Two-stage approach for the predispatch problem with uncertain demand using splitting variables in interior-point methods

Jordi Castro [a,*] Catalina J. Villalba [b] Aurelio R.L. Oliveira [b]

[a] Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Catalonia
[b] Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil
[*] Corresponding author.
E-mail addresses: jordi.castro@upc.edu (J. Castro), catajv@unicamp.br (C. J. Villalba), aurelio@unicamp.br (A.R.L. Oliveira).

Abstract

The stochastic predispatch optimal power flow problem aims to minimize generation costs and transmission losses subject to network constraints under demand uncertainty. We formulate it as a two-stage stochastic quadratic optimization problem with fixed recourse, in which hydroelectric generation constitutes the here-and-now decision, while thermal generation and transmission flows are recourse decisions. Using a splitting-variable reformulation, we cast the extensive form into a block-angular structure and solve the resulting problem with a specialized interior-point method (implemented in the academic solver BlockIP), which combines direct and iterative methods for computing the Newton direction. On a benchmark of seven networks from 30 to 9 241 buses with up to 100 demand scenarios, the method returns an optimal solution on every instance, including a large-scale case with millions of variables on which the commercial solvers CPLEX and Gurobi and the open-source solver HiPO-HiGHS all fail to return a solution (CPLEX and Gurobi exhaust the available memory, while HiPO-HiGHS terminates with a solver error). We further quantify the economic value of the stochastic solution through the VSS and EVPI metrics, characterizing the regimes in which the stochastic model yields the largest benefit over its deterministic counterpart.

Keywords. Interior-Point Methods, Stochastic Optimization, Optimal Power Flow, Splitting Variables.

Mathematics Subject Classification. 90C51, 90C20, 90C90

1 Introduction

Optimal Power Flow (OPF) is a mathematical optimization problem in energy systems management. Its primary objective is to optimize a specific operating criterion, such as generation cost or power losses, subject to the physical laws of the electrical network and operational constraints [18]. From a mathematical optimization perspective, OPF is important for various critical grid applications, including economic dispatch, congestion management, reactive power planning, and security analysis [14, 13]. While various network flow models exist, the Direct Current Optimal Power Flow (DC-OPF) linear approximation remains highly effective for active power flow applications, providing a reliable balance between computational tractability and physical accuracy [5, 20].

While traditional OPF provides a static state of the system at a single point in time, real-world power grid operation is dynamic. This requires the extension of OPF to the *predispatch* problem, a multi-period optimization framework that schedules generation and transmission over a short-term horizon, typically ranging from 24 hours to a week [30]. The predispatch formulation connects a sequence of OPF problems through inter-temporal constraints, such as the operational ramp rates of thermal power plants and the storage dynamics of hydroelectric reservoirs. In practical applications, predispatch models are important for hydro-thermal coordination, ensuring that water resources are conserved while expensive thermal generation is minimized, as well as for scheduling transmission line maintenance and managing daily demand peaks [4].

Decisions in the energy sector, however, are influenced by uncertain parameters. In short-term operational planning, uncertainty manifests dynamically through load demand fluctuations, unexpected equipment outages, and the integration of renewable energy sources. To rigorously handle operational uncertainties like energy demand, stochastic optimization provides a robust probabilistic framework [1].

In this context, we propose a two-stage stochastic quadratic optimization model with fixed recourse to solve the predispatch OPF problem under uncertain demand. The first stage represents the "here-and-now" decisions, specifically, the generation targets for hydroelectric plants, while the second stage models the "recourse" actions of thermal plants generation and transmission flows once a specific demand scenario is realized. By analyzing historical consumption data, we generate discrete scenarios to approximate the probability distribution function of the load demand.

The primary computational challenge in multistage stochastic optimization is the large scale of the resulting formulation, which can lead to excessive memory requirements for some solvers. To address this issue, our methodological strategy employs variable splitting to decouple the scenarios. This technique reformulates the problem so that the different demand scenarios and stages give rise to a primal block-angular constraint matrix [10, 17]. We then exploit this structure using a specialized interior-point method (IPM) that combines direct and iterative techniques with suitable preconditioners [9, 7, 8]. Iterative methods provide an effective alternative to direct factorization for the solution of the Newton systems arising in large-scale IPMs and have also been considered in other interior-point approaches [11].

The main contributions of this paper are twofold. First, we formulate a two-stage stochastic DC-OPF predispatch model with a splitting-variable reformulation that yields a primal block-angular structure, enabling specialized interior-point methods which combine direct and iterative methods for the soltution of the Newton direction. Second, we demonstrate that this structure enables the efficient solution of large-scale stochastic predispatch problems, solving benchmark instances with up to 100 demand scenarios and millions of variables beyond the practical limits of general-purpose commercial solvers on commodity hardware. Finally, we assess the economic value of stochastic optimization through the VSS and EVPI metrics.

This paper is organized as follows. Section 2 details the multi-period optimal power flow model, and its stochastic formulation for uncertain demand, together with the splitting-variable reformulation. Section 3 outlines the specialized interior-point method for exploiting the resulting block-angular structure. Section 4 covers definitions related to the stochastic solution metrics. Section 5 presents the computational experiments analyzing energy systems, including large-scale networks. Finally, Section 6 presents our conclusions.

2 Optimal power flow and predispatch model

Power flow problems involve the generation and transmission of electrical energy to meet demand over a specified time period. When these strategies aim to optimize a specific economic or operational criterion, they are formulated as OPF problems [2, 4, 20].

While a standard OPF evaluates the grid at a single point in time, the operations of modern energy systems require short-term planning over a multi-hour horizon. This extension is known as the predispatch problem. It connects a sequence of OPF problems through inter-temporal constraints. In systems with a heavy reliance on water resources, this short-term planning is important for hydro-thermal coordination: meeting fluctuating demand while minimizing thermal generation costs and satisfying daily water utilization targets.

2.1 Static single-period model

The static OPF problem is an optimization model that aims to minimize an objective function, such as generation cost or transmission losses, subject to constraints reflecting the physical laws of the electrical network. In this paper, we consider a linear network flow approximation known as the DC-OPF, which has shown satisfactory results in large-scale transmission applications [4, 5, 23]. The mathematical formulation of the DC-OPF is given by:

$$\begin{aligned} \min \quad & \frac{\beta}{2}\left(p^\top Q p + c^\top p\right) + \frac{\alpha}{2} f^\top R f \\ \text{s.t.} \quad & Af = Hp - d \\ & Tf = 0 \\ & p_{\min} \le p \le p_{\max} \\ & f_{\min} \le f \le f_{\max}. \end{aligned} \tag{1}$$

The matrix $A \in \mathbb{R}^{m \times n}$ in (1) is the network incidence matrix, where m is the number of nodes (buses) and n is the number of edges (transmission lines). The vector $d \in \mathbb{R}^m$ represents the power demand at each node. The variables $f \in \mathbb{R}^n$ and $p \in \mathbb{R}^g$ represent the active power flow in the lines and the active power generation, respectively, with g being the number of generators.

The objective function has two terms weighted by parameters α and β. The first term, $\frac{\alpha}{2} f^\top R f$, penalizes active power losses in the transmission system, where the diagonal matrix R contains the resistance of each transmission line. The second term, $\frac{\beta}{2}(p^\top Q p + c^\top p)$, represents the generation cost, where Q is a diagonal matrix with strictly positive diagonal entries. The quadratic component models the incremental cost curve of thermal generators and the hydro generation losses associated with variations in tailrace elevation, penstock head losses, and turbine-generator efficiency [27].

The matrix $H \in \mathbb{R}^{m \times g}$ maps the generators to their respective buses, so, the first constraint ($Af = Hp - d$) ensures nodal active power balance (Kirchhoff's Current Law). The matrix $T \in \mathbb{R}^{(n-m+1) \times n}$ is the fundamental loop matrix representing the reactances of the transmission network, which encodes the reactance relationships along the independent loops of the transmission network. That is, given a spanning tree of the network graph, each non-tree edge defines a unique fundamental loop. The rows of T correspond to these loops, and for each loop, the entry associated with edge j is the signed reactance if edge j belongs to the loop, and zero otherwise. The constraint $Tf = 0$ ensures that the voltage drops around each independent loop sum to zero, which is equivalent to Kirchhoff's voltage law. An efficient heuristic for constructing a sparse loop matrix by choosing a spanning tree with small depth is described in [20].

Finally, $p_{\min}, p_{\max} \in \mathbb{R}^g$ and $f_{\min}, f_{\max} \in \mathbb{R}^n$ are the physical operational limits of generation and line flows.

Thus, the formulation in (1) represents the transmission system through a network flow model in which both Kirchhoff's current law ($Af = Hp - d$) and Kirchhoff's voltage law ($Tf = 0$) are stated as independent constraints [3]. This representation has the advantage that power flows are explicitly represented as optimization variables, allowing direct consideration of transmission limits as constraints and transmission losses as a performance criterion [20, 21].

2.2 Deterministic multi-period predispatch model

To transition from a static to a short-term operational planning horizon, we extend the OPF model across t hours. At this stage, we assume a planning of the hourly demand, denoted as d_i for hour $i \in \{1, \dots, t\}$.

In hydro-thermal coordination, it is necessary to differentiate the generation sources due to their distinct operational roles and costs. We split the generation vector p into two components: the generation from the g_1 hydroelectric plants ($p^1 \in \mathbb{R}^{g_1}$) and the generation from the g_2 thermal power plants ($p^2 \in \mathbb{R}^{g_2}$). Equivalent, the incidence matrix H is partitioned into H_1 and H_2, and the cost matrices are partitioned into (Q_1, c_1) and (Q_2, c_2).

A key feature of hydro-dominant systems is that the total water volume to be turbined over the planning horizon is determined by medium- and long-term models that coordinate reservoir management across the national system. In Brazil, this target is established by the *Operador Nacional do Sistema Elétrico* (ONS) and must be respected by the short-term predispatch [26, 19]. Mathematically, this inter-temporal coupling is captured by the constraint $\sum_{i=1}^{t} p_i^1 = p_h$, which links the hydroelectric generation decisions across all time periods. This constraint distinguishes the predispatch problem from a collection of independent OPF problems, as it requires coordinating the hourly allocation of a finite water resource to follow the load curve while minimizing the overall system cost. The deterministic predispatch problem is formulated as:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} \left\{ \frac{\beta_1}{2} \left((p_i^1)^\top Q_1 p_i^1 + c_1^\top p_i^1 \right) + \frac{\beta_2}{2} \left((p_i^2)^\top Q_2 p_i^2 + c_2^\top p_i^2 \right) + \frac{\alpha}{2} f_i^\top R f_i \right\} \\
\text{s.t.} \quad & \sum_{i=1}^{t} p_i^1 = p_h \\
& \left.\begin{array}{l}
H_1 p_i^1 + H_2 p_i^2 - A f_i = d_i \\
T f_i = 0 \\
p_{\min}^1 \le p_i^1 \le p_{\max}^1 \\
p_{\min}^2 \le p_i^2 \le p_{\max}^2 \\
f_{\min} \le f_i \le f_{\max}
\end{array}\right\} i = 1, \dots, t.
\end{aligned}
\tag{2}
$$

2.3 Multi-period predispatch stochastic model

Incorporating uncertainty into the predispatch problem is essential for robust decision-making. In the energy sector, the demand d is highly variable. If operators optimize based only on a deterministic forecast as in (2), the system may lack the necessary flexibility to respond to real-time fluctuations, leading to emergency dispatch of expensive peak generators.

To explicitly account for this uncertainty, it is necessary to distinguish between decisions taken before and after the realization of demand. This structure is captured by two-stage stochastic optimization models, defined as follows.

Definition 1 (Two-stage stochastic optimization problem with fixed recourse). A two-stage stochastic optimization problem with fixed recourse is an optimization problem of the form:

$$
\min_{x \in X} \left\{ c^\top x + \mathbb{E}_\xi [Q(x, \xi)] \right\},
$$

where x is the first-stage decision vector chosen before the random variable ξ is realized, X is the feasible set for first-stage decisions, $c^\top x$ is the deterministic first-stage cost, and $Q(x, \xi) = \min_y \{ q^\top y : Wy = h(\xi) - T(\xi)x,\ y \ge 0 \}$ is the optimal value of the second-stage (recourse) problem. The recourse is called *fixed* when the recourse matrix W does not depend on ξ [1].

To handle demand uncertainty, we propose a two-stage stochastic quadratic optimization model with fixed recourse, defining the following:

- **First stage:** Decisions that must be made before the uncertainty is realized. In our hydro-thermal context, the generation from hydroelectric plants, denoted as $p^1 \in \mathbb{R}^{g_1}$, acts as the primary base load and must be scheduled in advance to meet the daily target (p_h).

- **Second stage:** Decisions made after the uncertain demand is revealed. The thermal power plants, denoted as $p^2 \in \mathbb{R}^{g_2}$, and the resulting power flows f act as the recourse variables to balance the grid dynamically and satisfy the demand.
- The recourse is fixed because the network topology (matrices A, T, H_1, H_2) and the physical limits of the system remain unchanged across all scenarios.

Assuming the demand $d(\xi)$ is a discrete random variable with N possible scenarios, with probabilities $a_1, \ldots, a_N$, we obtain the extensive form of the stochastic problem.

