Optimizing Diesel Generator Dispatch for a Microgrid with a Solar PV Array

Glen Woodworth

Alaska Center for Energy and Power
University of Alaska Fairbanks

February 5, 2025

Outline

Outline

General Set Up

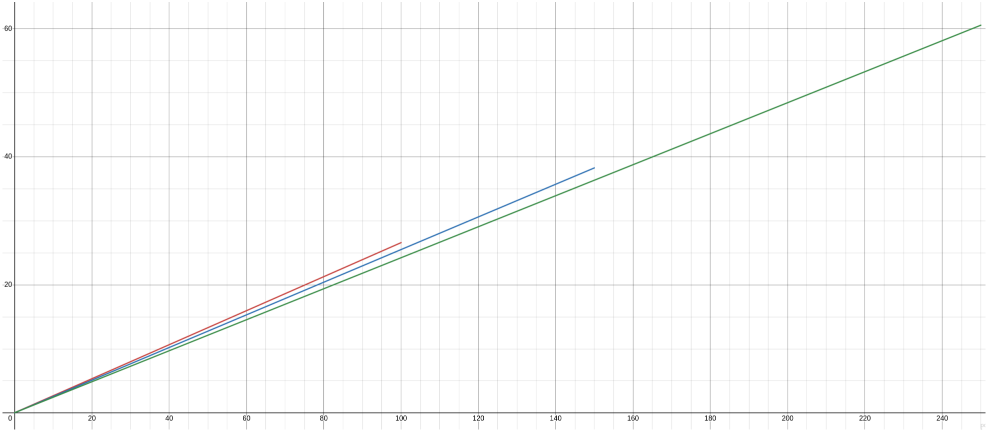
Minimize the fuel consumption for a simple microgrid with

- $K = 3$ generators,
- $M = (100, 150, 250)^T$ max kW for each generator, respectively,
- $L = 168$ timetsteps (1 week, hourly resolution)
- D, demand, and P, normalized solar output, are data from [3],
- $r = 250$ kW of solar PV

All data, code, and a detailed report (math heavy) is posted at `https://github.com/gwoodworth/microgrid-optimization`.

Plots of Fuel Curves

Fuel consumption (L/h) vs. generator setpoint (kW)



Red: Gen 1, 100 kW; Blue: Gen 2, 150 kW; Green: Gen 3, 250 kW.

Outline

Assumptions

The following assumptions simplify the task and ensure we can state the problem using linear functions only.

1. Assume perfect knowledge (no forecasting required).
2. Diesel generators have linear fuel curve.
3. No inverter.
4. Must meet load exactly.
5. Solar PV output never exceeds demand (forced using preprocessing).

Outline

Optimization

A general optimization problem consists of

- an objective function f, e.g.,$f(x) = x^2$,
- any number of equality constraints, e.g., $2x + 7 = 10$ and/or,
- any number of inequality constraints, e.g., $x \geq 2$ and $x \leq 7$.

Definition

Values of x that satisfy all of the constraints are called **feasible points** and the set of all feasible points is callled the **feasible set**, usually denoted by S.

General Optimization Problem Statement

In general, any optimization problem can be represented in the following form:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \geq 0, \; i \in \mathcal{I} \\ & g_j(x) = 0, \; j \in \mathcal{E}, \end{aligned}$$

where f and each g_i, g_j are real-valued functions while $\mathcal{I}$ and $\mathcal{E}$ are index sets for the inequality and equality constraints, respectively.

- Solving general optimization problems is hard.
- Approximating a general problem with a linear problems can be a good way to get started.
- We approximate the generator dispatch problem using a linear program

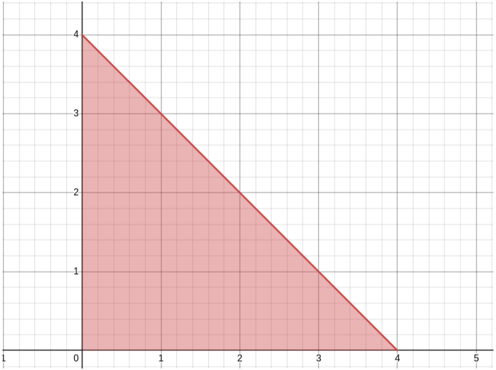
Linear Programming

Definition

A **linear program**, LP, is an optimization problem where the objective function is linear and the feasible set is defined by finitely many linear equality and linear inequality constraints.

