GTOpt: Open-Source Energy-System Optimization, Consistent from Minutes to Decades

Marcelo Matus A., *Member, IEEE,* María del Pilar Buitrago Villada, Erick Sierra Baeza, Carlos Enrique García Bujanda, Catalina López, Marcos Falcone Salazar, and Gonzalo Bustos Turu

***Abstract*—Planning and operating energy systems with high shares of variable renewables, storage and hydro requires models that remain consistent from daily dispatch to multi-decade expansion. This paper presents GTOpt, an open-source optimizer written in C++26 for the expansion planning of generation, transmission, storage and demand, hydrothermal scheduling, and unit commitment and dispatch. The same formulation is used at every time scale, from 15-minute blocks to 30-year horizons. It covers DC power flow with Kirchhoff constraints and line losses, hydro cascades with water rights, batteries, unit commitment, reserves and inertia, multi-carrier chains (fuels, LNG, hydrogen, ammonia, heat) and multi-pollutant emission zones with caps and allowance banking. The time scales are coupled through optimality cuts, which carry water values and storage opportunity costs, and through feasibility cuts, which enforce reservoir and water-right limits. Three methods are provided: a monolithic LP/MIP; stochastic dual dynamic programming (SDDP) with backward-pass apertures, multi-cut and elastic-filter feasibility cuts; and a multi-level cascade that refines a coarse model into a detailed one while inheriting its cuts. Operating rules such as the Chilean irrigation agreements are added without code through PAMPL, an AMPL-like constraint language, and LP/MIP solvers are loaded at run time as plugins. GTOpt is validated against IEEE benchmark networks, pandapower and published SDDP benchmarks. It is also compared with the official daily schedule of the Chilean national system on 14 days, using the system operator's PLEXOS inputs and results. The comparison shows that the line-loss model drives most of the differences: a piecewise-linear loss model with free losses inflates them to 2.5 times their physical value, whereas a tangent outer approximation with a small loss price stays within 5%.**

***Index Terms*—Energy system optimization, generation and transmission expansion planning, hydrothermal scheduling, stochastic dual dynamic programming, multi-scale consistency, DC optimal power flow, line losses, energy storage, water values, irrigation agreements, emissions, open-source software.**

Preprint, November 2026. Source code, data and documentation: https://github.com/gtopt-community/gtopt.

M. Matus A. is with Black Bear Engineering SpA, Santiago, Chile (https://www.blackbeareng.cl), and is a Fellow of Centro Enlace, Santiago, Chile (https://centroenlace.org/) (e-mail: mmatus@blackbeareng.cl).

M. P. Buitrago Villada is with the Department of Electrical Energy and Automation, Universidad Nacional de Colombia, Medellín, Colombia.

E. Sierra Baeza is with Black Bear Engineering SpA, Santiago, Chile, and CENIA.

C. E. García Bujanda is with Black Bear Engineering SpA, Santiago, Chile.

C. López is with Black Bear Engineering SpA, Santiago, Chile (e-mail: clopez@blackbeareng.cl).

M. Falcone Salazar is with the School of Engineering, Universidad Adolfo Ibáñez, Santiago, Chile, and formerly with Colbún S.A., Santiago, Chile.

G. Bustos Turu is with the Department of Chemical Engineering, Imperial College London, London, UK, and the Centre for Energy, Universidad de Chile, Santiago, Chile.

I. Introduction

The transition toward low-carbon energy systems requires planning tools that co-optimize generation, transmission and storage investments while capturing the operational detail of variable renewables, storage and hydro cascades [1]. Generation and Transmission Expansion Planning (GTEP) determines the minimum-cost combination of investments (CAPEX) and operating decisions (OPEX) over long horizons under uncertainty [2], [3]. In hydro-dominated systems the same questions extend to the medium term (months to years), where reservoir management under inflow uncertainty is solved with stochastic dual dynamic programming (SDDP) [4], and to daily operation, where unit commitment and network constraints set the dispatch and the marginal prices.

In practice these three horizons are usually served by different models, with different data formats, element definitions and simplifications. A long-term expansion model, a medium-term hydrothermal model and a short-term scheduling model can therefore disagree on the value of water, the opportunity cost of storage or the feasibility of operating rules, and the policy handed from one horizon to the next is only approximately consistent. The commercial tools used by system operators and utilities (e.g. PLEXOS, PSR SDDP, and in Chile the PLP model maintained by the system operator) are closed or tied to specific data formats, which limits the auditability of their results. Open-source tools such as PyPSA [5], GenX [6], IDAES-GTEP [7] and SDDP.jl [8] have broadened access to planning and stochastic optimization. None of them, however, combines in one model a detailed hydro and water-rights representation, a network with losses, SDDP and capacity expansion.