Let p_i^1 be the first-stage hydro generation at hour $i \in \{1, \ldots, t\}$. Let p_{ij}^2 and f_{ij} be the second-stage thermal generation and power flow, respectively, at hour i under scenario $j \in \{1, \ldots, N\}$. Given the discrete uncertain demand d_{ij}, the stochastic predispatch problem is formulated as:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} \left\{ \frac{\beta_1}{2} \left((p_i^1)^\top Q_1 p_i^1 + c_1^\top p_i^1 \right) + \sum_{j=1}^{N} a_j \left[\frac{\beta_2}{2} \left((p_{ij}^2)^\top Q_2 p_{ij}^2 + c_2^\top p_{ij}^2 \right) + \frac{\alpha}{2} f_{ij}^\top R f_{ij} \right] \right\} \\
\text{s.t.} \quad & \sum_{i=1}^{t} p_i^1 = p_h \\
& \left. \begin{array}{l} H_1 p_i^1 + H_2 p_{ij}^2 - A f_{ij} = d_{ij} \\ T f_{ij} = 0 \\ p_{\min}^1 \leq p_i^1 \leq p_{\max}^1 \\ p_{\min}^2 \leq p_{ij}^2 \leq p_{\max}^2 \\ f_{\min} \leq f_{ij} \leq f_{\max} \end{array} \right\} i = 1, \ldots, t; \quad j = 1, \ldots, N.
\end{aligned}
\tag{3}
$$

Here, the total objective minimizes the deterministic cost of the scheduled hydro generation and the expected cost of the recourse thermal generation and transmission losses across all scenarios.

The matrix of the stochastic quadratic problem (3) with N scenarios, excluding variable bounds, has the dual block-angular structure

$$
\tilde{A} = \begin{bmatrix} \tilde{I} & & & & \\ M_1 & W_1 & & & \\ M_2 & & W_2 & & \\ \vdots & & & \ddots & \\ M_N & & & & W_N \end{bmatrix},
$$

where

$$
\tilde{I} = [I^1 \ldots I^t], \qquad M_j = \begin{bmatrix} \tilde{H}_1 & & \\ & \ddots & \\ & & \tilde{H}_t \end{bmatrix}, \quad W_j = \begin{bmatrix} \tilde{T}_1 & & \\ & \ddots & \\ & & \tilde{T}_t \end{bmatrix}, \quad j = 1, \ldots, N, \tag{4}
$$

for $I^i = I_{g_1}$, with $i = 1, \ldots, t$, being I_{g_1} an identity matrix $g_1 \times g_1$, and

$$
\tilde{H}_i = \begin{bmatrix} H_1 \\ 0 \end{bmatrix}, \quad \text{and } \tilde{T}_i = \begin{bmatrix} H_2 & -A \\ 0 & T \end{bmatrix}, \qquad \text{for } i = 1, \ldots, t. \tag{5}
$$

2.4 Splitting formulation for Block-Angular structure

The extensive form in (3) is computationally demanding for large-scale grids due to the dense coupling created by the first-stage variables p_i^1 across all scenarios. To enable the efficient application of IPMs, we use the technique of *splitting variables* to reduce fill-in and achieve a block-angular structure [17, 10]. To support this, we refer to the following proposition from [10].

Proposition 1 ([10, Proposition 1])**.** *Any multistage stochastic optimization problem can be recast as a primal block-angular problem through the splitting variable technique.*

Since the proposed model is a two-stage stochastic optimization problem, Proposition 1 applies directly.

Therefore, based on Proposition 1, by replicating the first-stage variables for each scenario, creating the variable p_{ij}^1, we decouple the scenario blocks. To ensure that the first-stage decision remains identical across all scenarios, satisfying the non-anticipativity requirement, we introduce explicit linking equality constraints $p_{ij}^1 - p_{i,j+1}^1 = 0$, for $i = 1, \ldots, t$

and $j = 1, \ldots, N-1$.. The model defined in (3) is thus reformulated as:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} \left\{ \frac{\beta_1}{2} \left((p_{i1}^1)^\top Q_1 p_{i1}^1 + c_1^\top p_{i1}^1 \right) + \sum_{j=1}^{N} a_j \left[\frac{\beta_2}{2} \left((p_{ij}^2)^\top Q_2 p_{ij}^2 + c_2^\top p_{ij}^2 \right) + \frac{\alpha}{2} f_{ij}^\top R f_{ij} \right] \right\} \\
\text{s.t.} \quad & \sum_{i=1}^{t} p_{i1}^1 = p_h \\
& \left. \begin{aligned} & H_1 p_{ij}^1 + H_2 p_{ij}^2 - A f_{ij} = d_{ij} \\ & T f_{ij} = 0 \\ & p_{\min}^2 \le p_{ij}^2 \le p_{\max}^2 \\ & f_{\min} \le f_{ij} \le f_{\max} \\ & p_{\min}^1 \le p_{ij}^1 \le p_{\max}^1 \end{aligned} \right\} i = 1, \ldots, t; j = 1, \ldots, N,
\end{aligned}
\tag{6}
$$

$$
p_{ij}^1 - p_{i,j+1}^1 = 0, \qquad i = 1, \ldots, t; j = 1, \ldots, N-1. \tag{7}
$$

The constraints (7) enforce non-anticipativity, which implies that the constraint matrix of problem (6)–(7), excluding the bound constraints, has the classic primal block-angular structure:

$$
\tilde{A} = \begin{bmatrix}
\tilde{I} & & & & & & & & & & \\
M_1 & W_1 & & & & & & & & & \\
& & M_2 & W_2 & & & & & & & \\
& & & & M_3 & W_3 & & & & & \\
& & & & & & \ddots & & & & \\
& & & & & & & M_{N-1} & W_{N-1} & & \\
& & & & & & & & & M_N & W_N \\
I & & -I & & & & & & & & \\
& & I & & -I & & & & & & \\
& & & & & \ddots & & & & & \\
& & & & & & & I & & -I &
\end{bmatrix}. \tag{8}
$$

The matrices M_j represent the connection of the hydro generators to the grid blocks, and W_j encapsulates the thermal generation and transmission network reactances. The bottom rows containing I and $-I$ correspond to the non-anticipativity constraints.

According to the definitions in equations (4) and (5), let

$$
A_1 = \begin{bmatrix} \tilde{I} & \\ M_1 & W_1 \end{bmatrix} ; A_j = [M_j \quad W_j], j = 2, \ldots, N.
$$

$$
L_1 = \begin{bmatrix} I & 0_1 \\ 0 & 0_2 \\ \vdots & \vdots \\ 0 & 0_{N-1} \end{bmatrix} ; L_j = \begin{bmatrix} 0 & 0_1 \\ \vdots & \vdots \\ 0 & 0_{j-2} \\ -I & 0_{j-1} \\ I & 0_j \\ 0 & 0_{j+1} \\ \vdots & \vdots \\ 0 & 0_{N-1} \end{bmatrix} j = 2, \ldots, N-1 ; L_N = \begin{bmatrix} 0 & 0_1 \\ \vdots & \vdots \\ 0 & 0_{N-2} \\ -I & 0_{N-1} \end{bmatrix}, \tag{9}
$$

where each $0_j \in (t \cdot g_1) \times t(g_2 + m)$ is a zero matrix, for $j = 1, \ldots, N-1$. Therefore, the constraints matrix of the block-angular formulation of the two-stage predispatch problem (6)-(7) can be written compactly as:

$$
\tilde{A} = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_N \\ L_1 & L_2 & \ldots & L_N \end{bmatrix}. \tag{10}
$$

In the following section, we outline the specialized IPM designed to exploit this exact block-angular structure (10).

3 Interior-point method for primal block-angular structures

IPMs are a class of algorithms for solving large-scale linear and quadratic optimization problems [29, 15, 28]. While each IPM iteration may be computationally more expensive than those of active-set methods, IPMs require fewer iterations and scale exceptionally well [15]. This makes it a preferred choice for complex power system optimization [25], including short-term scheduling [20], optimal power flow [4], and particularly stochastic optimization [16].

In this paper, we employ a primal-dual path-following IPM for quadratic optimization developed in [6, 8, 9]. Unlike conventional stochastic formulations, our split-variable formulation yields the block-angular constraint matrix $\tilde{A}$ in (8). This structure is explicitly exploited by the solver, enabling the efficient solution of the extensive-form two-stage stochastic predispatch problem. For completeness, we outline below this specialized IPM for block-angular structures.

3.1 Primal-Dual formulation and KKT conditions

For the variables of the predispatch with uncertain demand using the block-angular structure defined in Section 2.4, let $x_j = (p^1_{1j}, \dots, p^1_{tj}, p^2_{1j}, \dots, p^2_{tj}, f_{1j}, \dots, f_{tj})^\top$ for $j = 1, \dots, N$, and define the global state vector $x = (x_1^\top, \dots, x_N^\top)^\top$. Using the block-angular constraint matrix $\tilde{A}$ in (8) and appropriately matching vectors c, limits u, and the quadratic cost matrix Q, the primal problem can be written in standard form:

$$\begin{array}{ll} \min & c^\top x + \frac{1}{2} x^\top Q x \\ \text{s.t.} & \tilde{A}x = b, \\ & x + s = u, \\ & x, s \geq 0, \end{array} \tag{11}$$

where s is the vector of slack variables related to the power generation and flow limits. The corresponding dual problem is:

$$\begin{array}{ll} \max & b^\top y - \frac{1}{2} x^\top Q x - w^\top u \\ \text{s.t.} & \tilde{A}^\top y - Qx + z - w = c \\ & z, w \geq 0. \end{array}$$

To find the optimal solution, IPM relies on solving the perturbed Karush-Kuhn-Tucker (KKT) conditions. Introducing the barrier parameter $\mu > 0$, the central path is defined by the following system:

$$\begin{array}{ll} \left.\begin{array}{rl} \tilde{A}x & = b \\ x + s & = u \\ x, s \geq 0 & \end{array}\right\} & \text{Primal feasibility} \\ \left.\begin{array}{rl} \tilde{A}^\top y - Qx + z - w & = c \\ z, w \geq 0 & \end{array}\right\} & \text{Dual feasibility} \\ \left.\begin{array}{rl} XZe & = \mu e \\ SWe & = \mu e \end{array}\right\} & \text{Complementarity} \end{array} \tag{12}$$

where X, S, W, and Z are diagonal matrices whose entries are the components of the vectors x, s, w, and z, respectively, and e is a vector of ones. As the barrier parameter $\mu \to 0$, the solution of system (12) converges to the optimal solution of problem (11) [29].

Because the complementarity conditions in (12) are nonlinear, IPM applies the Newton method to find the search direction $(\Delta x, \Delta y, \Delta z, \Delta w)$. This requires solving the following linear system at each iteration:

$$\begin{bmatrix} -Q & \tilde{A}^\top & I & \\ \tilde{A} & & & \\ Z & & X & \\ -W & & & S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta w \end{bmatrix} = \begin{bmatrix} c - (\tilde{A}^\top y - Qx + z - w) \\ b - \tilde{A}x \\ \mu e - XZe \\ \mu e - SWe \end{bmatrix} := \begin{bmatrix} r_c \\ r_b \\ r_{xz} \\ r_{sw} \end{bmatrix}. \tag{13}$$

After eliminating Δz and Δw in (13):

$$\Delta z = X^{-1}(r_{xz} - Z\Delta x),$$
$$\Delta w = S^{-1}(r_{sw} + W\Delta x),$$

the system reduces to the *augmented system*:

$$\begin{bmatrix} -\Theta^{-1} & \tilde{A}^\top \\ \tilde{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ r_b \end{bmatrix}, \tag{14}$$

where $\Theta = (Q + S^{-1}W + X^{-1}Z)^{-1}$ and $r = r_c + S^{-1}r_{sw} - X^{-1}r_{xz}$ combines the residuals. Since Q is positive definite and the IPM maintains strictly positive iterates, Θ^{-1} is a well-defined positive definite diagonal matrix. Further eliminating $\Delta x = \Theta(\tilde{A}^\top \Delta y - r)$ from the first block of (14) yields the strictly symmetric positive-definite *normal equations* for the dual step Δy:

$$\tilde{A}\Theta\tilde{A}^\top \Delta y = r_b + \tilde{A}\Theta(r_c + S^{-1}r_{sw} - X^{-1}r_{xz}) := q. \tag{15}$$

The positive definiteness of $\tilde{A}\Theta\tilde{A}^\top$ follows from the full row rank of $\tilde{A}$ and the positive definiteness and well-definition of Θ because the IPM maintains strictly positive entries for X, Z, S, and W [29, 15].

The computational bottleneck of IPM lies in factorizing the normal equations matrix $\tilde{A}\Theta\tilde{A}^\top$. However, because we applied splitting variables in Section 2, the matrix $\tilde{A}$ has the primal block-angular structure shown in (10). Consequently, the normal equations matrix takes the following structured form:

$$\begin{aligned}\tilde{A}\Theta\tilde{A}^\top &= \left[\begin{array}{ccc|c} A_1\Theta_1A_1^\top & & & A_1\Theta_1L_1^\top \\ & \ddots & & \vdots \\ & & A_N\Theta_NA_N^\top & A_N\Theta_NL_N^\top \\ \hline L_1\Theta_1A_1^\top & \dots & L_N\Theta_NA_N^\top & \Theta_0 + \sum_{i=1}^N L_i\Theta_iL_i^\top \end{array}\right] \\ &:= \begin{bmatrix} B & C \\ C^\top & D \end{bmatrix}.\end{aligned} \tag{16}$$

To solve (15) efficiently based on the structure of normal equations in (16):

$$\begin{bmatrix} B & C \\ C^\top & D \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix},$$

we apply block Gaussian elimination to obtain the Schur complement system:

$$\left(D - C^\top B^{-1}C\right)\Delta y_2 = \left(q_2 - C^\top B^{-1}q_1\right) \tag{17}$$

$$B\Delta y_1 = (q_1 - C\Delta y_2)\,. \tag{18}$$

The advantage of this decomposition is that the matrix B is block-diagonal. Therefore, computing systems of equations with B does not require a massive global factorization; instead, it simply requires N independent, smaller Cholesky factorizations for each $A_j\Theta_jA_j^\top$ system. These factorizations can also be parallelized [8, 10].