Example LP

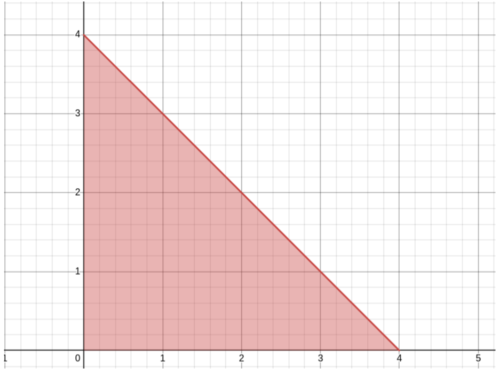
$$
\begin{aligned}
\text{minimize} \quad & f(x) = -x_1 - 2x_2 \\
\text{subject to} \quad & x_1 + x_2 \leq 4, \\
& x_1, x_2 \geq 0.
\end{aligned}
$$



An optimal solution to this LP is...

Example LP

$$
\begin{aligned}
\text{minimize} \quad & f(x) = -x_1 - 2x_2 \\
\text{subject to} \quad & x_1 + x_2 \leq 4, \\
& x_1, x_2 \geq 0.
\end{aligned}
$$



An optimal solution to this LP is $x = (0, 4)$.

Representing Generators

Goal: Write our dispatch problem as a LP. Let

- $l = 1, 2, \ldots, L$ represent the timesteps,
- $k = 1, 2, \ldots, K$ represent the diesel generators,
- x_i be the load setpoint for generator k at timestep l where $i = l + L(k-1)$, and
- M_k is the maximum capacity in kW of generator k.

Define fuel curve $g_k : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ by

$$g_k(x) = 0.4234(M_k)^{-0.1012}x + 0.0940 \cdot (M_k)^{-0.2735}$$

where $g_k(x)$ is the fuel consumption of generator k in L/kWh at load setpoint x kW [3, 56].

Example: Setpoints as a Vector

Suppose we have two generators and three total timesteps so that $x \in \mathbb{R}^{2 \cdot 3}$. We have

l	1	2	3
Generator 1	100	200	300
Generator 2	350	250	150
Demand	450	450	450

and

$$x = \begin{pmatrix} 100 \\ 200 \\ 300 \\ 350 \\ 250 \\ 150 \end{pmatrix}.$$

Recall the load setpoint

$$x_i, \; i = l + L(k-1)$$

for generator k at timestep l, with L total timesteps.

Outline

Objective Function

Goal:

$$\text{minimize} \quad z = \sum_{l=1}^{L} \sum_{k=1}^{K} g_k(x_i),$$

i.e., minimize total fuel consumption, where

- $l = 1, 2, \ldots, L$ represent the timesteps,
- $k = 1, 2, \ldots, K$ represent the diesel generators,
- x_i be the load setpoint for generator k at timestep l where $i = l + L(k-1)$,
- M_k is the maximum capacity in kW of generator k, and
- $g_k(x)$ is the fuel consumption of generator k in L/kWh at load setpoint x kW.

Demand Constraints

Let

- $D \in \mathbb{R}^L$ be the electrical demand in kW,
- $P \in \mathbb{R}^L$ be normalized output of the solar PV array, where $0 \leq P_l \leq 1$ is the ratio of the output of the array at timestep l to its maximum capacity,
- r be the rated capacity of the solar PV array in kW.

We want the sum of the output of the diesel generators and the solar PV array to meet the demand exactly at each timestep:

$$P_l \cdot r + \sum_{k=1}^{K} x_i = D_l.$$

Preprocessing for Demand Constraints

We require:

$$D - P \cdot r \geq 0 \text{ and}$$
$$\sum_{k=1}^{K} M_k \geq D_l - P_l \cdot r \text{ for all } l \in \{1, 2, \ldots, L\}.$$

The second condition was met by our data, but first condition required the preprocessing of $D - P \cdot r$. We used

$$D_l' = \max(D_l - P_l \cdot r, 0),$$

so that our new demand constraints are

$$\sum_{k=1}^{K} x_i = D_l', \; l \in \{1, 2 \ldots, L\}.$$

Generator Constraints

- Each generator k has a max capacity M_k, for a total of $K \cdot L$ upper bound inequality constraints, each of the following form

$$x_i \leq M_k,$$

where the indexing is taken care of by $i = l + L(k-1)$.

- Since we have L equality constraints from the demand, and $K \cdot L$ upperbound inequality constraints from the generators, we have a total of

$$m = L + K \cdot L$$

general equality and inequality constraints.

