Specialisation Project (VT1)
HS2024

Platform for Investment Analysis

Optimization Framework for Energy Asset Management using Linear Programming in Python

Submitted by
Rui Vieira
Institute of Product Development and Production Technologies (IPP)

Supervisor
Dr. Andrea Giovanni Beccuti
IEFE Model Based Process Optimisation

Study Program
Business Engineering, MSc in Engineering

Zurich University of Applied Sciences
School of Engineering

January 18, 2025

Imprint

Project: Specialisation Project (VT1)
Title: Platform for Investment Analysis
Author: Rui Vieira
Date: January 18, 2025
Keywords: linear programming, quantitative modeling, python, strategic planning, risk assessment, portfolio optimization, asset valuation, power-flow, platform

Study program:
Business Engineering, MSc in Engineering
ZHAW School of Engineering

Supervisor:
Dr. Andrea Giovanni Beccuti
IEFE Model Based Process Optimisation
Email: giovanni.beccuti@zhaw.ch

Abstract

This project presents a platform for investment analysis in energy systems, focusing on the optimization of asset management through linear programming techniques. The framework integrates DC Optimal Power Flow (DC-OPF) for network analysis with economic evaluation methods to support investment decisions in energy infrastructure.

The platform, implemented in Python, combines power system modeling with financial analysis tools to evaluate different investment scenarios. It features automated scenario generation, optimization of operational costs, and AI-assisted analysis of results. The methodology incorporates both technical constraints from power system operations and financial metrics such as Net Present Value (NPV) to provide comprehensive decision support.

Key features include modular architecture for extensibility, integration with industry-standard optimization solvers, and a flexible scenario analysis framework. The platform demonstrates practical applicability through case studies in energy infrastructure investment, showing how it can be used to evaluate complex investment decisions while considering both technical and economic factors.

Keywords: linear programming, quantitative modeling, python, strategic planning, risk assessment, portfolio optimization, asset valuation, power-flow, platform

Contents

1 Introduction

In the rapidly evolving energy landscape, the need for sophisticated tools to evaluate investment decisions in electrical infrastructure has become increasingly critical. Power system operators and investors face complex decisions involving multiple variables, from generation capacity placement to storage system optimization, all while ensuring grid stability and economic viability.

1.1 Project Context

1.2 Objectives

- Develop a Python-based platform for energy investment analysis
- Implement linear programming optimization for asset management
- Create scenario analysis capabilities
- Provide AI-powered insights for decision support

1.1 Context and Motivation The transition towards renewable energy sources has introduced new challenges in power system planning. Unlike traditional power plants, renewable sources such as solar and wind exhibit variable generation patterns, requiring more sophisticated analysis tools. Additionally, the geographic distribution of these resources often necessitates careful consideration of grid topology and transmission constraints. These factors, combined with the significant capital investments required, make it crucial to have robust analytical tools for investment decision-making. 1.2 Problem Statement The core challenge addressed in this project is the development of a comprehensive platform that can: Optimize power flow across a given grid topology Evaluate multiple investment scenarios simultaneously Consider seasonal variations in both generation and demand Account for various types of electrical assets (conventional generation, renewables, storage) Provide economic analysis through NPV and annuity calculations Deliver automated, AI-enhanced interpretation of results 1.3 Project Objectives This project aims to create an integrated platform that combines power system analysis with investment evaluation. The specific objectives include: Development of a flexible scenario definition system that allows users to specify different investment cases Implementation of a DC Optimal Power Flow (DCOPF) solver to determine optimal dispatch of generation assets Integration of seasonal analysis capabilities to account for varying generation and demand patterns 4. Creation of an automated reporting system enhanced by AI-powered analysis Development of comparative analysis tools to evaluate different investment scenarios 1.4 Solution Overview The developed platform integrates several key components: A Python-based DCOPF solver using the PuLP optimization library A scenario management system using standardized CSV formats Seasonal analysis capabilities for comprehensive yearly evaluation Automated report generation with AI-enhanced insights Economic analysis tools for investment evaluation This solution uniquely combines power system optimization with investment analysis, providing a comprehensive tool for decision-makers in the energy sector. — Would you like me to: Expand any particular section? Add more technical details about the DCOPF implementation? Include a specific diagram or visualization? Adjust the tone or level of technical detail?