While solving (18) is straightforward, solving the Schur complement system (17) exactly is computationally prohibitive due to the fill-in caused by $C^\top B^{-1}C$. To overcome this, we employ the Preconditioned Conjugate Gradient (PCG) iterative method. The following result from [6] is used to derive the preconditioner:

Theorem 1 (Spectral radius bound [6])**.** *Let B, C, and D be defined as in (16). The eigenvalues of the matrix $D^{-1}(C^\top B^{-1}C)$ lie strictly within the interval $[0, 1)$, i.e.,*

$$\rho(D^{-1}(C^\top B^{-1}C)) < 1.$$

Consequently, the Neumann series $\sum_{i=0}^{\infty}(D^{-1}(C^\top B^{-1}C))^i$ converges, and the inverse of the Schur complement can be expressed as

$$(D - C^\top B^{-1}C)^{-1} = \left(\sum_{i=0}^{\infty}(D^{-1}(C^\top B^{-1}C))^i\right)D^{-1}. \tag{19}$$

Based on Theorem 1 and following [6], to solve the system (17) efficiently with PCG, the preconditioner P is constructed by truncating the Neumann series in (19). For stochastic optimization, taking a partial sum with one term provides the optimal balance between iteration reduction and computational overhead per iteration [10], yielding the preconditioner:

$$P = \Theta_0 + \sum_{i=1}^{N} L_i\Theta_iL_i^\top.$$

Remark 1 (Parallelization potential)**.** The block-diagonal structure of B is one of the key computational advantages of the splitting variable approach. At each IPM iteration, the solution of system (18) and each PCG iteration requires solving N independent linear systems with coefficient matrices $A_j\Theta_jA_j^\top$, $j = 1, \dots, N$. These factorizations are:

1. **Independent**: each factorization involves only the data of a single scenario, enabling parallel computation across N processors;
2. **Structurally identical**: for the predispatch problem, the sparsity pattern of $A_j\Theta_jA_j^\top$ is the same for all scenarios $j = 2, \dots, N$ (since $A_j, j > 1$ share the same network topology, as defined in (9)), so in theory symbolic factorization needs to be performed only once. The current implementation only exploits this fact when all matrices A_j are equal, which is not the case since A_1 is defined differently from A_j, $j > 1$, as seen in (9).

3. **Small-scale**: for $j = 2, \ldots, N$, each system has dimension $t(n+1)$ which is independent of the number of scenarios N.

This property is shared with other decomposition approaches for stochastic optimization, such as the L-shaped method [1] and SDDP [22], but the IPM approach has the advantage of solving all scenarios simultaneously within a single optimization framework.

For the two-stage predispatch problem (6)–(7), the linking matrices L_j consist of identity and negative identity blocks of dimension $t \cdot g_1$ (the total number of first-stage hydro generation variables across all time periods). Consequently, following Proposition 2 of [10], the preconditioner $P = \Theta_0 + \sum_{j=1}^{N} L_j \Theta_j L_j^\top$ is a v-shifted tridiagonal matrix, where $v = t \cdot g_1$. Specifically:

$$P = \begin{bmatrix} \Theta_{x_1} + \Theta_{x_2} & -\Theta_{x_2} & & & \\ -\Theta_{x_2} & \Theta_{x_2} + \Theta_{x_3} & -\Theta_{x_3} & & \\ & \ddots & \ddots & \ddots & \\ & & -\Theta_{x_{N-1}} & \Theta_{x_{N-1}} + \Theta_{x_N} \end{bmatrix} + \Theta_0,$$

where Θ_{x_j} denotes the diagonal submatrix of Θ_j corresponding to the replicated first-stage variables p_{ij}^1 in scenario j. This shifted tridiagonal structure can be factorized with zero fill-in, making the preconditioner application computationally inexpensive regardless of the number of scenarios N.

4 The stochastic solution and information value

To quantify the operational and economic benefits of solving the extensive form of the predispatch optimal power flow over a simplified deterministic approach, we introduce the fundamental metrics of stochastic optimization based on [1]. These metrics evaluate the quality of decisions made under uncertainty and the potential value of obtaining better demand forecasts. They are used in Subsection 5.5.1.

Let x represent the first-stage decision variables (hydroelectric generation) and ξ the random variable representing the uncertain demand. The objective function evaluated for a decision x under realization ξ is denoted as $z(x, \xi)$.

First, we define the baseline where the decision-maker has perfect knowledge of future demand.

Definition 2. The **Wait-and-See (WS)** solution represents the expected objective value if the decision-maker had perfect information about the future before making the first-stage decisions. It is the expected value of the optimal solutions computed independently for each individual scenario:

$$WS = \mathbb{E}_\xi \left[\min_x z(x, \xi) \right].$$

In reality, perfect information is unavailable. The first-stage decision must be made *before* the uncertainty is resolved. Therefore, instead of taking the expected value of the individual minimums (as in WS), we must find a single initial decision that minimizes the expected value across all scenarios. This leads to the stochastic optimization model we defined in Section 2, known as the Recourse Problem.

Definition 3. The **Recourse Problem (RP)** solution is the optimal here-and-now decision made at the beginning of the planning horizon, considering all possible scenarios and their probabilities. It is the objective value of our multi-scenario stochastic optimization problem:

$$RP = \min_x \mathbb{E}_\xi [z(x, \xi)].$$

Definition 4. The **Expected Value (EV)** problem is the deterministic optimization problem where the random variables are replaced by their expected values $\bar{\xi} = \mathbb{E}[\xi]$:

$$EV = \min_x z(x, \bar{\xi}).$$

If $\bar{x}(\bar{\xi})$ is the optimal first-stage decision obtained from solving the EV problem, we must evaluate how this "average" decision actually performs in the real world where demand fluctuates.

Definition 5. The **Expected result of using the EV solution (EEV)** is the expected cost of implementing the first-stage decision $\bar{x}(\bar{\xi})$ across all possible real-world scenarios:

$$EEV = \mathbb{E}_\xi [z(\bar{x}(\bar{\xi}), \xi)].$$

With these values defined, we can establish two metrics that justify the application of stochastic optimization [1].

Definition 6. The **Value of the Stochastic Solution (VSS)** measures the cost of ignoring uncertainty. It quantifies how much is saved by solving the Recourse Problem instead of implementing the deterministic Expected Value solution:

$$VSS = EEV - RP.$$

Definition 7. The **Expected Value of Perfect Information (EVPI)** measures the maximum amount the decision-maker should be willing to pay for a perfect forecast. It is the difference between the Recourse Problem and the Wait-and-See solution:

$$EVPI = RP - WS.$$

Because minimizing the expected cost with additional flexibility is always advantageous, the relationships between these models are bounded by a fundamental set of inequalities.

Proposition 2 ([1])**.** *For any stochastic minimization problem, the following inequalities hold:*

$$WS \leq RP \leq EEV.$$

Consequently, both the Value of the Stochastic Solution and the Expected Value of Perfect Information are non-negative ($VSS \geq 0$ and $EVPI \geq 0$).

Remark 2. The inequality chain $WS \leq RP \leq EEV$ has a direct operational interpretation for the predispatch problem. The gap $EVPI = RP - WS$ represents the maximum economic benefit that would be achieved by a *perfect demand forecast*, which quantifies the value of investing in better demand prediction technologies. The gap $VSS = EEV - RP$ represents the cost penalty incurred by the grid operator who ignores demand variability and schedules hydroelectric plants based solely on the average demand forecast. In practice, if VSS is large relative to RP, the operator has a strong economic incentive to adopt the stochastic predispatch model rather than the simpler deterministic approach [1].

5 Computational experiments

This section presents the computational study of the proposed two-stage stochastic predispatch model in Section 2 solved with a tailored stochastic optimization solver based on BlockIP, an efficient implementation of the IPM for block-angular problems [10]). All experiments were performed on a Linux machine equipped with an Intel Core Ultra 7 155H processor (up to 4.80 GHz, 16 cores, 22 threads) and 32 GB of RAM. The reference solvers used for comparison are CPLEX 22.1, Gurobi 13.0.1 and HiPO-HiGHS 1.15. We purposely excluded the use of larger servers, with more than 32GB of RAM, to observe the potential of all four tested solvers (BlockIP, CPLEX, Gurobi, HiPO-HiGHS) to solve these instances in moderate hardware. For every test, we fix a 24-hour planning horizon ($t = 24$) with $N = 10$ and $N = 100$ demand scenarios, as described in Sections 5.2 and 5.3.

5.1 Network descriptions

Seven transmission networks of increasing size are considered. Table 1 summarizes their structural characteristics. For all systems, $2/3$ of the installed generation capacity is assigned to hydroelectric plants (first-stage) and the remaining $1/3$ to thermal plants (second-stage). Generation lower bounds are set to zero; upper bounds on active power flow follow each specification of the system. Thermal generation costs are set fifteen times higher than hydroelectric costs to reflect typical hydro-thermal economic ratios.

Table 1: Topological characteristics of the test networks.

Network	Buses	Lines	Gen.	g_1 (hydro)	g_2 (thermal)
IEEE30	30	41	6	4	2
IEEE118	118	186	50	33	17
SSE810	810	1,091	81	54	27
SSE1654	1,654	2,063	86	57	29
SSE1732	1,732	2,160	81	54	27
BRAZIL	1,993	2,476	121	80	41
CASE9241PEGASE	9,241	14,207	1,445	963	482

5.2 Energy demand

To conduct an academic analysis of optimizing the *RP* using a block-based method, we employ demand data from Brazil. The temporal shape of the load, that is, the proportion of hourly demand relative to the daily peak, is derived from the hourly load curve dataset provided by the Operador Nacional de Sistema Elétrico (ONS) [26] for the period 01/01/2022 to 31/12/2025. This yields a normalized 24-hour profile. However, the absolute magnitude of the load at each bus is network-case dependent: each power system model supplies its own bus demand values. The final hourly demand for a given bus is obtained by scaling its network-specific peak by the ONS-derived hourly proportion.

5.3 Scenario definition

The construction of discrete demand scenarios is crucial for the quality of the stochastic solution. We generate $N = 10$ and $N = 100$ scenarios for the demand over the 24-hour planning horizon using a two-step approach

Historical scenarios. A subset of $0.8N$ scenarios is derived from the ONS historical database using a Monte Carlo simulation-based sampling approach. We first verified that the hourly demand data did not satisfy the normality assumption by applying the Shapiro-Wilk test at a significance level of 5% (p-value < 0.01), which rejected the null hypothesis of normality. Consequently, rather than assuming a parametric distribution, we employ stratified sampling from the empirical distribution of historical demand profiles. We assume that 98% of the events fall within this historical range. Therefore, each historical scenario is assigned an equal probability of $a_j = \frac{0.98}{0.8N}$, for $j = 1, \ldots, 0.8N$.

Extreme scenarios. To capture tail risk and stress-test the system, the remaining $0.2N$ scenarios represent a $\pm 20\%$ perturbation relative to the mean historical demand profile. Assuming extreme events occur with a 2% probability, each of these extreme scenarios is assigned a probability of $a_j = \frac{0.02}{0.2N}$, for $j = 0.8N + 1, \ldots, N$. These extreme scenarios ensure that the stochastic solution accounts for demand realizations significantly different from the historical range, which is critical for maintaining system reliability during atypical conditions [1, 24]. The resulting probability weights satisfy $\sum_{j=1}^{N} a_j = 1$, and the expected demand $\bar{d} = \sum_{j=1}^{N} a_j d_j$ closely approximates the empirical mean of the historical dataset.

Here is how the scenario counts and probability weights (a_j) scale for the specific tested values of N:

For $N = 10$:

- **Historical:** 8 scenarios.
  - Probability per scenario: $a_j = \frac{0.98}{8} = 0.1225$
- **Extreme:** 2 scenarios .
  - Probability per scenario: $a_j = \frac{0.02}{2} = 0.01$

For $N = 100$:

- **Historical:** 80 scenarios.
  - Probability per scenario: $a_j = \frac{0.98}{80} = 0.01225$
- **Extreme:** 20 scenarios.
  - Probability per scenario: $a_j = \frac{0.02}{20} = 0.001$

The spread of the scenario set and the amount of available recourse jointly govern how much the stochastic solution can improve upon a deterministic, mean-value schedule: wider demand variability and tighter recourse enlarge the value of the stochastic solution, whereas a moderate scenario set combined with ample recourse leaves less room for improvement. The scenario construction we adopted, based on historical realizations together with symmetric $\pm 20\%$ tail scenarios, is one such moderate design.

5.4 Computational results and comparison of BlockIP with CPLEX, Gurobi, and HiPO-HiGHS

For the present paper, we obtain the *RP* solution according to the block structure of the constraint matrix defined in (10) using the MSSO-BlockIP solver [10] which runs on top of BlockIP. The results are compared against three state-of-the-art commercial and open-source solvers: CPLEX 22.1 (barrier method), Gurobi 13.0.1 (barrier method), and HiPO-HiGHS 1.15.1 (interior-point solver). No wall-clock time limit was imposed on any solver; each run was allowed to proceed until the solver reported termination (optimal, suboptimal, or failure) on its own.