Dispatch as a LP

We can now state the problem as a LP:

$$
\begin{aligned}
\text{minimize} \quad & z = \sum_{l=1}^{L} \sum_{k=1}^{K} g_k(x_i) \\
\text{subject to} \quad & \sum_{k=1}^{K} x_i = D_l',\ l \in \{1, 2, \ldots L\} \\
& x_i \leq M_k.,\ l \in \{1, \ldots, L\}, k \in \{1, \ldots K\} \\
& x \geq 0.
\end{aligned}
$$

Standard Form

- Standard Form (SF) is a representation of a linear program that can easily be written down using matrices and vectors only:

$$\begin{aligned} \text{minimize} \quad & z = c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

  where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $b \geq 0$, and A is an $m \times n$ matrix.
- SF is useful for putting a LP on a computer.
- SF is required by algorithms for solving LP, such as Simplex method.
- In general, SF usually increases the dimension of the problem.
- SF can be obtained for every LP and in an algorithmic way (it can be coded)

Convert LP to SF

To convert our generator dispatch LP into standard form, we have three tasks.[1]

1. Add $K \cdot L$ slack variables to turn the general inequalities into equalities,
2. rewrite objective z as $z - p = c^T x$ where p is some scalar, and
3. write the resulting constraints in matrix vector form.

[1] These steps were implemented in matlab functions available at the github repository for this project in the folder 'functions': https://github.com/gwoodworth/microgrid-optimization.

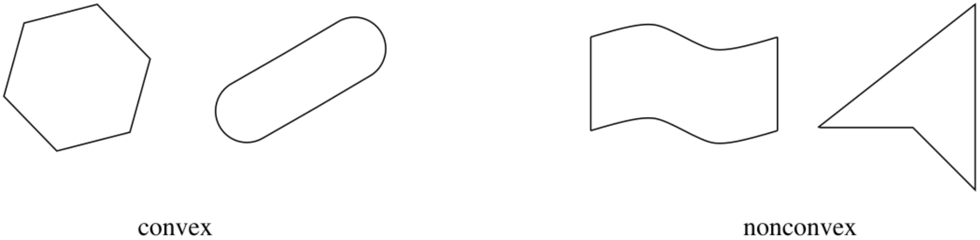
Outline

Convex Sets

Definition

A set $S \subset \mathbb{R}$ is *convex* if for all $x, y \in S$

$$\alpha x + \alpha y \in S \quad \text{for all } 0 \leq \alpha \leq 1 \text{ [1].}$$



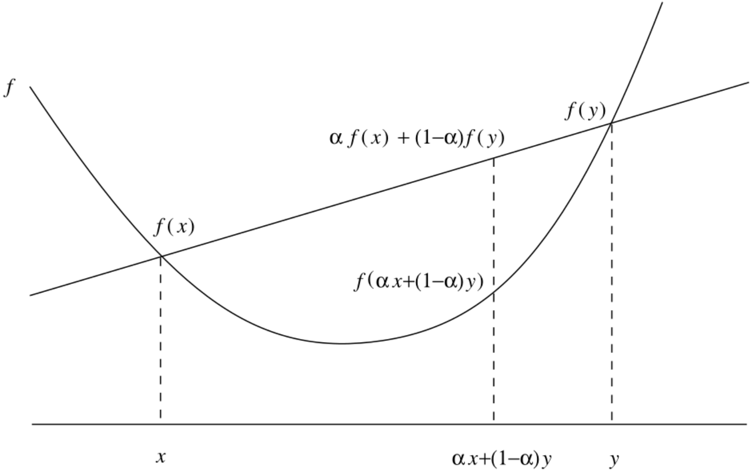
Convex Functions

Definition

A function $f : S \to \mathbb{R}$ where S is a convex set is a *convex function* if for all $0 \leq \alpha \leq 1$ and any $x, y \in S$,

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) \quad [1]$$

Convex Functions Picture



Basic Feasible Solutions

Given a LP in SF,

$$\begin{aligned} \text{minimize} \quad & z = c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

a Basic Feasible Solution (BFS) x is a

- Solution to the system of linear equations $Ax = b$,
- Feasible point: $x \geq 0$,
- Basic point: the columns of the constraint matrix A corresponding to the nonzero components of x are linearly independent [1].

Why do we care?