This paper presents GTOpt, an open-source optimizer for planning and operating energy systems, written in C++26. GTOpt descends from the FESOP (Fabulous Energy System Optimizer) framework [9], which extended long-term hydrothermal scheduling [10] with capacity expansion, variable renewables and modern flexibility mechanisms. Its central design decision is to use *one* formulation of the energy system—electricity, water, fuels and LNG, hydrogen, ammonia and heat—at every time scale, and to change only the time structure and the solution method between scales.

The main contributions of this work are:

1) A complete LP/MIP formulation of the energy system (Section III). It includes:
   - DC optimal power flow with Kirchhoff constraints, transmission switching, and several line-loss models

with a guard against loss arbitrage under negative prices;
- batteries, and hydro cascades with reservoirs, turbines, pumps, seepage and head-dependent efficiency;
- a generic water-rights layer for irrigation agreements;
- unit commitment (three-binary formulation) with piecewise heat rates, reserves and synchronous inertia;
- LNG terminals and hydrogen, ammonia and heat chains;
- multi-pollutant emission zones with caps, banking and carbon capture;
- multi-stage capacity expansion of generation, transmission, storage and demand.

2) A multi-scale design in which the same element definitions are used from 15-minute operation to 30-year planning, with the scales coupled through SDDP optimality and feasibility cuts (Section IV-E).
3) Three solution methods—monolithic LP/MIP; SDDP with apertures, multi-cut and elastic-filter feasibility cuts; and a multi-level cascade—plus an investment master over SDDP operation (Section IV).
4) An open software architecture (Section V):
   - JSON and Apache Parquet data, and PAMPL user constraints;
   - run-time solver plugins and solver-neutral warm starts;
   - exportable LP files with structural fingerprints;
   - converters from PLP, PLEXOS, PSR SDDP, PSS/E and pandapower;
   - command-line, browser and web-service interfaces.
5) Validation on IEEE benchmark networks, pandapower and published SDDP benchmarks (Section VI), and a comparison of GTOpt with the official daily schedule of the Chilean national system on 14 days, computed by the system operator with PLEXOS (Section VII).

The remainder of this paper is organized as follows. Section II reviews the state of the art. Section III presents the formulation. Section IV describes the solution methods and the multi-scale coupling. Section V details the software architecture. Section VI presents the validation on benchmarks, and Section VII the comparison with the Chilean daily schedule. Section VIII concludes and outlines future work.

II. Literature Review

A. Generation and Transmission Expansion Planning

GTEP is a classical problem in power systems that has been studied extensively since the 1970s. Romero et al. [11] provided canonical test systems and mathematical models—including transportation, hybrid, DC power flow, and disjunctive formulations—that remain standard benchmarks. Hemmati et al. [2] presented a comprehensive review covering generation expansion planning (GEP), transmission expansion planning (TEP), and their integration. More recently, Mahdavi et al. [3] classified TEP literature into categories by mathematical model, solution method, and uncertainty handling, identifying a trend toward stochastic, multi-objective, and market-based formulations.

Lumbreras and Ramos [1] surveyed modern challenges in TEP including renewable integration, energy storage, and the need for multi-stage stochastic models. These challenges motivate the development of tools that can handle joint generation, transmission, and storage expansion under operational uncertainty.

B. DC Optimal Power Flow

The DC power flow approximation linearizes the AC power flow equations by assuming lossless lines, flat voltage profiles, and small angle differences [12]. Despite its simplifications, DC OPF is the standard formulation for TEP because it yields a linear program amenable to efficient solution [13]. Molzahn and Hiskens [14] provided a comprehensive survey of relaxations and approximations, confirming that DC OPF remains the workhorse for expansion planning despite advances in convex relaxations of AC OPF.

GTOpt adopts the B-θ formulation (susceptance-angle) for DC OPF, as recommended by Stott et al. [12], with extensions for resistive losses via piecewise-linear approximations and phase-shifting transformers.

C. Stochastic Dual Dynamic Programming

The Stochastic Dual Dynamic Programming (SDDP) algorithm, introduced by Pereira and Pinto [4], is the standard method for solving large-scale multi-stage stochastic programs in hydrothermal scheduling. The algorithm constructs piecewise-linear approximations of the expected cost-to-go function through iterative forward and backward passes.

Shapiro [15] provided theoretical convergence analysis of SDDP, and Philpott and de Matos [16] extended the algorithm with risk-averse sampling. Recent advances include batch learning SDDP [17], which frames the algorithm as Q-learning for improved convergence, and extensions to stochastic OPF under uncertainty [18].