We evaluate performance across the seven transmission networks of Table 1, from `IEEE30` with 30 buses to `CASE9241PEGASE` with 9,241 buses. The optimality tolerance was set to 10^{-3} for all four solvers, which is a reasonable value for large stochastic optimization problems. The crossover option was disabled for CPLEX, Gurobi and HiPO-HiGHS, while default values were used for all other parameters. We considered three objective configurations $(\alpha, \beta) \in \{(0,1), (1,0), (1,1)\}$ (where $\beta_1 = \beta_2 = \beta$), three thread counts $T_h \in \{1, 10, 22\}$ for $N = 10$ scenarios, and a large-scale test with $N = 100$ scenarios and $T_h = 22$ threads. The problem dimension of each tested system per number of scenarios is detailed in Table 2; the MPS files for all these instances are available from the authors upon request. The complete numerical results are reported in Tables 3–6. Since no time limit is enforced, every run terminates on its own, either with a solution or with a failure status. We use the following notation: a superscript $^{\mathrm{s}}$ on an objective value marks a *suboptimal* termination, in which the solver returns a feasible objective value but its barrier stops without certifying optimality; a dash (—) indicates that no usable solution was returned, further qualified as $^{\mathrm{u}}$ unknown (the solver stops without reporting a solution, typically after exhausting available memory), $^{\mathrm{i}}$ infeasible (the solver terminates declaring the instance infeasible), or $^{\mathrm{e}}$ solver error. Objective values without a superscript correspond to certified optimal solutions.

Table 2: Problem dimensions per test system and number of scenarios.

System	Scenarios	Variables	Constraints	Blocks
`IEEE30`	10	22,176	11,044	11
	100	220,896	110,404	101
`IEEE118`	10	106,632	51,153	11
	100	1,059,192	511,233	101
`SSE810`	10	556,896	274,854	11
	100	5,557,296	2,748,054	101
`SSE1654`	10	1,025,928	509,097	11
	100	10,246,968	5,090,457	101
`SSE1732`	10	1,070,496	531,654	11
	100	10,693,296	5,316,054	101
`BRAZIL`	10	1,238,640	613,760	11
	100	12,369,120	6,136,880	101
`CASE9241`	10	7,420,392	3,642,003	11
	100	73,995,912	36,411,363	101

Table 3: Computational results for 10 scenarios, 1 thread.

Network	(α, β)	**Objective value**				**Wall-clock Time (s)**				**Iterations**			
		BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS
`IEEE30`	$(0,1)$	9.13695×10^4	9.13512×10^4	9.13385×10^4	9.13382×10^4	0.38	0.12	0.64	0.46	25	37	43	22
	$(1,0)$	3.51845×10^5	3.51842×10^5	3.51789×10^5	3.51788×10^5	0.65	0.12	0.62	0.58	27	39	50	30
	$(1,1)$	4.44261×10^5	4.44239×10^5	4.44176×10^5	4.44176×10^5	0.98	0.12	0.77	0.54	27	39	59	26
`IEEE118`	$(0,1)$	1.06798×10^6	1.06801×10^6	1.06759×10^6	1.06759×10^6	1.01	1.95	3.34	3.22	21	56	38	19
	$(1,0)$	2.69084×10^8	2.69492×10^8	2.69468×10^8	2.69468×10^8	2.25	2.18	5.35	4.29	25	60	50	27
	$(1,1)$	2.72225×10^8	2.72196×10^8	2.72179×10^8	2.72179×10^8	3.93	2.10	2.62	4.22	25	60	25	27
`SSE810`	$(0,1)$	6.49934×10^7	$6.49928 \times 10^{7\text{s}}$	6.49867×10^7	6.49864×10^7	21.88	11.06	24.24	22.41	36	52	38	23
	$(1,0)$	1.00902×10^9	1.00864×10^9	1.00855×10^9	1.00855×10^9	69.24	12.37	112.84	31.58	35	59	147	34
	$(1,1)$	1.08848×10^9	1.08833×10^9	1.08820×10^9	1.08828×10^9	43.60	12.01	543.60	31.00	31	57	616	35
`SSE1654`	$(0,1)$	3.32744×10^9	3.32718×10^9	3.32701×10^9	3.32691×10^9	44.49	23.36	43.90	33.41	31	59	32	20
	$(1,0)$	1.32029×10^{10}	1.32029×10^{10}	1.32021×10^{10}	1.31294×10^{10}	49.14	22.05	51.93	57.98	37	56	31	34
	$(1,1)$	1.89307×10^{10}	1.89285×10^{10}	1.89285×10^{10}	1.88549×10^{10}	96.88	21.61	54.63	48.17	33	56	33	31
`SSE1732`	$(0,1)$	4.31971×10^9	4.31869×10^9	4.31841×10^9	4.31825×10^9	30.83	16.91	48.11	35.97	30	43	34	21
	$(1,0)$	1.55770×10^{10}	1.55739×10^{10}	1.55716×10^{10}	1.54910×10^{10}	54.99	16.42	149.05	69.89	35	42	64	42
	$(1,1)$	2.19064×10^{10}	2.19050×10^{10}	2.19043×10^{10}	2.18060×10^{10}	114.31	16.83	216.59	50.66	33	43	89	32
`BRAZIL`	$(0,1)$	3.59797×10^9	3.59782×10^9	3.59750×10^9	3.59731×10^9	44.99	26.60	57.21	40.06	31	61	33	20
	$(1,0)$	5.21995×10^{10}	5.22011×10^{10}	5.21934×10^{10}	5.21191×10^{10}	101.50	25.34	205.58	66.08	36	58	76	33
	$(1,1)$	5.98634×10^{10}	5.98504×10^{10}	5.98415×10^{10}	5.97738×10^{10}	98.38	24.61	206.03	56.07	33	57	76	30
`CASE9241`	$(0,1)$	2.12238×10^{10}	$2.75616 \times 10^{10\text{s}}$	2.12223×10^{10}	2.12169×10^{10}	495.40	682.41	1285.25	595.00	34	50	50	24
	$(1,0)$	3.45485×10^{12}	$2.00678 \times 10^{15\text{s}}$	3.45584×10^{12}	3.45578×10^{12}	1418.79	671.01	1307.44	929.26	41	50	46	42
	$(1,1)$	3.52645×10^{12}	$2.00681 \times 10^{15\text{s}}$	3.52632×10^{12}	3.52626×10^{12}	2137.02	673.15	1453.41	837.14	41	50	46	41

Remark 3. The objective values reported in the tables do not include the constant term introduced by the variable transformation in model (3), where $p_{\min} = 0$ and $f_{\min} = -f_{\max}$. This constant shift does not affect the optimal solution or the relative comparison between instances, but it should be taken into account when interpreting absolute objective values.

Solver robustness

Across all configurations with $N = 10$ scenarios (Tables 3–5), BlockIP produces a certified optimal solution for every network and every (α, β) configuration, demonstrating complete robustness across the entire benchmark set. CPLEX returns a feasible objective value on all 63 of the $N = 10$ runs, but on 20 of them its barrier stops at a suboptimal point without certifying optimality: these suboptimal terminations concentrate on all three configurations of `IEEE30` (at $T_h = 10$ and 22), on the generation-cost configuration $(\alpha = 0, \beta = 1)$ of `IEEE118` and `SSE810`, and on every configuration of `CASE9241PEGASE` at all three thread counts. The `CASE9241PEGASE` case is the most remarkable: under the loss and mixed configurations CPLEX not only stops suboptimally but returns objective values that are inaccurate

Table 4: Computational results for 10 scenarios, 10 threads.

Network	(α, β)	**Objective value**				**Wall-clock Time (s)**				**Iterations**			
		BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS
IEEE30	$(0,1)$	9.13695×10^{4}	$9.13512 \times 10^{4\,\mathrm{s}}$	9.13385×10^{4}	9.13382×10^{4}	0.22	0.16	0.76	0.45	25	37	44	22
	$(1,0)$	3.51845×10^{5}	$3.51842 \times 10^{5\,\mathrm{s}}$	3.51788×10^{5}	3.51788×10^{5}	0.41	0.17	0.64	0.56	27	39	49	30
	$(1,1)$	4.44261×10^{5}	$4.44239 \times 10^{5\,\mathrm{s}}$	4.44177×10^{5}	4.44176×10^{5}	0.58	0.16	0.79	0.52	27	39	56	26
IEEE118	$(0,1)$	1.06798×10^{6}	$1.06801 \times 10^{6\,\mathrm{s}}$	1.06759×10^{6}	1.06759×10^{6}	0.34	1.01	2.43	2.89	21	56	39	19
	$(1,0)$	2.69084×10^{8}	2.69492×10^{8}	2.69468×10^{8}	2.69468×10^{8}	0.72	1.02	4.11	3.70	25	60	52	27
	$(1,1)$	2.72225×10^{8}	2.72196×10^{8}	2.72179×10^{8}	2.72179×10^{8}	1.21	1.07	2.15	3.71	25	60	28	27
SSE810	$(0,1)$	6.49934×10^{7}	$6.49928 \times 10^{7\,\mathrm{s}}$	6.49866×10^{7}	6.49864×10^{7}	9.45	5.95	20.76	22.43	36	52	42	23
	$(1,0)$	1.00902×10^{9}	1.00864×10^{9}	1.00855×10^{9}	1.00855×10^{9}	29.23	6.64	96.77	32.37	35	59	155	34
	$(1,1)$	1.08848×10^{9}	1.08833×10^{9}	$1.03205 \times 10^{9\,\mathrm{s}}$	1.08828×10^{9}	18.59	6.34	247.89	31.60	31	57	333	35
SSE1654	$(0,1)$	3.32744×10^{9}	3.32718×10^{9}	3.32701×10^{9}	3.32691×10^{9}	16.91	11.90	35.64	33.54	31	59	32	20
	$(1,0)$	1.32029×10^{10}	1.32029×10^{10}	1.32020×10^{10}	1.31294×10^{10}	19.06	11.56	40.86	57.57	37	56	31	34
	$(1,1)$	1.89307×10^{10}	1.89285×10^{10}	1.89285×10^{10}	1.88549×10^{10}	36.89	11.59	64.49	49.38	33	56	38	31
SSE1732	$(0,1)$	4.31971×10^{9}	4.31869×10^{9}	4.31840×10^{9}	4.31825×10^{9}	11.66	9.11	33.96	36.13	30	43	31	21
	$(1,0)$	1.55770×10^{10}	1.55739×10^{10}	1.55716×10^{10}	1.54910×10^{10}	21.30	9.10	125.96	70.54	35	42	64	42
	$(1,1)$	2.19064×10^{10}	2.19050×10^{10}	2.19043×10^{10}	2.18060×10^{10}	43.53	9.07	213.07	50.89	33	43	84	32
BRAZIL	$(0,1)$	3.59797×10^{9}	3.59782×10^{9}	3.59750×10^{9}	3.59731×10^{9}	17.08	13.42	46.30	38.70	31	61	31	20
	$(1,0)$	5.21995×10^{10}	5.22011×10^{10}	5.21934×10^{10}	5.21191×10^{10}	37.94	12.73	201.77	67.70	36	58	73	33
	$(1,1)$	5.98634×10^{10}	5.98504×10^{10}	5.98415×10^{10}	5.97738×10^{10}	36.95	12.94	205.44	54.67	33	57	83	30
CASE9241	$(0,1)$	2.12238×10^{10}	$2.75617 \times 10^{10\,\mathrm{s}}$	2.12228×10^{10}	2.12169×10^{10}	110.86	231.38	809.54	422.53	34	50	50	24
	$(1,0)$	3.45485×10^{12}	$2.00672 \times 10^{15\,\mathrm{s}}$	3.45584×10^{12}	3.45578×10^{12}	315.15	230.97	893.30	652.71	41	50	44	42
	$(1,1)$	3.52645×10^{12}	$2.00676 \times 10^{15\,\mathrm{s}}$	3.52634×10^{12}	3.52626×10^{12}	474.39	230.75	869.39	571.53	41	50	43	41

Table 5: Computational results for 10 scenarios, 22 threads.