2 Theoretical Background

Power systems fundamentals DC Optimal Power Flow explanation Investment analysis concepts (NPV, annuities) Mathematical formulation of the optimization problem

2.1 Linear Programming in Energy Systems

Overview of optimization techniques in energy asset management.

2.2 DC Optimal Power Flow

The DC Optimal Power Flow (DC-OPF) is a simplified version of the AC power flow problem, commonly used in power systems analysis and optimization [**?**]. This linearized model, thoroughly described in [**?**], makes the following key assumptions [**?**]:

- Voltage magnitudes are assumed to be 1.0 per unit (p.u.)
- Line resistances are negligible compared to reactances ($R \ll X$)
- Voltage angle differences between connected buses are small
- Reactive power flows are ignored

2.2.1 Mathematical Formulation

The DCOPF problem is formulated as a linear programming optimization:

$$\min_{\mathbf{P_g},\boldsymbol{\theta}} \sum_{i \in \mathcal{G}} c_i(P_{g,i}) \tag{1}$$

Subject to the following constraints:

Power Balance Constraints:

$$P_{g,i} - P_{d,i} = \sum_{j \in \mathcal{N}_i} B_{ij}(\theta_i - \theta_j) \quad \forall i \in \mathcal{N} \tag{2}$$

Generation Limits:

$$P_{g,i}^{\min} \leq P_{g,i} \leq P_{g,i}^{\max} \quad \forall i \in \mathcal{G} \tag{3}$$

Line Capacity Limits:

$$-P_{ij}^{\max} \leq B_{ij}(\theta_i - \theta_j) \leq P_{ij}^{\max} \quad \forall (i,j) \in \mathcal{L} \tag{4}$$

Where:

- $P_{g,i}$: Power generation at bus i
- $P_{d,i}$: Power demand at bus i
- θ_i: Voltage angle at bus i
- x_{ij}: Line reactance between buses i and j
- P_{ij}: Power flow on line between buses i and j
- c_i: Generation cost coefficient at bus i

2.2.2 Additional Constraints for Investment Analysis

For our investment optimization, we add the following constraints:

Investment Budget Constraint:

$$\sum_{i \in G} I_i x_i \leq B_{max} \tag{5}$$

Capacity Factor Constraints:

$$\frac{1}{T} \sum_{t=1}^{T} \frac{P_{g,i,t}}{P_{g,i}^{max}} \geq CF_{min,i} \quad \forall i \in G \tag{6}$$

Where:

- I_i: Investment cost for generator i
- x_i: Binary decision variable for investment in generator i
- B_{max}: Maximum investment budget
- $CF_{min,i}$: Minimum required capacity factor for generator i
- T: Number of time periods

2.2.3 Implementation Considerations

- The problem is implemented using Python with Pyomo optimization framework
- Seasonal variations are handled through multiple time periods
- Generator availability is modeled using capacity factors
- Investment decisions are binary variables

3 Methodology and Implementation

3.1 System Architecture

Description of the linear programming implementation.

3.2 Core Components

How different scenarios are generated and compared.

3.3 Technical Implementation

Implementation of AI-powered analysis features.

4 Results and Validation

4.1 Test Cases

Description of validation scenarios.

4.2 Performance Analysis

Computational efficiency and scalability.

4.3 Case Studies

Real-world applications and insights.

5 Discussion

5.1 Platform Capabilities

Current functionality and limitations.

5.2 Future Improvements

Potential enhancements and extensions.

6 Conclusion

A Technical Documentation

A.1 Installation Guide

A.2 User Manual

A.3 API Reference

B Mathematical Formulations

B.1 Optimization Model

B.2 Economic Calculations