In the Julia ecosystem, Dowson and Kapelevich [8] developed SDDP.jl, a modular SDDP implementation, together with a set of reference problems used here for validation (Section VI). GTOpt provides a C++ SDDP implementation with backward-pass apertures, multi-cut and Markov-chain variants, elastic-filter feasibility cuts, and boundary cuts that couple studies of different horizons (Section IV-E).

D. Energy Storage in Expansion Planning

The integration of battery energy storage systems (BESS) into expansion planning has received significant attention. Go et al. [19] assessed the economic value of co-optimized grid-scale storage investments. Frazier et al. [20] presented an integrated MILP for generation, transmission, and storage expansion with Benders decomposition.

Sandia National Laboratories released QuESt Planning [21], an open-source capacity expansion tool focused on storage, while IDAES-GTEP [7] provides a modular Pyomo-based

framework for GTEP. GTOpt supports BESS with detailed charge/discharge efficiency modeling, state-of-charge tracking across stages, and co-located generation-storage coupling via its converter mechanism.

E. Open-Source Power System Tools

Table I compares GTOpt with the leading open-source GTEP tools. PyPSA [5] is the most widely adopted, offering multi-sector coupling and a large community, and targets linear expansion and dispatch rather than stochastic hydrothermal scheduling. GenX [6] provides a flexible Julia/JuMP framework but lacks native DC OPF and hydro cascade modeling. pandapower [22] excels at power flow analysis but does not support expansion planning.

GTOpt distinguishes itself through:

1) one formulation used from daily operation to multi-decade planning, with the time scales coupled by SDDP cuts;
2) complete hydro cascade modeling with reservoirs, waterways, turbines, pumps and seepage;
3) a generic water-rights layer for irrigation agreements with volume-dependent dynamic bounds;
4) three solution methods, including SDDP and cascade decomposition;
5) an auditable model: a published formulation, exportable LP files and structural fingerprints;
6) native Apache Parquet I/O and converters from the commercial models used in the region.

III. Mathematical Formulation

A. Sets and Indices

The GTEP problem is defined over the following sets:

- $\mathcal{S}$: scenarios, indexed by s, with probability π_s
- $\mathcal{T}$: stages (planning periods), indexed by t
- $\mathcal{B}_t$: blocks (operating periods) in stage t, indexed by b, with duration Δ_b [h]
- $\mathcal{N}$: buses (electrical nodes), indexed by n
- $\mathcal{G}$: generators, indexed by g, with bus n_g
- $\mathcal{D}$: demands, indexed by d, with bus n_d
- $\mathcal{L}$: transmission lines, indexed by ℓ, connecting buses a_ℓ and b_ℓ
- $\mathcal{E}$: batteries (energy storage), indexed by e
- $\mathcal{V}$: converters (battery↔grid), indexed by v
- $\mathcal{J}$: hydraulic junctions, indexed by j
- $\mathcal{R}$: reservoirs, indexed by r, with junction j_r
- $\mathcal{W}$: waterways, indexed by w
- $\mathcal{H}$: turbines, indexed by h
- $\mathcal{Z}$: reserve zones, indexed by z
- $\mathcal{P}$: reserve provisions, indexed by p
- $\mathcal{FR}$: flow rights (flow-based water rights), indexed by i
- $\mathcal{VR}$: volume rights (volume-based water rights), indexed by o
- $\mathcal{H}_P$: hydraulic pumps, indexed by h
- $\mathcal{LNG}$: LNG storage terminals, indexed by τ
- $\mathcal{UC}$: generators with unit commitment, indexed by g
- $\mathcal{IZ}$: inertia zones, indexed by z
- $\mathcal{IP}$: inertia provisions, indexed by p

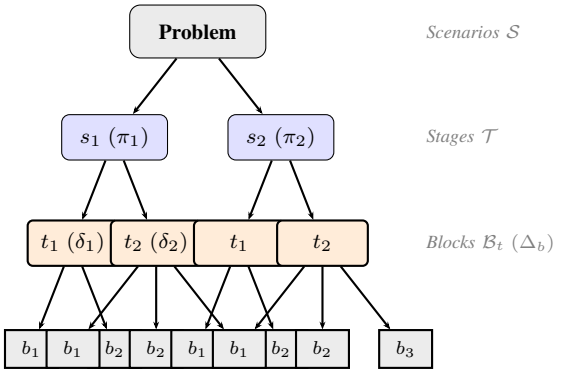

Fig. 1. Three-level time hierarchy. The problem decomposes into scenarios s (probability π_s), stages t (discount factor δ_t), and blocks b (duration Δ_b).