Network	(α, β)	**Objective value**				**Wall-clock Time (s)**				**Iterations**			
		BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS
IEEE30	$(0,1)$	9.13695×10^{4}	$9.13512 \times 10^{4\,\mathrm{s}}$	9.13385×10^{4}	9.13382×10^{4}	0.47	0.16	1.12	0.47	25	37	44	22
	$(1,0)$	3.51845×10^{5}	$3.51842 \times 10^{5\,\mathrm{s}}$	3.51788×10^{5}	3.51788×10^{5}	0.66	0.17	1.33	0.58	27	39	49	30
	$(1,1)$	4.44261×10^{5}	$4.44239 \times 10^{5\,\mathrm{s}}$	4.44177×10^{5}	4.44176×10^{5}	0.99	0.18	2.35	0.51	27	39	56	26
IEEE118	$(0,1)$	1.06798×10^{6}	$1.06801 \times 10^{6\,\mathrm{s}}$	1.06759×10^{6}	1.06759×10^{6}	0.42	0.96	3.06	3.01	21	56	39	19
	$(1,0)$	2.69084×10^{8}	2.69492×10^{8}	2.69468×10^{8}	2.69468×10^{8}	0.89	1.06	4.92	3.96	25	60	52	27
	$(1,1)$	2.72225×10^{8}	2.72196×10^{8}	2.72179×10^{8}	2.72179×10^{8}	1.53	1.02	2.76	3.94	25	60	28	27
SSE810	$(0,1)$	6.49934×10^{7}	$6.49928 \times 10^{7\,\mathrm{s}}$	6.49866×10^{7}	6.49864×10^{7}	10.02	6.77	26.00	22.22	36	52	42	23
	$(1,0)$	1.00902×10^{9}	1.00864×10^{9}	$9.64871 \times 10^{8\,\mathrm{s}}$	1.00855×10^{9}	31.41	7.57	303.89	31.64	35	59	341	34
	$(1,1)$	1.08848×10^{9}	1.08833×10^{9}	$1.03258 \times 10^{9\,\mathrm{s}}$	1.08828×10^{9}	19.53	7.31	285.52	30.87	31	57	323	35
SSE1654	$(0,1)$	3.32744×10^{9}	3.32718×10^{9}	3.32702×10^{9}	3.32691×10^{9}	19.57	13.12	63.43	32.93	31	59	34	20
	$(1,0)$	1.32029×10^{10}	1.32029×10^{10}	1.32020×10^{10}	1.31294×10^{10}	21.56	12.81	48.45	58.93	37	56	32	34
	$(1,1)$	1.89307×10^{10}	1.89285×10^{10}	1.89284×10^{10}	1.88549×10^{10}	42.73	12.51	72.52	49.83	33	56	41	31
SSE1732	$(0,1)$	4.31971×10^{9}	4.31869×10^{9}	4.31840×10^{9}	4.31825×10^{9}	13.55	9.86	46.36	36.67	30	43	31	21
	$(1,0)$	1.55770×10^{10}	1.55739×10^{10}	1.55716×10^{10}	1.54910×10^{10}	24.42	9.72	211.97	70.24	35	42	64	42
	$(1,1)$	2.19064×10^{10}	2.19050×10^{10}	2.19043×10^{10}	2.18060×10^{10}	50.68	9.85	228.36	49.39	33	43	84	32
BRAZIL	$(0,1)$	3.59797×10^{9}	3.59793×10^{9}	3.59750×10^{9}	3.59731×10^{9}	19.49	14.49	53.60	39.78	31	61	33	20
	$(1,0)$	5.21995×10^{10}	5.21995×10^{10}	5.21934×10^{10}	5.21191×10^{10}	44.25	13.81	256.06	66.08	36	58	73	33
	$(1,1)$	5.98634×10^{10}	5.98502×10^{10}	5.98415×10^{10}	5.97738×10^{10}	42.14	13.52	288.31	55.37	33	57	83	30
CASE9241	$(0,1)$	2.12238×10^{10}	$2.75644 \times 10^{10\,\mathrm{s}}$	2.12226×10^{10}	2.12169×10^{10}	107.49	245.25	902.39	421.63	34	50	43	24
	$(1,0)$	3.45485×10^{12}	$2.00419 \times 10^{15\,\mathrm{s}}$	3.45592×10^{12}	3.45578×10^{12}	307.63	245.96	920.86	654.89	41	50	42	42
	$(1,1)$	3.52645×10^{12}	$2.00408 \times 10^{15\,\mathrm{s}}$	3.52633×10^{12}	3.52626×10^{12}	457.50	246.15	1105.00	576.70	41	50	44	41

(about 2.0×10^{15} against the certified optimum of roughly 3.5×10^{12}), so that its apparently short running times on this network do not correspond to usable solutions. HiPO-HiGHS certifies optimality on all 63 of the $N = 10$ instances, whereas Gurobi, although it certifies optimality on most instances, terminates suboptimally on 3 of them: it returns a feasible but uncertified solution (suboptimal status) on `SSE810` under $(1, 1)$ at $T_h = 10$ and under both $(1, 0)$ and $(1, 1)$ at $T_h = 22$, where its barrier stops before certifying optimality. In terms of running time on the $N = 10$ set, CPLEX is generally the fastest of the four solvers; BlockIP is competitive on the small networks and is in fact the fastest solver on several instances (for example `IEEE118` under $(\alpha, \beta) = (0, 1)$ at $T_h = 22$, solved in 0.42 s versus 0.96 s for CPLEX, and `CASE9241PEGASE` under $(0, 1)$ at $T_h = 22$, solved in 107 s versus 245 s for CPLEX, which there terminates suboptimally). Gurobi, by contrast, is the slowest solver on most of the medium and large networks, with extreme outliers on the `SSE810` loss and mixed configurations (for instance 544 s and 616 barrier iterations on `SSE810` under $(1, 1)$ at $T_h = 1$) before terminating suboptimally at higher thread counts.

Table 6: Computational results for 100 scenarios, 22 threads.

Network	(α, β)	Objective value				Wall-clock Time (s)				Iterations			
		BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS	BlockIP	CPLEX	Gurobi	HiPO-HiGHS
IEEE30	$(0,1)$	1.33578×10^5	$2.57253 \times 10^{6\mathrm{s}}$	$—^{\mathrm{i}}$	$—^{\mathrm{i}}$	15.54	3.49	—	—	7	137	—	—
	$(1,0)$	1.24308×10^{13}	$3.46542 \times 10^{14\mathrm{s}}$	$—^{\mathrm{i}}$	$—^{\mathrm{e}}$	15.74	1.49	—	—	7	51	—	—
	$(1,1)$	1.40525×10^{13}	$9.33782 \times 10^{14\mathrm{s}}$	$—^{\mathrm{i}}$	$—^{\mathrm{e}}$	13.42	4.56	—	—	7	178	—	—
IEEE118	$(0,1)$	1.04515×10^6	1.04482×10^6	1.04479×10^6	1.04479×10^6	13.13	15.01	22.96	37.44	39	63	29	22
	$(1,0)$	2.79502×10^8	2.79471×10^8	2.79446×10^8	2.79445×10^8	85.62	15.81	57.17	46.76	38	67	32	29
	$(1,1)$	2.83820×10^8	2.83815×10^8	2.83806×10^8	2.83804×10^8	57.53	17.02	50.07	47.89	38	68	30	30
SSE810	$(0,1)$	6.02742×10^7	$6.02631 \times 10^{7\mathrm{s}}$	5.99726×10^7	$—^{\mathrm{e}}$	97.61	148.44	130.06	—	45	71	31	—
	$(1,0)$	9.83605×10^8	9.83949×10^8	9.83364×10^8	$—^{\mathrm{e}}$	232.77	155.47	773.06	—	49	76	100	—
	$(1,1)$	1.05504×10^9	$1.05486 \times 10^{9\mathrm{s}}$	1.05465×10^9	$—^{\mathrm{e}}$	118.65	153.33	542.13	—	46	75	69	—
SSE1654	$(0,1)$	3.27443×10^9	$3.27350 \times 10^{9\mathrm{s}}$	3.27336×10^9	3.23886×10^9	108.06	206.19	223.82	490.16	36	65	27	24
	$(1,0)$	1.31477×10^{10}	1.31499×10^{10}	1.31484×10^{10}	1.06743×10^{10}	265.65	198.07	513.09	1070.16	39	62	27	64
	$(1,1)$	1.87206×10^{10}	1.87155×10^{10}	1.87149×10^{10}	1.69944×10^{10}	220.96	194.78	601.27	695.98	35	61	28	37
SSE1732	$(0,1)$	4.23680×10^9	$4.23789 \times 10^{9\mathrm{s}}$	4.23603×10^9	4.19196×10^9	116.54	151.27	208.76	520.88	35	52	27	23
	$(1,0)$	1.54636×10^{10}	1.54644×10^{10}	1.54619×10^{10}	1.30748×10^{10}	309.10	153.95	509.46	3718.47	38	54	27	240
	$(1,1)$	4.20856×10^{11}	4.20870×10^{11}	4.22172×10^{11}	3.94616×10^{11}	542.37	144.50	1287.91	591.20	40	60	67	38
BRAZIL	$(0,1)$	3.60063×10^9	$3.59954 \times 10^{9\mathrm{s}}$	3.59934×10^9	3.56191×10^9	111.19	254.41	220.56	589.13	34	72	27	24
	$(1,0)$	5.21643×10^{10}	5.21790×10^{10}	5.21593×10^{10}	4.94520×10^{10}	585.80	240.19	1519.73	1329.28	38	66	66	67
	$(1,1)$	5.98963×10^{10}	5.98783×10^{10}	5.98747×10^{10}	5.75474×10^{10}	291.28	238.58	2120.56	1374.59	35	65	85	69
CASE9241	$(0,1)$	2.10540×10^8	$—^{\mathrm{u}}$	$—^{\mathrm{u}}$	$—^{\mathrm{e}}$	6911.56	—	—	—	39	—	—	—
	$(1,0)$	4.07654×10^{10}	$—^{\mathrm{u}}$	$—^{\mathrm{u}}$	$—^{\mathrm{e}}$	25946.10	—	—	—	52	—	—	—
	$(1,1)$	4.14859×10^{10}	$—^{\mathrm{u}}$	$—^{\mathrm{u}}$	$—^{\mathrm{e}}$	10053.20	—	—	—	49	—	—	—

The robustness gap widens when the number of scenarios increases to $N = 100$ (Table 6). At this scale, BlockIP again certifies optimality on all 21 instances, whereas every comparison solver loses some coverage. CPLEX certifies optimality on 10 of the 21 instances, terminates suboptimally on another 8, and returns no solution on the three CASE9241PEGASE cases; on IEEE30 its suboptimal terminations still report objective values that are orders of magnitude away from the certified optimum (for instance 2.57×10^6 against the optimal 1.34×10^5 under $(0,1)$). Gurobi solves 15 of the 21 instances and fails on the remaining 6: it declares all three configurations of IEEE30 infeasible and returns no solution (unknown status) on the three CASE9241PEGASE cases. HiPO-HiGHS solves 12 instances but fails on IEEE30, SSE810, and CASE9241PEGASE (one infeasible declaration and eight solver errors). Most importantly, all three comparison solvers fail on CASE9241PEGASE at $N = 100$, apparently because they run out of memory: CPLEX and Gurobi return no solution, and HiPO-HiGHS reports a solver error, so that BlockIP is the only solver able to solve this largest network with 100 scenarios, solving all three configurations with wall-clock times ranging from approximately 6912 s to 25946 s and 39 to 52 IPM iterations. This result underscores the scalability advantage of the specialized block-angular interior-point approach for large-scale stochastic problems.

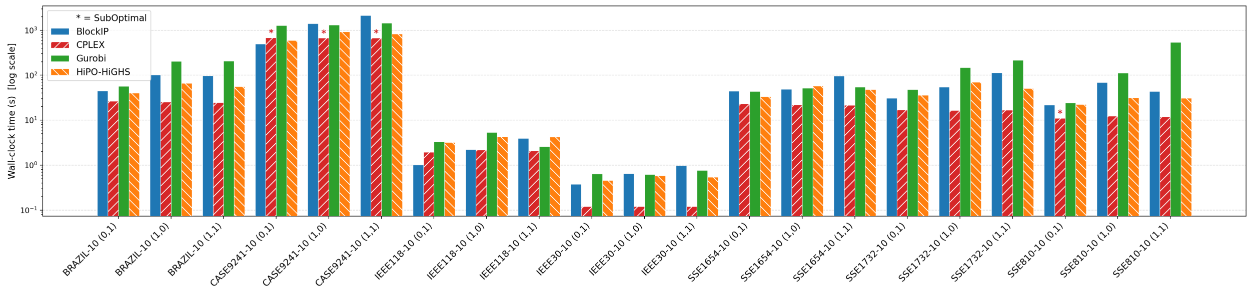
Table 7 provides an aggregate summary of the termination statuses for all solvers across the entire benchmark set (incorporating all networks, scenario sizes, and thread configurations). As the table shows, BlockIP produces a certified optimal solution for every single instance (84 out of 84), demonstrating complete robustness. The comparison solvers are more limited, and the gap widens with the number of scenarios. CPLEX certifies optimality on 53 of the 84 instances, terminating suboptimally on 28 of them (concentrated on IEEE30, the generation-cost configuration, and CASE9241PEGASE) and returning no solution on the three CASE9241PEGASE cases at $N = 100$. Gurobi has optimal count among the comparison solvers with 75 optimal instances; its remaining 9 non-optimal terminations are of three kinds: it terminates suboptimally on the three SSE810 loss and mixed configurations at $N = 10$ (higher thread counts), declares the three IEEE30 configurations infeasible at $N = 100$, and returns an unknown status on the three CASE9241PEGASE cases at $N = 100$. HiPO-HiGHS also reports an optimal status on 75 instances but is the most reliable at $N = 10$ (all 63 certified optimal); its 9 failures (one infeasible and eight solver errors) all occur at $N = 100$ and are confined to IEEE30, SSE810, and CASE9241PEGASE.

Table 7: Aggregate solution status across all benchmark instances, including all networks, scenario sizes ($N = 10, 100$), and objective configurations.

