} B_{n,m}(\theta_n - \theta_m) - \sum_{g \in \Psi_n} p_g^G\right) + \gamma \theta_{n_0}
\end{aligned} \tag{3.10}$$

And the dual problem is:

$$\max_{\overline{\mu}_d^D \geq 0, \underline{\mu}_d^D \geq 0, \overline{\mu}_g^G \geq 0, \underline{\mu}_g^G \geq 0, \overline{\eta}_{n,m} \geq 0, \underline{\eta}_{n,m} \geq 0, \lambda_n, \gamma} \left(\min_{p_g^G \geq 0, p_d^D \geq 0, \theta_n} \mathcal{L}(\dots) \right) \tag{3.11}$$

Taking the partial derivatives of $\mathcal{L}(\dots)$ with respect to the primal variables to get the Kuhn-Tucker conditions of the inner optimisation:

$$\frac{\partial \mathcal{L}(\dots)}{\partial p_d^D} = -U_d + \overline{\mu}_d^D - \underline{\mu}_d^D + \lambda_{n:d \in \Psi_n} = 0 \quad \forall d \tag{3.12}$$

$$\frac{\partial \mathcal{L}(\dots)}{\partial p_g^G} = C_g + \overline{\mu}_g^G - \underline{\mu}_g^G - \lambda_{n:g \in \Psi_n} = 0 \quad \forall g \tag{3.13}$$

$$\frac{\partial \mathcal{L}(\dots)}{\partial \theta_n} = \sum_{m \in \Omega_n} B_{n,m} \left(\overline{\eta}_{n,m} - \overline{\eta}_{m,n} - \underline{\eta}_{n,m} + \underline{\eta}_{m,n} + \lambda_n - \lambda_m \right) + \gamma = 0 \quad \text{when} \quad n = n_0 \tag{3.14}$$

$$\frac{\partial \mathcal{L}(\dots)}{\partial \theta_n} = \sum_{m \in \Omega_n} B_{n,m} \left(\overline{\eta}_{n,m} - \overline{\eta}_{m,n} - \underline{\eta}_{n,m} + \underline{\eta}_{m,n} + \lambda_n - \lambda_m \right) = 0 \quad \forall n \neq n_0 \tag{3.15}$$

The remaining terms with no primal variables in them can be extracted to the outer optimisation and the dual problem becomes:

$$\max_{\overline{\mu}_d^D \geq 0, \underline{\mu}_d^D \geq 0, \overline{\mu}_g^G \geq 0, \underline{\mu}_g^G \geq 0, \overline{\eta}_{n,m} \geq 0, \underline{\eta}_{n,m} \geq 0, \lambda_n, \gamma} \sum_d \overline{\mu}_d^D \overline{P}_d^D + \sum_g \overline{\mu}_g^G \overline{P}_g^G + \sum_n \sum_{m \in \Omega_n} F_{n,m} \left(\underline{\eta}_{n,m} + \overline{\eta}_{n,m} \right) \tag{3.16}$$

subject to the constraints from Eq. 3.12 through Eq. 3.15. We can simplify this one step further by noting that since $\underline{\mu}_d^D$ and $\underline{\mu}_g^G$ do not appear in the objective, and are always positive, we can eliminate them from the problem and turn Eq. 3.12 and Eq. 3.13 into inequalities:

$$-U_d + \overline{\mu}_d^D - \lambda_{n:d \in \Psi_n} \geq 0 \quad \forall d \tag{3.17}$$

$$C_g + \overline{\mu}_g^G - \lambda_{n:g \in \Psi_n} \geq 0 \quad \forall g \tag{3.18}$$

This is a problem with 3943 variables and 3617 + 309 + 3 = 3929 constraints (note that three variables from the formulation in Section 2 were excluded, so the dual problem has 3 fewer constraints than the primal problem has variables). The MOSEK solves this, with binding transmission constraints, in approximately 50ms; this is higher than the previous formulation's time, but considering the variance between runs, not significantly so. The results[2] are, as expected, identical to those in Section 2.

4 Karsh-Kuhn-Tucker conditions

The Karush-Kuhn–Tucker (KKT) conditions for optimality are the intersection of the first-order conditions (Eq. 3.12 through Eq. 3.15), the original equality constraints Eq. 3.8 and

[2] In the accompanying notebook `proj1-dual-tc.ipynb`

Eq. 3.9, and the complementarity conditions for the inequality constraints Eq. 3.2 to Eq. 3.7, which are as follows:

$$0 \leq -p_d^D + \overline{P}_d^D \perp \overline{\mu}_d^D \geq 0 \quad \forall d \tag{4.1}$$

$$0 \leq p_d^D \perp \underline{\mu}_d^D \geq 0 \quad \forall d \tag{4.2}$$

$$0 \leq -p_g^G + \overline{P}_g^G \perp \overline{\mu}_g^G \geq 0 \quad \forall g \tag{4.3}$$

$$0 \leq p_g^G \leq 0 \perp \underline{\mu}_g^G \geq 0 \quad \forall g \tag{4.4}$$

$$0 \leq -B_{n,m}(\theta_n - \theta_m) + F_{n,m} \perp \overline{\eta}_{n,m} \geq 0 \quad \forall t_{n,m} \tag{4.5}$$

$$0 \leq F_{n,m} + B_{n,m}(\theta_n - \theta_m) \perp \underline{\eta}_{n,m} \geq 0 \quad \forall t_{n,m} \tag{4.6}$$

For the KKT implementation in MOSEK[3] these bilinear conditions are implemented using the Big-M method. As an example, Eq. 4.2 is implemented as:

$$0 \leq p_d^D \leq Mz \qquad 0 \leq \underline{\mu}_d^D \leq M(1-z) \qquad z \in 0,1 \qquad M \gg 0 \tag{4.7}$$

However, this method does not give the same results; further investigation as to why is currently ongoing but has taken longer than planned.

.

5 References

[1] J. Kazempour, "Lecture 1: Market clearing as an optimization problem," DTU 31792 – Advanced Optimization and Game Theory for Energy Systems, Jan. 2021. [Online]. Available: https://cn.inside.dtu.dk/cnnet/filesharing/download/dcb15e39-cb18-4656-a13b-fec8ef3201bd

[2] "MISO Market Data." [Online]. Available: https://www.misoenergy.org/markets-and-operations/real-time--market-data/market-reports/#t=10&p=0&s=MarketReportPublished&sd=desc

[3] "MISO Energy and Operating Reserve Markets Business Practices Manual," MISO, Tech. Rep. BPM-002-r19, Oct. 2020.

[4] MOSEK ApS, "MOSEK Fusion API for Python manual. Version 9.2.35," Tech. Rep., 2021. [Online]. Available: https://docs.mosek.com/9.2/pythonfusion/index.html

[3] In the accompanying notebook `proj1-kkt-tc.ipynb`