B. Decision Variables

Operational variables (per scenario s, stage t, block b):

$$p_{g,s,t,b} \in [\underline{p}_g, \overline{p}_g] \quad \text{generator output [MW]} \tag{1}$$
$$\ell_{d,s,t,b} \in [0, \overline{\ell}_d] \quad \text{served demand [MW]} \tag{2}$$
$$q_{d,s,t,b} \geq 0 \quad \text{unserved demand [MW]} \tag{3}$$
$$f_{\ell,s,t,b} \in [-\overline{F}^{ba}_\ell, \overline{F}^{ab}_\ell] \quad \text{line flow [MW]} \tag{4}$$
$$\theta_{n,s,t,b} \in \mathbb{R} \quad \text{voltage angle [rad]} \tag{5}$$

Storage state variables:

$$e_{e,s,t,b} \in [\underline{e}_e, \overline{e}_e] \quad \text{battery SoC [MWh]} \tag{6}$$
$$v_{r,s,t,b} \in [\underline{v}_r, \overline{v}_r] \quad \text{reservoir volume [hm}^3\text{]} \tag{7}$$
$$v^{\text{lng}}_{\tau,s,t,b} \in [\underline{V}_\tau, \overline{V}_\tau] \quad \text{LNG tank volume [m}^3\text{]} \tag{8}$$

Unit commitment variables (per scenario s, stage t, period p):

$$u_{g,s,t,p} \in \{0,1\} \quad \text{unit status (1 = online)} \tag{9}$$
$$v_{g,s,t,p} \in \{0,1\} \quad \text{startup event} \tag{10}$$
$$w_{g,s,t,p} \in \{0,1\} \quad \text{shutdown event} \tag{11}$$

Investment variables (per stage t):

$$m_{g,t} \in [0, \overline{m}_g] \quad \text{expansion modules built} \tag{12}$$

C. Objective Function

Minimize total expected discounted cost:

$$\min \sum_{s\in\mathcal{S}} \pi_s \sum_{t\in\mathcal{T}} \delta_t \left[\sum_{b\in\mathcal{B}_t} \Delta_b \cdot C^{\text{op}}_{s,t,b} + C^{\text{inv}}_t \right] \tag{13}$$

where $\delta_t = (1+r)^{-\tau_t/8760}$ is the discount factor with annual rate r and τ_t the cumulative hours to stage t.

The operational cost for each (s,t,b) is:

$$C^{\text{op}}_{s,t,b} = \sum_{g\in\mathcal{G}} c_g\, p_{g,s,t,b} + \sum_{d\in\mathcal{D}} c^f_d\, q_{d,s,t,b} + \sum_{\ell\in\mathcal{L}} c_\ell (f^+_{\ell,s,t,b} + f^-_{\ell,s,t,b}) \tag{14}$$

TABLE I
COMPARISON OF OPEN-SOURCE GTEP AND POWER SYSTEM TOOLS

Feature	GTOpt	PyPSA	GenX	IDAES-GTEP	QuESt	PLP
Language	C++26	Python	Julia	Python	Python	Fortran
License	AGPL-3[a]	MIT	GPL-2	BSD-3	BSD-3	–[b]
DC OPF (Kirchhoff)	✓	✓	Transport	✓	–	✓
Hydro cascades	✓	Partial	–	–	–	✓
Battery storage	✓	✓	✓	Partial	✓	Partial
SDDP solver	✓	–	–	–	–	✓
Capacity expansion	✓	✓	✓	✓	✓	–
Multi-scenario	✓	Partial	Partial	–	–	✓
Multi-sector	✓	✓	Partial	–	–	–
Line losses	✓	✓	✓	–	–	✓
Water rights	✓	–	–	–	–	✓
Unit commitment	✓	✓	✓	✓	–	–
Emission caps	✓	✓	✓	–	–	–
Cut-coupled scales	✓	–	–	–	–	Partial
Parquet I/O	✓	–	–	–	–	–
Web GUI	✓	–	–	–	✓	–

[a]AGPL-3.0-or-later with a solver-plugin exception, or commercial. [b]Maintained by the Chilean system operator (CEN).

The investment cost for stage t is:

$$C_t^{\text{inv}} = \sum_{g\in\mathcal{G}} \omega_{g,t}^{\text{cap}} m_{g,t} + \sum_{d\in\mathcal{D}} \omega_{d,t}^{\text{cap}} m_{d,t} + \sum_{\ell\in\mathcal{L}} \omega_{\ell,t}^{\text{cap}} m_{\ell,t} + \sum_{e\in\mathcal{E}} \omega_{e,t}^{\text{cap}} m_{e,t} \quad (15)$$

where $\omega_{(\cdot),t}^{\text{cap}} = K_{(\cdot)}\delta_t T_t/\sigma$ is the weighted investment cost coefficient, c_g is the variable generation cost [\$/MWh], c_d^f is the demand failure penalty [\$/MWh], c_ℓ is the transmission cost [\$/MWh], $K_{(\cdot)}$ is the annualized investment cost [\$/year] per module for each component, and T_t is the stage duration [h]. All objective coefficients are divided by a scaling factor σ (`scale_objective`, default 1000) to improve solver numerical conditioning. Concretely, each operational LP cost coefficient is weighted by $\omega_{s,t,b} = \pi_s\delta_t\Delta_b/\sigma$, so the reported objective equals z/σ.