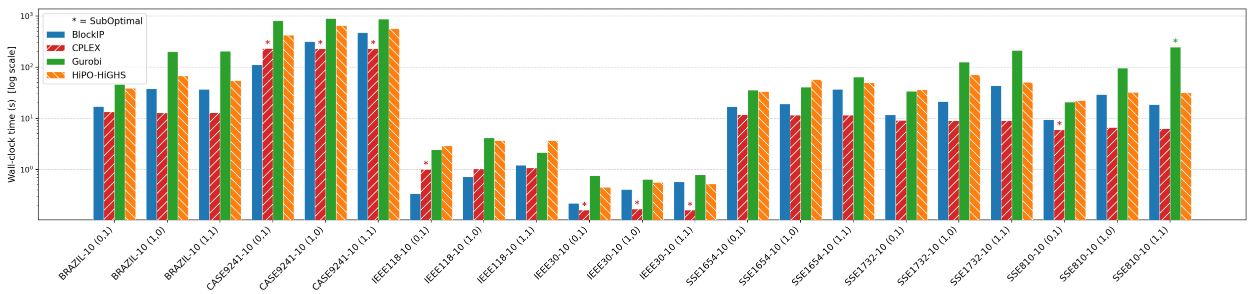
Solver	Optimal	SubOptimal	Infeasible	Unknown	Error
BlockIP	84	0	0	0	0
Gurobi	75	3	3	3	0
CPLEX	53	28	0	3	0
HiPO-HiGHS	75	0	1	0	8

Computational time and thread scaling

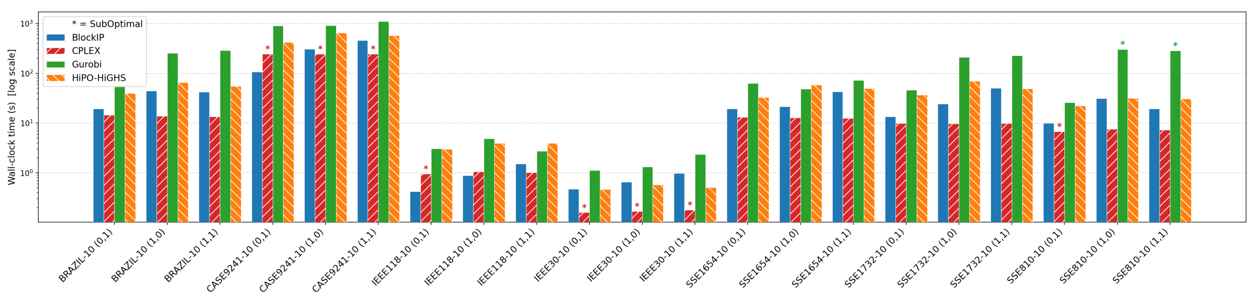
Figure 1 presents the wall-clock times on a logarithmic scale for all networks and solvers with $N = 10$ scenarios. Since no time limit is imposed, every reported time corresponds to the solver running to its own termination.



(a) $T_h = 1$ threads



(b) $T_h = 10$ threads

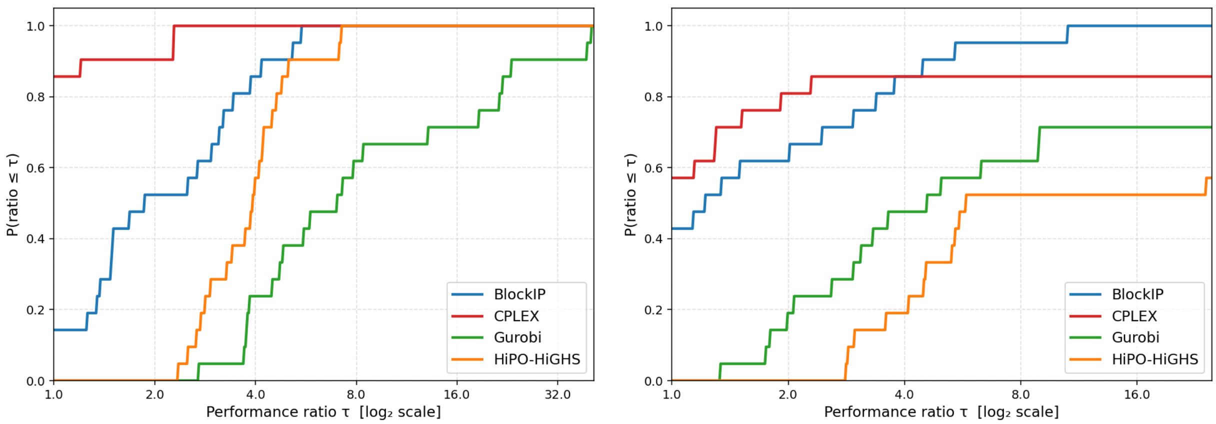


(c) $T_h = 22$ threads

Figure 1: Wall-clock time on logarithmic scale for $N = 10$ scenarios and all thread counts, shown for different penalty parameter combinations (α, β).

Across the small and medium networks (`IEEE30` through `SSE1732`), CPLEX most often records the shortest times. BlockIP is competitive on the small networks (`IEEE30`, `IEEE118`) and, once several threads are available, is even the fastest solver on some of them (for example `IEEE118` under $(0, 1)$ at $T_h = 22$, in 0.42 s against 0.96 s for CPLEX). On the medium networks under the loss and mixed configurations it is typically slower than CPLEX: at $T_h = 10$, for instance, it solves `SSE810` under $(1, 0)$ in 29.2 s versus 6.6 s for CPLEX, and at $T_h = 22$ it solves `SSE1732` under $(1, 1)$ in 50.7 s versus 9.9 s for CPLEX. This gap reflects the additional preconditioned conjugate-gradient work of the block-angular solver on these instances. Gurobi, however, is no longer competitive on these networks: on the same instances it takes 96.8 s and 228.4 s respectively, far slower than both CPLEX and BlockIP, and on the `SSE810` loss and mixed configurations its barrier either requires hundreds of iterations or fails to converge altogether.

On the largest network, `CASE9241PEGASE`, at $N = 10$ all four solvers return a value, and BlockIP is fully competitive. At $T_h = 22$ it is the fastest solver on the generation-cost configuration $(0, 1)$, finishing in about 107 s against 245 s for CPLEX, 902 s for Gurobi, and 422 s for HiPO-HiGHS; on $(1, 0)$ it finishes in about 308 s and on $(1, 1)$ in about 458 s, in both cases far faster than Gurobi (921 s and 1105 s) and HiPO-HiGHS (655 s and 577 s). Crucially, CPLEX terminates suboptimally on all three `CASE9241PEGASE` configurations at every thread count, and on $(1, 0)$ and $(1, 1)$ its reported objective is off by three orders of magnitude, so its shorter times (246 s) do not correspond to usable solutions. BlockIP, by contrast, certifies an optimal solution in every configuration and is therefore the fastest solver that actually



(a) $N = 10$ scenarios, $T_h = 22$ threads

(b) $N = 100$ scenarios, $T_h = 22$ threads

Figure 2: Performance profiles (Dolan–Moré) for $N = 10$ and $N = 100$ scenarios test.

solves this network. The decisive scalability advantage of the block-angular decomposition then emerges at $N = 100$, where BlockIP is the only solver capable of producing a solution at all.

On the medium and large networks at $N = 10$, Gurobi is the slowest of the four solvers on the majority of instances; for example, on SSE1654 with $(\alpha, \beta) = (0, 1)$ at $T_h = 22$ it requires approximately 63 s, compared with about 13 s for CPLEX, 20 s for BlockIP, and 33 s for HiPO-HiGHS.

Performance profile

To provide an aggregate assessment of both efficiency and robustness across the benchmark set, Figure 2 presents the Dolan–Moré performance profiles [12] for $N = 10$ and $N = 100$ scenarios, both at $T_h = 22$ threads. The profiles plot the fraction of instances solved within a factor τ of the best-performing wall-clock time of the solver. The y-intercept ($\tau = 1$) indicates the probability that a solver is the fastest on a given instance, while the asymptotic value as $\tau \to \infty$ represents the overall reliability of the solver.

Because these profiles are built purely on wall-clock time and count any returned solution as a success, they must be read together with the solution-quality caveats above: a solver that stops early at a suboptimal or numerically inaccurate point still appears "fast" here. The $N = 10$ profile (Figure 2, left panel) shows all four solvers eventually reaching $P(\tau) = 1$, since each returns a usable value on all 21 instances at this scale (Gurobi's two SSE810 loss and mixed runs at $T_h = 22$ terminate suboptimally rather than failing, and so still count as solved); the solvers therefore differ not in coverage but in how quickly they climb. CPLEX has the highest baseline efficiency, being the fastest solver on about 86% of the instances at $\tau = 1$ and reaching $P(\tau) = 1$ almost immediately (by $\tau \approx 2.3$), although part of this apparent dominance comes from its early suboptimal terminations on IEEE30, on the generation-cost configurations, and on all of CASE9241PEGASE. BlockIP is the fastest solver on roughly 14% of the instances at $\tau = 1$ (the generation-cost and loss configurations of IEEE118 and the generation-cost configuration of CASE9241PEGASE) and reaches $P(\tau) = 1$ by $\tau \approx 5.5$. HiPO-HiGHS is never the fastest, remaining at $P(\tau) = 0$ for small τ, but it climbs steadily and reaches full coverage by $\tau \approx 7$. Gurobi is never the fastest solver and, although it too eventually reaches $P(\tau) = 1$, it does so only at $\tau \approx 40$, because on its worst instances (the SSE810 loss and mixed configurations, where its barrier now stops at a suboptimal point after hundreds of iterations) it is up to about 40 times slower than the best solver.

At $N = 100$ (Figure 2, right panel), BlockIP is the only solver whose profile reaches $P(\tau) = 1$, since it is the only one that returns a usable solution on all 21 instances, and it is the fastest solver on about 43% of them at $\tau = 1$ (including the three CASE9241PEGASE cases, where it is the sole solver). CPLEX has the highest $\tau = 1$ intercept (fastest on about 57% of the instances) but plateaus near $P(\tau) \approx 0.86$, reflecting the three CASE9241PEGASE cases on which it returns no solution, while Gurobi plateaus near 0.71 and HiPO-HiGHS near 0.57 because of the $N = 100$ instances (all of IEEE30 and CASE9241PEGASE for both, plus all of SSE810 for HiPO-HiGHS) on which they return no usable solution.

Large-scale test: 100 scenarios

The transition from $N = 10$ to $N = 100$ scenarios increases the problem dimension by approximately a factor of 10. Figure 3 presents the wall-clock times for this configuration at $T_h = 22$ threads. As no time limit is imposed, each solver runs to its own termination.

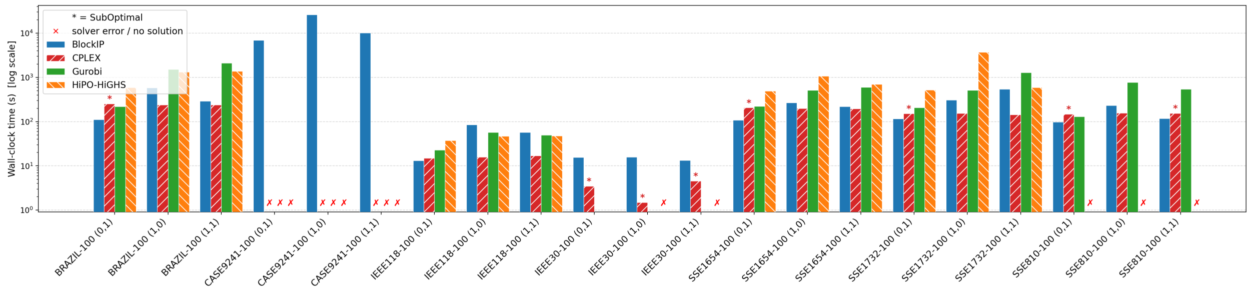


Figure 3: Wall-clock time on logarithmic scale for $N = 100$ scenarios and 22 threads, shown for different penalty parameter combinations (α, β).

For the small networks, every solver that returns a solution finishes quickly, because the constraint matrices still remain of a reasonable size. On `IEEE30`, only BlockIP and CPLEX return a value: BlockIP certifies optimality in about 13–16 seconds across the three configurations, and CPLEX terminates suboptimally in about 1.5–4.6 seconds with objective values that are far from optimal, while Gurobi declares the instance infeasible on all three configurations and HiPO-HiGHS either declares it infeasible ($(0,1)$) or fails with a solver error ($(1,0)$ and $(1,1)$). On `IEEE118` all four solvers succeed: CPLEX remains the most efficient (about 15–17 seconds), Gurobi takes about 23–57 seconds, BlockIP ranges from roughly 13 seconds for $(\alpha, \beta) = (0, 1)$ to 86 seconds for $(1, 0)$, and HiPO-HiGHS runs at about 37–48 seconds.

Across the medium networks (`SSE810` through `SSE1732`) the behavior depends strongly on the objective structure and on which solvers actually certify optimality. On `SSE810`, HiPO-HiGHS fails with a solver error on all three configurations; of the remaining three, BlockIP certifies optimality in about 98–233 seconds, Gurobi certifies optimality on all three in about 130–773 seconds, and CPLEX solves the loss configuration $(1, 0)$ to optimality while terminating suboptimally, but with accurate objective values, on $(0, 1)$ and $(1, 1)$, all in about 148–155 seconds. On `SSE1654` and `SSE1732` all four solvers return times: CPLEX records the shortest (roughly 144–206 seconds) but terminates suboptimally on the generation-cost configurations; BlockIP certifies optimality on all of them at a competitive but more variable cost, from about 108 seconds on `SSE1654` under $(0, 1)$ to about 542 seconds on `SSE1732` under $(1, 1)$; Gurobi is markedly slower, ranging from about 209 seconds up to about 1288 seconds on `SSE1732` under $(1, 1)$. HiPO-HiGHS is the least predictable here: it runs in about 490–1070 seconds on `SSE1654`, but on `SSE1732` under $(1, 0)$ it needs roughly 3718 seconds and 240 barrier iterations, by far the largest single running time in the benchmark.

On `BRAZIL`, BlockIP certifies optimality in all three configurations, with wall-clock times of about 111 seconds for $(0, 1)$, 586 seconds for $(1, 0)$, and 291 seconds for $(1, 1)$. CPLEX is the fastest here, at about 239–254 seconds, but terminates suboptimally on the generation-cost configuration $(0, 1)$; Gurobi solves all three configurations but is the slowest, at about 221, 1520, and 2121 seconds, and HiPO-HiGHS takes roughly 589–1375 seconds.