- Convex optimization problems are optimization problems where the feasible set and the objectvive function are both convex.
- All LP are convex optimization problems.
- Corners of the feasible set of a LP in SF are BFS.
- If an optimal solution to a convex optimization problem exists, at least one of them will be a BFS [1].

Once a LP is in SF, we need only check the corners of the feasible set for an optimal solution.

Simplex – It's Not Brute Force (usually)

Simplex method:

- is an algortihm for solving LP.
- requires a Basic Feasible Solution to get started.
- checks the corners of the feasible set for an optimal solution in a way that usually avoids checking all of them.
- is faster than brute force in most real world cases.[2]

[2] In contrived examples, Simplex will check every BFS and be no better than brute force [2].

Getting Started

Finding an initial BFS is often difficult. Phase One is an algorithm that can find a BFS for a LP in SF. Phase One steps:

1. add variables to the constraints of the original LP,
2. change the objective function to be the sum of those additional variables,
3. use Simplex method to find a BFS where all of the additional variables are 0.

The result is either that there are no BFS's for the problem, which means there is not an optimal solution for the LP, or, a BFS for the original LP that can be used to start Simplex.

Steps of Simplex

Given a LP in SF,

$$\begin{aligned} \text{minimize} \quad & z = c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

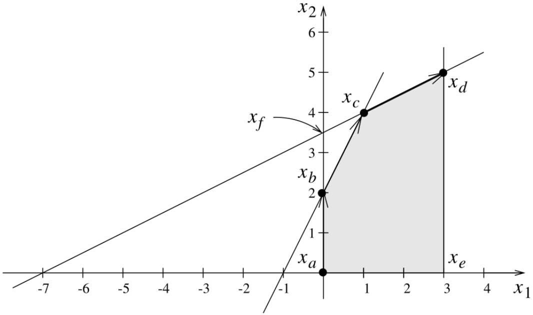
Simplex method operates as follows:

1. Start at a BFS, $\hat{x}$ (use Phase One).
2. Test for optimality: Check if z can be made smaller by increasing any of the nonbasic variables (i.e., the entries of $\hat{x}$ equal to zero).
3. If $\hat{x}$ is optimal, terminate.
4. Otherwise, set one of the basic variables to 0 and increase one of the nonbasic variables, as identified by the optimality test.
5. Iterate steps 2-5.

Simplex Example

The following example is from [1].

$$\begin{aligned} \text{minimize} \quad & z = -x_1 - 2x_2 \\ \text{subject to} \quad & -2x_1 + 2x_2 \leq 2, \\ & -x_1 + 2x_2 \leq 7, \\ & x_1 \leq 3 \\ & x_1, x_2 \geq 0. \end{aligned}$$

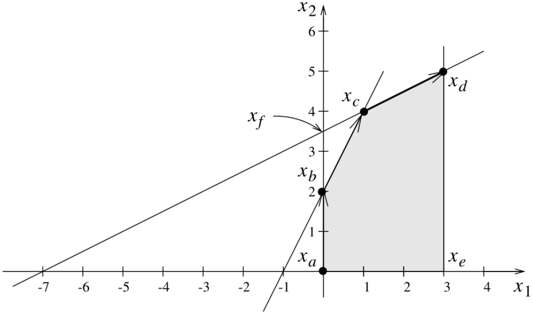


An optimal solution to this LP is...

Simplex Example

The following example is from [1].

$$
\begin{aligned}
\text{minimize} \quad & z = -x_1 - 2x_2 \\
\text{subject to} \quad & -2x_1 + 2x_2 \le 2, \\
& -x_1 + 2x_2 \le 7, \\
& x_1 \le 3 \\
& x_1, x_2 \ge 0.
\end{aligned}
$$



An optimal solution to this LP is $x = (3, 5)$.

Outline

Steps for Solving Dispatch LP

1. Write problem in SF.
2. Find a BFS using a function called Phase One.
3. Run Simplex
4. Convert result of Simplex back to original variables.

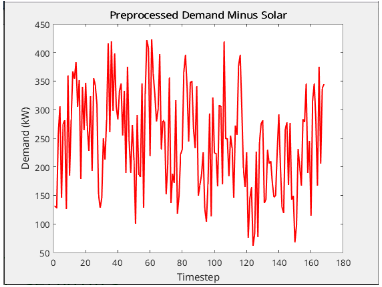
For implementation in MATLAB, see https://github.com/gwoodworth/microgrid-optimization.