D. Bus Power Balance

$$\sum_{\substack{g\in\mathcal{G}\\ n_g=n}} (1-\lambda_g) p_{g,s,t,b} - \sum_{\substack{d\in\mathcal{D}\\ n_d=n}} (1+\lambda_d)\,\ell_{d,s,t,b} + \sum_{\ell\in\mathcal{L}} A_{n,\ell}\, f_{\ell,s,t,b} = 0 \quad (16)$$

where λ_g, λ_d are injection and withdrawal loss factors, respectively. The line flow terms use directional contributions rather than a simple incidence matrix: for each line ℓ from bus a_ℓ to b_ℓ, the sending bus sees $-f_\ell^+$ (or $+(1-\lambda_\ell)f_\ell^-$ when losses are modeled) and the receiving bus sees $+(1-\lambda_\ell)f_\ell^+$ (or $-f_\ell^-$), where λ_ℓ is the line loss factor. When $\lambda_\ell = 0$, a single bidirectional variable suffices and the incidence reduces to $A_{n,\ell} \in \{+1, -1, 0\}$. The dual of (16) is the locational marginal price (LMP) at bus n.

E. Demand Balance

When demand curtailment is allowed, served and unserved demand sum to the scheduled load:

$$\ell_{d,s,t,b} + q_{d,s,t,b} = \overline{\ell}_{d,t,b} \quad \forall\, d, s, t, b \quad (17)$$

where $\overline{\ell}_{d,t,b}$ is the scheduled demand at block b, incorporating any demand profile scaling. When curtailment is disabled, the load is fixed: $\ell_{d,s,t,b} = \overline{\ell}_{d,t,b}$.

F. DC Power Flow (Kirchhoff Constraints)

When `use_kirchhoff` is enabled, for each line ℓ and (s, t, b):

$$f_{\ell,s,t,b} = \frac{V_\ell^2}{X_\ell}\left(\theta_{a_\ell,s,t,b} - \theta_{b_\ell,s,t,b}\right) \quad (18)$$

where X_ℓ is the line reactance [Ω] and V_ℓ is the nominal voltage [kV]. A scaling factor σ_θ (`scale_theta`, default 1000) is applied to the angle variables for numerical conditioning: $\theta^{\text{scaled}} = \sigma_\theta \cdot \theta$.

Line flow is bounded by transfer capacities:

$$-\overline{F}_\ell^{ba} \le f_{\ell,s,t,b} \le \overline{F}_\ell^{ab} \quad (19)$$

a) Loop-flow formulation and switching.: Besides the angle-based form (18), Kirchhoff's voltage law can be written over a cycle basis of the network (one row per independent loop, with no angle variables), which is smaller for radial-heavy systems. Lines can be declared switching candidates with a binary state; the Kirchhoff row of a switchable line (or of every cycle through it) is relaxed with a big-M disjunction, giving optimal transmission switching as a MIP.

b) Transformer model.: For tap-changing and phase-shifting transformers, the Kirchhoff constraint generalizes to:

$$-\theta_{a_\ell} + \theta_{b_\ell} + (\tau_\ell\chi_\ell)(f_\ell^+ - f_\ell^-) = -\sigma_\theta\,\psi_\ell \quad (20)$$

where τ_ℓ is the off-nominal tap ratio (default 1), $\chi_\ell = \sigma_\theta X_\ell/V_\ell^2$ is the scaled reactance, and ψ_ℓ is the phase-shift angle [rad]. When $\tau_\ell = 1$ and $\psi_\ell = 0$, (20) reduces to (18).