The most notable result occurs on the `CASE9241PEGASE` network, the largest system with $N = 100$ scenarios (about 74 million variables and 36 million constraints): BlockIP is the only solver that produces a solution, requiring approximately 6912 seconds for $(\alpha, \beta) = (0, 1)$, 25946 seconds for $(1, 0)$, and 10053 seconds for $(1, 1)$. All three comparison solvers fail on this network under every objective configuration: CPLEX and Gurobi return no solution (unknown status), and HiPO-HiGHS terminates with a solver error. All these failed runs are likely due to memory exhaustion. This demonstrates that the specialized block-angular decomposition enables the solution of stochastic predispatch problems at scales that are completely intractable for general-purpose solvers on a moderate hardware.

In summary, the 100-scenario results show that BlockIP trades per-instance speed for robustness and reach. On the medium and large networks, BlockIP is competitive on the generation-cost configuration $(0, 1)$ (it is, for instance, faster than both Gurobi and HiPO-HiGHS on `SSE1654` and `BRAZIL` under $(0, 1)$), whereas on the loss and mixed configurations CPLEX is usually faster when it succeeds, and Gurobi and HiPO-HiGHS are frequently slower than BlockIP. Its decisive advantage is reliability in hardware with limited memory resources: it is the only solver that certifies optimality on every medium and large instance, and the only one able to solve `CASE9241PEGASE` at $N = 100$ at all, where CPLEX, Gurobi, and HiPO-HiGHS all fail. The wide variation in its wall-clock times across configurations reflects the sensitivity of the block-angular preconditioner to the conditioning of the objective; the dependence of PCG convergence on the spectral radius of the preconditioned system is analyzed in [10].

IPM iteration counts

Figure 4 reports the interior-point iteration counts for the $N = 100$ instances. Since BlockIP uses PCG to solve the Newton system, unlike the other solvers that rely on direct factorizations, its outer iteration count is more sensitive to problem conditioning. Nevertheless, on this benchmark, its iteration counts are comparable to, and often lower than, those of the other solvers. This can be explained by the loose optimality tolerance; a tighter tolerance would probably affect BlockIP substantially more than the other solvers. Across the $N = 100$ networks its counts range from about 7 to 52: they are very low on `IEEE30` (7), moderate on `SSE810`, the `SSE1654`/`SSE1732`, and `BRAZIL` systems (34–49), and largest on `CASE9241PEGASE` (39–52). CPLEX uses a consistently higher 52–76 iterations on the networks where it converges (`IEEE118`, `SSE810`, `SSE1654`, `SSE1732`, `BRAZIL`), and it has much higher counts on `IEEE30` (51–178) that occur precisely on the instances where its barrier fails to converge and terminates suboptimally, so they reflect non-convergence rather than efficient solving. Gurobi, where it returns a solution, uses about 27–34 iterations on most networks but spikes to 66–100 on the loss configurations of `SSE810` and `BRAZIL`; HiPO-HiGHS uses about 22–30 iterations in most cases, with a single extreme outlier of 240 iterations on `SSE1732` under $(1, 0)$, consistent with its long wall-clock time there. On the instances where all solvers certify optimality, the iteration counts of BlockIP are therefore comparable to or below those of CPLEX and within a small factor of Gurobi and HiPO-HiGHS, although HiPO-HiGHS pays a much higher wall-clock cost per iteration. The missing bars on `CASE9241PEGASE` in Figure 4, together with the missing Gurobi and HiPO-HiGHS bars on `IEEE30` and the missing HiPO-HiGHS bars on `SSE810`, correspond to the unknown, infeasible, and error terminations discussed above.

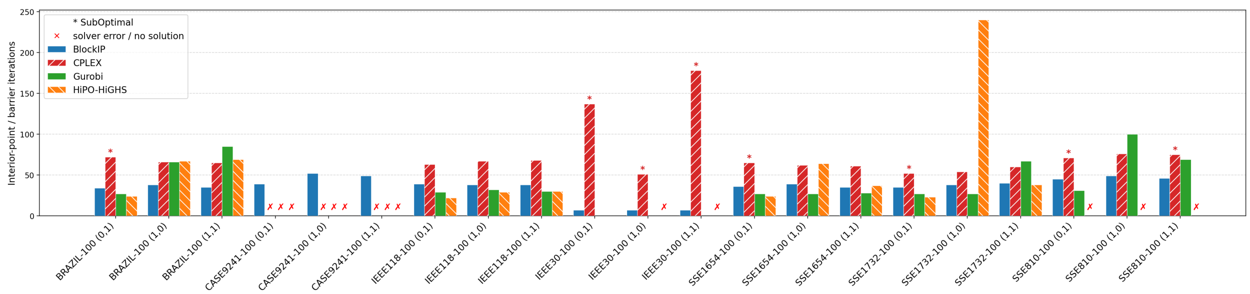


Figure 4: Number of interior-point iterations required by solvers for the $N = 100$ scenario instances at 22 threads. Missing bars indicate failures.

5.5 Case study: `CASE9241PEGASE`

To provide deeper insight into the stochastic model behavior, we examine the largest system, `CASE9241PEGASE`, in detail based on the BlockIP results. This network comprises 9,241 buses and represents an approximation of the European high-voltage transmission grid, making it a demanding stress test for any predispatch solver.

Demand scenarios

Figure 5 shows the ten demand scenarios over the 24-hour planning horizon. The eight historical scenarios (drawn via Monte Carlo sampling from the ONS dataset, each with probability 12.25%) are displayed as light-blue curves. Scenarios 1 and 2 (probability 1% each, i.e. 2% in total) represent a $\pm 20\%$ perturbation relative to the mean and are shown as dashed lines. The spread between scenarios motivates the stochastic formulation: a deterministic model based on mean demand would ignore the tails of this distribution, potentially leading to under-scheduled thermal reserves during high-demand realizations.

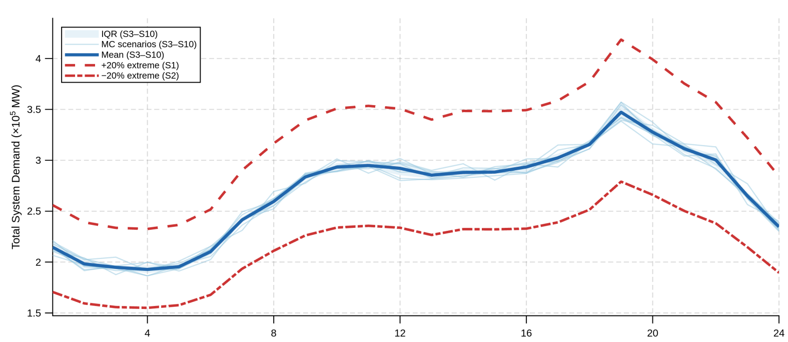
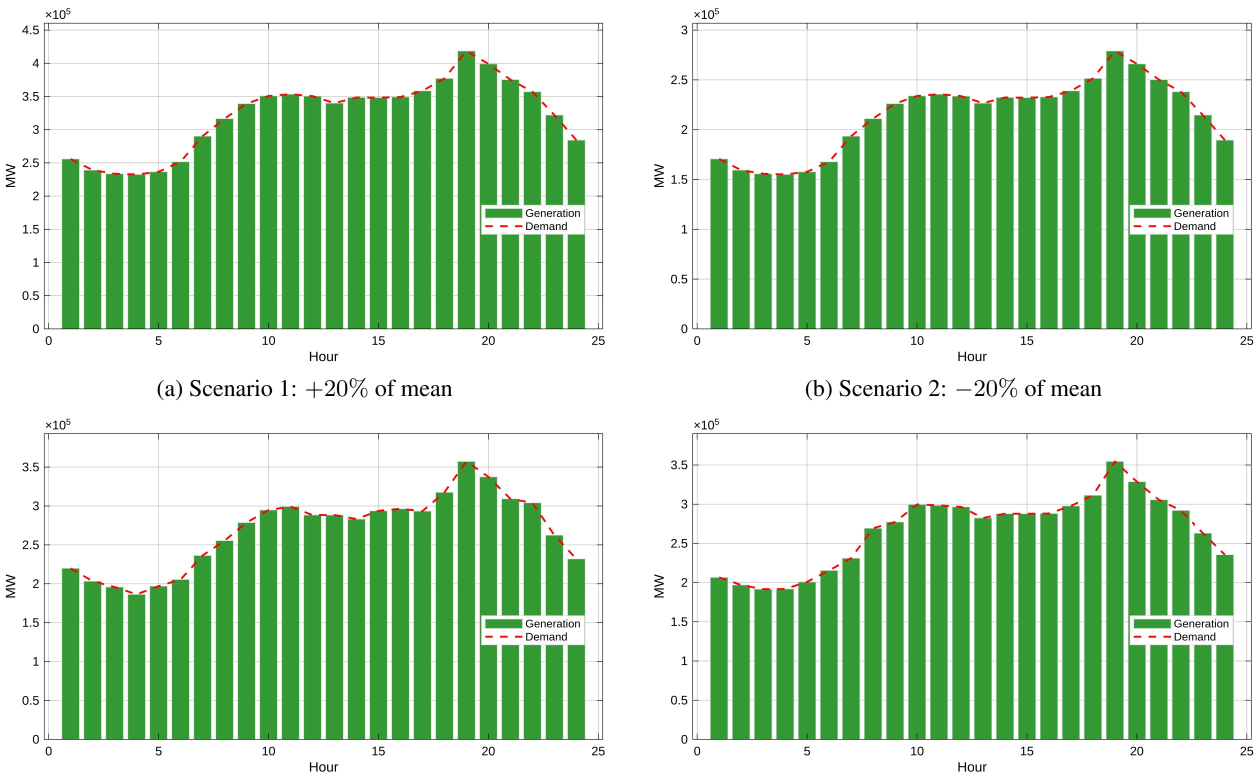


Figure 5: Demand scenarios for `CASE9241PEGASE` over the 24-hour planning horizon. Blue curves: eight historical Monte Carlo scenarios, where dark blue is the mean. Red dashed lines: extreme scenarios at $\pm 20\%$ of the mean.

Supply–demand balance across scenarios

Figure 6 shows the total hourly generation (green bars) against the demand (red dashed line) for the first four scenarios: the two extreme ones and two generated with Monte-Carlo. In every scenario and every hour, the bars track the demand curve without visible slack, confirming that the nodal balance constraint of total generation and demand is satisfied by the BlockIP interior-point solution. The four panels also illustrate the magnitude of inter-scenario variability absorbed by the stochastic model: peak demand ranges from roughly 2.8×10^5 MW in scenario 2 to about 4.2×10^5 MW in scenario 1, a spread of about 50% relative to the lower extreme. All four scenarios share the same characteristic daily profile: an overnight trough, a midday plateau, and a pronounced evening peak. So the inter-scenario variability is primarily one of overall level rather than shape, with the extreme scenarios 1 and 2 shifted uniformly up and down relative to the Monte Carlo samples 3 and 4.

It is also observed that all scenarios follow the same pattern, in which the lowest demand occurs during the early morning hours, around 4:00, while the highest peak is observed at approximately 19:00. Additionally, there is a local peak around 11:00, followed by a slight decrease, after which demand continues to increase until reaching its maximum at 19:00.



(a) Scenario 1: $+20\%$ of mean

(b) Scenario 2: -20% of mean

(c) Scenario 3: Monte Carlo

(d) Scenario 4: Monte Carlo

Figure 6: Total generation versus demand for four demand scenarios for four representative scenarios of `CASE9241PEGASE`. Scenario 1 corresponds to a $+20\%$ deviation from the mean, Scenario 2 to a -20% deviation, and Scenarios 3 and 4 are generated via Monte Carlo sampling.

Stage-1 hydro commitment vs. stage-2 thermal resource

To expose the two-stage structure of the optimal dispatch, Figure 7 decomposes the solution for the scenario 1 corresponding to the $+20\%$ high-demand extreme into its hydro and thermal components. The blue area is the hydro generation fixed in the first stage; being a here-and-now decision, it is identical across all N scenarios by construction. The red area is the second-stage *thermal recourse for scenario 1*: the dispatch that reacts to the realized demand of the scenario, shown by the dashed curve (also scenario 1). It is therefore a single-scenario snapshot, not an average over scenarios.

Hydro provides the base supply throughout the horizon and follows the general morphology of the load curve: it dips during the overnight valley (hours 1–5, around 1.3×10^5 MW), rises through the morning, and peaks at roughly 2.4×10^5 MW at hour 19, coincident with the daily demand peak. Because it is committed in the first stage, this shape reflects the here-and-now decision taken in expectation, not a reaction to scenario 1 in particular. Thermal output then tracks the residual demand and grows from about 1.1×10^5 MW overnight to nearly 1.9×10^5 MW during the evening peak (hours 18–20), when the stage-1 hydro commitment alone is insufficient to cover the load. Rather than hydro being capped at a physical limit, the split reflects an economic allocation: the first stage commits as much hydro as is optimal in expectation, and the thermal resource absorbs the remainder once the realized demand is observed. This division of labor is precisely what the two-stage formulation is designed to exploit: renewable hydro capacity is committed here-and-now at low cost, while thermal units are held as a recourse against scenario-dependent demand variations.