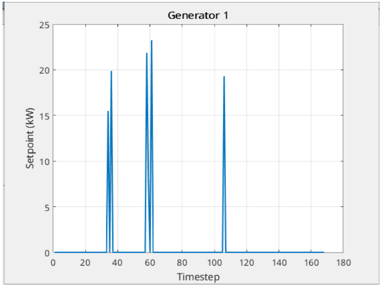
Parameters

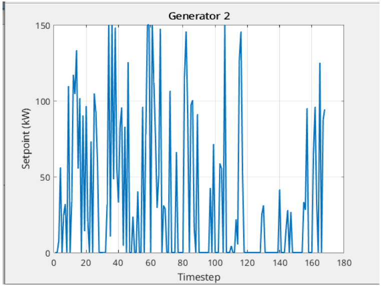
I used the following parameters:

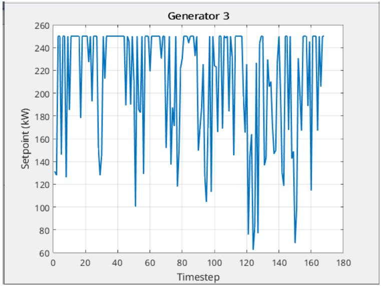
- $K = 3$ generators,
- $M = (100, 150, 250)^T$ max kW for each generator, respectively,
- $L = 168$ timetsteps (1 week, hourly resolution)
- D, demand, and P, normalized solar output, are data from [3],
- $r = 250$ kW of solar PV
- $m = L + K \cdot L = 168 + 3(168) = 672$
- $n = 2 \cdot K \cdot L = 2(3)(168) = 1,008$
- In SF, contstraint matrix has shape $m \times n = 672 \times 1,008$

Plots of results



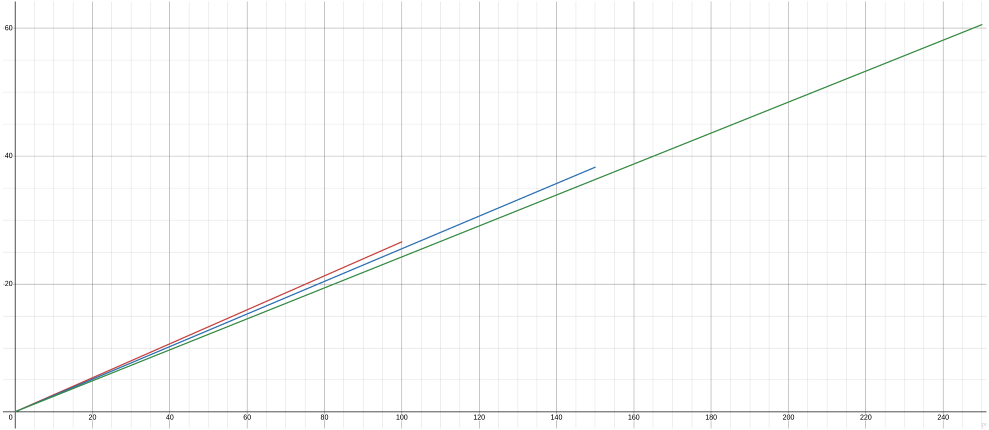






Plots of Fuel Curves

Fuel consumption (L/h) vs. generator setpoint (kW)



Red: Gen 1, 100 kW; Blue: Gen 2, 150 kW; Green: Gen 3, 250 kW.

Analysis

- Results indicate alignement with an obvious logic for solving the problem: use the most efficient generator first.
- Phase One always finds an optimal BFS
- For one week of data, Phase took 900 itertions to find a BFS but then Simplex takes only a single iteration
- Approach can be modified to be more realistic, e.g., add a penalty to the objective function for generators operating at lower setpoints than they should (nonlinear).
- A nonlinear problem would require a different algorithm from Simplex, e.g., one based on Newton's method (see POPDIP `https://github.com/bueler/popdip/tree/main`)
- Mixed Integer Linear Programming (where some variables are forced to be integers) could also work but the method would still be more complicated than Simplex.

References

Igor Griva, Stephen G Nash, and Ariela Sofer.
Linear and Nonlinear Optimization 2nd Edition.
SIAM, 2008.

Victor Klee and George J Minty.
How good is the simplex algorithm.
Inequalities, 3(3):159–175, 1972.

Anthony Truelove.
Pgmcpp.
https://github.com/gears1763-2/PGMcpp, 2024.

Questions?

- How to add a battery? Thoughts?