G. Line Losses

Losses are modeled with one of several formulations, selected per line or system-wide.

a) Linear and piecewise models.: The linear model splits the flow into forward and reverse components $f_\ell^+, f_\ell^- \geq 0$; the receiving bus sees $(1-\lambda_\ell)$ of the power sent. The piecewise model approximates the quadratic loss $R_\ell f^2/V_\ell^2$ with K equal segments per direction, so the loss slope increases with the flow.

b) Loss arbitrage.: Both models are exact only when the LP has no incentive to lose energy. When the sum of the duals of the two end buses is negative, $\pi_{a_\ell} + \pi_{b_\ell} < 0$, which happens with curtailed renewables, inflexible units or congestion with angular coupling, the LP can fill expensive segments first or circulate flow in both directions at once. The losses then become a fictitious energy sink that sets prices at zero instead of letting them become negative. Section VII shows the size of this effect on the Chilean system.

c) Tangent outer approximation with a loss price.: The `tangent_signed_flow` model uses a signed flow f_ℓ and a loss variable $\lambda_\ell^{\mathrm{L}}$ bounded below by K tangents of the convex quadratic [23],

$$\lambda^{\mathrm{L}}_{\ell,s,t,b} \geq 2c_\ell f_\ell^{(k)} f_{\ell,s,t,b} - c_\ell \big(f_\ell^{(k)}\big)^2, \quad k = 1,\dots,K, \tag{21}$$

with $c_\ell = R_\ell/V_\ell^2$. It is bounded above by a secant chord on an auxiliary $v_\ell \geq |f_\ell|$, which closes the bidirectional-flow channel. A small loss price ε_ℓ [\$/MWh] on $\lambda_\ell^{\mathrm{L}}$ removes the remaining incentive to inflate the losses whenever $\varepsilon_\ell \geq \frac{1}{2}|\pi_{a_\ell} + \pi_{b_\ell}|$. Its effect on legitimate operation is proportional to the losses, not to the flow. For lines where the chord is too loose, an L-segment chord with a type-2 special ordered set makes the upper bound tight at the breakpoints, with a worst-case error of $c_\ell \overline{F}_\ell^2/(4L^2)$.

H. Battery Energy Storage

State-of-charge balance for battery e between consecutive blocks:

$$\begin{aligned} e_{e,s,t,b} = e_{e,s,t,b-1}(1-\mu_e^h \Delta_b) \\ + p^{\mathrm{in}}_{e,s,t,b}\cdot \eta_e^{\mathrm{in}} \cdot \Delta_b - p^{\mathrm{out}}_{e,s,t,b}\cdot \Delta_b/\eta_e^{\mathrm{out}} \end{aligned} \tag{22}$$

where η_e^{in} and η_e^{out} are charge and discharge efficiencies, $p^{\mathrm{in}}, p^{\mathrm{out}}$ are charge and discharge power [MW], and $\mu_e^h = \mu_e/8760$ is the hourly self-discharge rate derived from the annual loss rate μ_e. Initial SoC $e_{e,\cdot,\cdot,0} = e_e^{\mathrm{ini}}$ is an equality constraint in the first stage; the final SoC of stage t links to the initial SoC of stage $t{+}1$.

I. Converter Constraints

A converter v couples a battery e to the electrical network through a generator g (discharge path) and a demand d (charge path) via a power conversion rate ρ_v:

$$p_{g,s,t,b} = \rho_v \cdot p^{\mathrm{out}}_{e,s,t,b} \tag{23}$$

$$\ell_{d,s,t,b} = \rho_v \cdot p^{\mathrm{in}}_{e,s,t,b} \tag{24}$$

When the converter has a capacity limit $\overline{C}_v$, a joint bound applies: $p_{g,s,t,b} + \ell_{d,s,t,b} \leq \bar{C}_{v,t}$. Converter capacity is expandable using the same modular framework as (29).

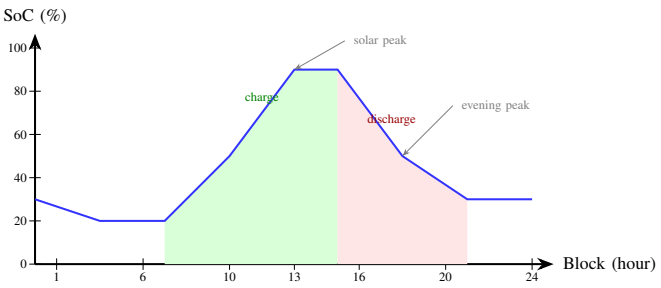

Fig. 2. Battery SoC over 24 blocks: charge during solar hours (green), discharge during evening peak (red).

J. Hydro Cascade

For each junction j with reservoir r, the water balance is:

$$\begin{aligned} v_{r,s,t,b} = v_{r,s,t,b-1} \\ + \alpha_r \Delta_b \Big(\textstyle\sum_{w\in \mathrm{in}(j)} q_w - \sum_{w\in \mathrm{out}(j)} q_w + F_j \Big) \end{aligned} \tag{25}$$

where q_w is the waterway discharge [m^3/s], F_j is the exogenous inflow at junction j, and $\alpha_r = 0.0036$ converts flow rates to volume increments: since 1 m^3/s flowing for 1 hour equals 3600 m^3 = 0.0036 hm^3, $\alpha_r = 3.6 \times 10^{-3}$ [hm^3/(m^3/s·h)].