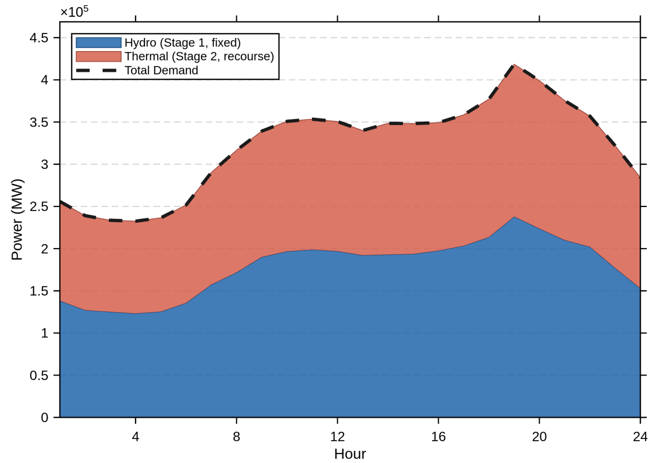


Figure 7: Generation mix in scenario 1 corresponding to the $+20\%$ high-demand extreme: stage-1 hydro in blue (fixed across scenarios) vs. stage-2 thermal resource in red. The dashed curve is the total hourly demand.

5.5.1 EVPI and VSS solution

We evaluated the stochastic information value on the `CASE9241PEGASE` network with $N = 10$ demand scenarios, for the three objective configurations that isolate the two terms of the cost function in (3): pure generation cost ($\alpha = 0,\ \beta = 1$), the combined cost-and-loss objective ($\alpha = 1,\ \beta = 1$), and pure transmission losses ($\alpha = 1,\ \beta = 0$). For each configuration, we solved the Wait-and-See problem scenario by scenario, the Recourse Problem in its extensive form, and the Expected Value problem, the latter evaluated across all scenarios to obtain the EEV. Table 8 reports the resulting objective values and the absolute VSS and EVPI, while Table 9 reports the two information-value measures under two normalizations.

Remark 4. Throughout this section, the comparison between the Expected Value of the Expected Value (EEV) and the stochastic solution relies on the recourse problem (RP) of BlockIP. Since the solver operates on the transformed model of (3), in which every variable admits a zero lower bound ($p_{\min} = 0$ and $f_{\min} = -f_{\max}$), its raw output accounts only for the variable part of the objective. The constant term induced by this change of variables is therefore restored in all reported values, which is required for a correct evaluation of the EEV metric and a consistent comparison across the WS, RP, and EEV solutions.

Table 8: Stochastic information value for `CASE9241PEGASE` with $N = 10$ scenarios for the three objective configurations.

α	β	**WS**	**RP**	**EEV**	**VSS**	**EVPI**	**EEV − WS**
0	1	3.2047×10^{8}	3.2068×10^{8}	3.2069×10^{8}	9.108×10^{3}	2.104×10^{5}	2.196×10^{5}
1	1	3.4593×10^{10}	3.5378×10^{10}	3.5383×10^{10}	5.023×10^{6}	7.857×10^{8}	7.907×10^{8}
1	0	3.3764×10^{10}	3.4558×10^{10}	3.4564×10^{10}	6.214×10^{6}	7.938×10^{8}	8.000×10^{8}

The results satisfy the theoretical ordering $WS \leq RP \leq EEV$ established in Section 4 in every configuration. Since the transformation constant is added to WS, RP, and EEV, it cancels out in $VSS = EEV - RP$, $EVPI = RP - WS$, and their sum $EEV - WS$. As a result, these three quantities do not change under the variable transformation, while RP still includes the constant and is dominated by the fixed cost of meeting demand. This explains the difference between the two normalizations in Table 9. Normalizing by RP makes both gaps appear smaller because they are compared with the total generation cost, whereas normalizing by the $EEV - WS$ puts VSS and $EVPI$ on the same transformation-invariant scale and divides this gap into the part recovered by the stochastic model (VSS) and the part due to imperfect information ($EVPI$).

Table 9: VSS and EVPI under two normalizations and the EVPI-to-VSS ratio.

		Normalized by RP		Normalized by $EEV - WS$		
α	β	**VSS** [%]	**EVPI** [%]	**VSS** [%]	**EVPI** [%]	**EVPI/VSS**
0	1	0.0028	0.0656	4.15	95.85	23.1
1	1	0.0142	2.2209	0.64	99.36	156.4
1	0	0.0180	2.2971	0.78	99.22	127.7

In all three configurations, the value of the stochastic solution is negligible: the VSS never exceeds 0.018% of RP, so the deterministic mean-demand schedule is already near-optimal in expectation, since the flexible thermal resource absorbs the scenario deviations at additional cost, which is a direct consequence of the relatively complete recourse of the formulation. Correspondingly, when the gap $EEV - WS$ is split into its two parts, the EVPI accounts for the large majority of it and the VSS for only a few percent, so the EVPI exceeds the VSS by roughly one to two orders of magnitude (a factor of about 23 for $(\alpha = 0, \beta = 1)$, 156 for $(\alpha = 1, \beta = 1)$, and 128 for $(\alpha = 1, \beta = 0)$): almost none of the cost difference between the mean-value schedule and a perfect forecast is recovered by adopting the stochastic model, while the rest would require better demand information. The relative EVPI is only 0.07% of RP for the pure generation-cost configuration $(\alpha = 0, \beta = 1)$, but rises to about 2.2–2.3% of RP once the transmission-loss term is present, because losses depend mainly on the spatial distribution of the flows, which is more sensitive to the exact demand realization than generation-cost minimization.

Overall, the cost of ignoring uncertainty is modest, while the EVPI, reaching about 2.3% of RP when transmission losses are penalized, dominates the gap and provides an upper bound on the economic value of improved forecasting. These magnitudes are specific to the assumed scenario set and the ample thermal resource of `CASE9241PEGASE`.

6 Conclusions

This paper addressed the predispatch optimal power flow problem under demand uncertainty. The proposed approach combines a DC network flow formulation that explicitly models Kirchhoff laws and transmission limits, a two-stage stochastic framework that separates first-stage hydroelectric commitment from second-stage thermal recourse, and a variable-splitting reformulation that turns the extensive-form problem into a block-angular structure amenable to a specialized interior-point method. This structure preserves the sparse network representation and yields a block-diagonal normal-equation system whose scenario blocks factorize independently. Across 84 benchmark instances spanning seven networks from 30 to 9,241 buses with 10 and 100 scenarios, BlockIP certified an optimal solution on every case, whereas CPLEX, Gurobi, and HiPO-HiGHS lost coverage as network size and scenario count grew, and on the `CASE9241PEGASE` network with 100 scenarios, a problem with millions of variables, BlockIP was the only solver to return a feasible solution on the modest hardware used. The method is also competitive on running time, matching or outperforming the commercial barrier solvers on the larger instances at 100 scenarios while keeping comparable or lower iteration counts.

Looking ahead, several extensions would strengthen both the modeling fidelity and the computational reach of this framework. Incorporating security constraints and scheduled line manipulations into the stochastic formulation would bring the model closer to real operational practice, building on the deterministic interior-point formulations developed for these settings [4, 5]. Extending the horizon to a multi-stage stochastic optimization problem, in which reservoir dynamics and inter-day water management are modeled explicitly, would address medium-term planning following the methodology of [10]. Finally, introducing wind and solar generation as additional uncertain parameters would extend the framework to modern power systems with variable renewable sources.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil (CNPq). J. Castro was supported by the MCIN/AEI/10.13039/501100011033/FEDER,EU grants PID2022-139219OB-I00 and PID2025-169807OB-I00.

Data availability

The MPS files for all the instances used in this study are available from the authors upon request.

References

[1] Birge, J.R., Louveaux, F.: Introduction to stochastic programming. Springer, New York, NY (2011). DOI 10.1007/978-1-4614-0237-4

[2] Capitanescu, F., Glavic, M., Ernst, D., Wehenkel, L.: Interior-point based algorithms for the solution of optimal power flow problems. Electric Power Systems Research **77**(5), 508–517 (2007). DOI 10.1016/j.epsr.2006.05.003

[3] Carvalho, M., Soares, S., Ohishi, T.: Optimal active power dispatch by network flow approach. IEEE Transactions on Power Systems **3**(4), 1640–1647 (1988). DOI 10.1109/59.192975

[4] Carvalho, S.M., Oliveira, A.R., Coelho, M.V.: Predispatch linear system solution with preconditioned iterative methods. Journal of Control, Automation and Electrical Systems **32**, 145–152 (2021). DOI 10.1007/s40313-020-00659-9

[5] Casacio, L., Lyra, C., Oliveira, A.R.: Interior point methods for power flow optimization with security constraints. International Transactions in Operational Research **26**(1), 364–378 (2019). DOI 10.1111/itor.12279

[6] Castro, J.: A specialized interior-point algorithm for multicommodity network flows. SIAM Journal on Optimization **10**(3), 852–877 (2000). DOI 10.1137/S1052623498341879

[7] Castro, J.: Solving difficult multicommodity problems with a specialized interior-point algorithm. Annals of Operations Research **124**, 35–48 (2003). DOI 10.1023/B:ANOR.0000004761.99649.a5

[8] Castro, J.: An interior-point approach for primal block-angular problems. Computational Optimization and Applications **36**(2), 195–219 (2007). DOI 10.1007/s10589-006-9000-1

[9] Castro, J., Cuesta, J.: Quadratic regularizations in an interior-point method for primal block-angular problems. Mathematical Programming **130**, 415–445 (2011). DOI 10.1007/s10107-010-0341-2

[10] Castro, J., Escudero, L.F., Monge, J.F.: On solving large-scale multistage stochastic optimization problems with a new specialized interior-point approach. European Journal of Operational Research **310**(1), 268–285 (2023). DOI 10.1016/j.ejor.2023.03.042

[11] Cipolla, S., Gondzio, J.: Proximal stabilized interior point methods and low-frequency-update preconditioning techniques. Journal of Optimization Theory and Applications **197**, 1061–1103 (2023). DOI 10.1007/s10957-023-02194-4

[12] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**, 201–213 (2002). DOI 10.1007/s101070100263

[13] Fan, M., Vittal, V.: Mathematical Models and Algorithms for Power System Optimization. Academic Press (2019)

[14] Garcés, A.: Mathematical Programming for Power Systems Operation: From Theory to Applications in Python. John Wiley & Sons (2021). DOI 10.1002/9781119747291

[15] Gondzio, J.: Interior point methods 25 years later. European Journal of Operational Research **218**(3), 587–601 (2012). DOI 10.1016/j.ejor.2011.09.017

[16] Gondzio, J., Grothey, A.: Solving non-linear portfolio optimization problems with the primal-dual interior point method. European Journal of Operational Research **181**(3), 1019–1029 (2007). DOI 10.1016/j.ejor.2006.03.006

[17] Lustig, I.J., Mulvey, J.M., Carpenter, T.J.: Formulating two-stage stochastic programs for interior point methods. Operations Research **39**(5), 757–770 (1991). DOI 10.1287/opre.39.5.757

[18] Momoh, J., Adapa, R., El-Hawary, M.: A review of selected optimal power flow literature to 1993. i. nonlinear and quadratic programming approaches. IEEE Transactions on Power Systems **14**(1), 96–104 (1999). DOI 10.1109/59.744492

[19] Ohishi, T., Soares, S., Carvalho, M.: A short-term hydrothermal scheduling approach for dominantly hydro systems. IEEE Transactions on Power Systems **6**(2), 637–643 (1991). DOI 10.1109/59.76707

[20] Oliveira, A.R., Soares, S., Nepomuceno, L.: Optimal active power dispatch combining network flow and interior point approaches. IEEE Transactions on Power Systems **18**(4), 1235–1240 (2003). DOI 10.1109/TPWRS.2003.814851

[21] Oliveira, A.R., Soares, S., Nepomuceno, L.: Short term hydroelectric scheduling combining network flow and interior point approaches. International Journal of Electrical Power & Energy Systems **27**(2), 91–99 (2005). DOI 10.1016/j.ijepes.2004.07.009

[22] Pereira, M., Pinto, L.: Multi-stage stochastic optimization applied to energy planning. Mathematical Programming **52**, 359–375 (1991). DOI 10.1007/BF01582895

[23] Probst, R.W., Oliveira, A.R.L.: A new predictor–corrector method for optimal power flow. Optimization and Engineering **16**(2), 335–346 (2015). DOI 10.1007/s11081-014-9265-7

[24] de Queiroz, A.R.: Stochastic hydro-thermal scheduling optimization: An overview. Renewable and Sustainable Energy Reviews **62**, 382–395 (2016). DOI 10.1016/j.rser.2016.04.065

[25] Quintana, V.H., Torres, G.L., Medina-Palomo, J.: Interior-point methods and their applications to power systems: a classification of publications and software codes. IEEE Transactions on Power Systems **15**(1), 170–176 (2000). DOI 10.1109/59.852117

[26] de Sistema Elétrico, O.N.: Curva de carga horária. https://www.ons.org.br/paginas/resultados-da-operacao/historico-da-operacao/dados-gerais

[27] Soares, S., Salmazo, C.: Minimum loss predispatch model for hydroelectric systems. IEEE Transactions on Power Systems **12**(3), 1220–1228 (1997). DOI 10.1109/59.630464

[28] Vanderbei, R.J.: Linear Programming: Foundations and Extensions, *International Series in Operations Research & Management Science*, vol. 285, 5 edn. Springer, Cham, Switzerland (2020). DOI 10.1007/978-3-030-39415-8

[29] Wright, S.J.: Primal-dual interior-point methods. SIAM, Philadelphia, PA (1997). DOI 10.1137/1.9781611971453

[30] Zhu, J.: Optimization of Power System Operation, 2 edn. IEEE Press Series on Power Engineering. John Wiley & Sons, Hoboken, NJ (2015). DOI 10.1002/9781118887004