Each turbine h linking waterway w to generator g provides:

$$p_{g,s,t,b} \leq \rho_h \cdot q_{w,s,t,b} \tag{26}$$

where ρ_h is the water-to-power conversion rate [MW·s/m^3].

a) Volume-dependent turbine efficiency.: In practice, the conversion rate depends on the hydraulic head, which varies with reservoir volume. When defined, ρ_h is replaced by a concave piecewise-linear function:

$$\rho_h(v) = \min_{i=1}^{N} \big\{ c_i + m_i (v - v_i) \big\} \tag{27}$$

where (v_i, m_i, c_i) are volume breakpoints, slopes, and intercepts of N segments. In the SDDP method, the active segment is updated at each forward-pass iteration using the current reservoir volume.

b) Filtration (seepage).: Seepage from a waterway into a reservoir is modeled as a linear function of the average volume:

$$\varphi_{i,s,t,b} = a_i + b_i \cdot \frac{v_{r,s,t,b-1} + v_{r,s,t,b}}{2} \tag{28}$$

where a_i is a constant term and b_i is the seepage rate per unit volume. As with turbine efficiency, this admits a piecewise-linear extension selected by the last known volume.

K. Capacity Expansion

Installed capacity evolves as:

$$\bar{C}_{g,t} = \bar{C}_{g,t-1}(1-\xi_{g,t}) + M_g \cdot m_{g,t} \tag{29}$$

where $\bar{C}_{g,0}$ is the initial installed capacity, $\xi_{g,t}$ is the stage derating factor, M_g is the capacity per expansion module (`expcap`), and $m_{g,t}$ is the number of modules built. The generator output is bounded by the installed capacity:

$$p_{g,s,t,b} \leq \bar{C}_{g,t} \tag{30}$$

Analogous expansion constraints apply to demands, lines, batteries, and converters.

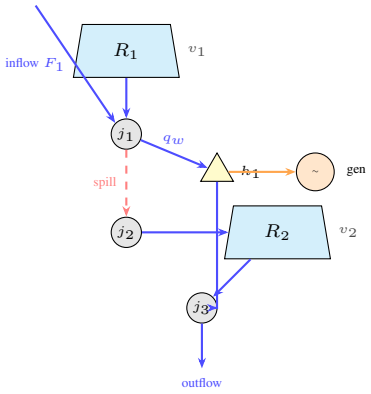

Fig. 3. Hydro cascade example: two reservoirs (R_1, R_2), three junctions (j_1–j_3), one turbine (h_1) driving a generator, with inflow, spillway, and waterway connections.

L. Generator Profile Constraints

When a generator has a capacity profile $\phi_g(s,t,b)$ (e.g., solar irradiance or wind availability), the output is forced to follow the profile:

$$p_{g,s,t,b} + s_{g,s,t,b} = \phi_g(s,t,b) \cdot \bar{C}_{g,t} \tag{31}$$

where $\phi_g(s,t,b) \in [0,1]$ is the per-block capacity factor and $s_{g,s,t,b} \geq 0$ is the curtailed (spilled) generation. The spillover variable carries cost $c_g^{\text{spill}} \cdot \omega_{s,t,b}$, which may be set to zero to allow free curtailment of renewables. This lets the optimizer spill excess renewable production when it is not economically efficient to deliver.

M. Demand Minimum Energy

An optional minimum energy constraint ensures that a demand d is served a minimum total energy within each stage:

$$\sum_{b \in \mathcal{B}_t} \ell^{\text{man}}_{d,s,t,b} \cdot \Delta_b \geq E^{\min}_d \tag{32}$$

where $\ell^{\text{man}}_{d,s,t,b}$ is the mandatory served load and $E^{\min}_d$ [MWh] is the stage energy requirement.

N. Reserve Constraints

For each reserve zone z and (s,t,b):

$$\sum_{p \in \mathcal{P}_z} \gamma_{p,t}\, r^{\text{up}}_{p,s,t,b} + q^{\text{up}}_{z,s,t,b} \geq R^{\text{up}}_{z,s,t,b} \tag{33}$$

where $r^{\text{up}}_{p,s,t,b}$ is the up-reserve provision from provision p, $\gamma_{p,t}$ is the provision factor (typically 1), $q^{\text{up}}_{z,s,t,b}$ is unserved reserve, and $R^{\text{up}}_{z,s,t,b}$ is the zone requirement. A symmetric constraint exists for down-reserve. Reserve provision is bounded by the generator's headroom: $p_{g,s,t,b} + r^{\text{up}}_{p,s,t,b} \leq \bar{C}_{g,t}$.

O. Hydro Pump

A `Pump` element $h \in \mathcal{H}_P$ models a hydraulic pump that consumes electrical power from a demand d and raises water through a waterway w from a downstream junction to an upstream junction—the reverse of a turbine. The conversion constraint ties the electrical load to the pumped flow:

$$\frac{\phi_{h,t}}{\eta_{h,t}}\, \varphi_{w,s,t,b} \leq \ell_{d,s,t,b} \qquad \forall\, h \in \mathcal{H}_P,\ s,t,b \tag{34}$$

where $\phi_{h,t}$ [MW/(m^3/s)] is the pump factor, $\eta_{h,t} \in (0,1]$ is the stage efficiency, $\varphi_{w,s,t,b}$ [m^3/s] is the waterway flow, and $\ell_{d,s,t,b}$ [MW] is the served electrical load. An optional capacity constraint bounds the flow: $\varphi_{w,s,t,b} \leq \overline{Q}_{h,t}$.

The pump element only adds coupling rows to the LP; the waterway retains its junction water balance contribution (Section III-J) and the demand retains its bus power balance contribution (Section III-D). Reversible turbine-pump units (e.g. pump-storage hydro) are modeled as a pair of elements: a `Turbine` on the generation waterway and a `Pump` on the pumping waterway. Economic optimality (round-trip efficiency < 1.0) prevents simultaneous generation and pumping without explicit binary exclusivity constraints.

P. LNG Terminal

An `LngTerminal` element $\tau \in \mathcal{LNG}$ models an LNG storage tank whose inventory is coupled to gas-fired generators through their heat rates. The tank volume balance mirrors the battery SoC equation (Section III-H) with delivery inflow and boil-off:

$$\begin{aligned} v^{\text{lng}}_{\tau,s,t,b} = v^{\text{lng}}_{\tau,s,t,b-1}\big(1 - \lambda^h_\tau \Delta_b\big) + \rho_\tau \Delta_b\, d^{\text{lng}}_{\tau,s,t,b} \\ - \rho_\tau \Delta_b\, \text{vent}_{\tau,s,t,b} - \rho_\tau \Delta_b \sum_{g \in \mathcal{G}_\tau} \kappa^{hr}_g\, p_{g,s,t,b} \end{aligned} \tag{35}$$

where $v^{\text{lng}}_{\tau,s,t,b}$ [m^3] is the tank volume, $\lambda^h_\tau = \lambda_\tau/8760$ is the per-hour boil-off rate, ρ_τ is a dimensionless unit-conversion factor (default 1.0, used when delivery is in m^3/h and volume in m^3 so that $\rho_\tau \cdot \Delta_b \cdot d$ has units m^3), $d^{\text{lng}}_{\tau,s,t,b}$ is the delivery inflow, vent is a vented-gas variable penalized at 'spillway_cost', and κ^{hr}_g [m^3/MWh] is the heat rate of each linked generator $g \in \mathcal{G}_\tau$.

The last term is implemented by injecting a coefficient $-\rho_\tau \kappa^{hr}_g \Delta_b$ onto each generator's generation variable in the existing tank balance row. Volume bounds and initial/final conditions follow the generic storage template. The delivery schedule is fixed to a constant hourly rate $d_{\tau,s,t,b} = D_{\tau,t}/T_t$ across each stage.

This formulation enables the optimizer to manage fuel inventory (buy decisions and storage costs) jointly with economic dispatch, capturing supply security constraints and seasonal LNG price dynamics.

Q. Unit Commitment

The `Commitment` element provides thermal unit commitment with the classical three-bin (3-bin) formulation using status u, startup v, and shutdown w binary variables [24], [25]. The fundamental status transition logic is:

$$u_{g,s,t,p} - u_{g,s,t,p-1} - v_{g,s,t,p} + w_{g,s,t,p} = 0 \tag{36}$$

$$v_{g,s,t,p} + w_{g,s,t,p} \leq 1 \tag{37}$$

$$\underline{P}_g u_{g,s,t,p(b)} \leq p_{g,s,t,b} \leq \overline{P}_g u_{g,s,t,p(b)} \tag{38}$$

where p indexes commitment periods (typically hourly) and b indexes dispatch blocks (sub-hourly allowed). Binary variables live on the commitment-period grid, enabling sub-hourly dispatch with hourly on/off decisions.

Minimum up/down time constraints, ramp limits (while online, during startup, and shutdown), hot/warm/cold startup cost tiers, shutdown costs, no-load costs, and piecewise heat-rate curves are all supported. Commitment constraints apply only to stages flagged `chronological` = true (consecutive hourly blocks). On representative-period stages (duration-weighted blocks not in calendar order), binary variables are silently relaxed to continuous.

R. Simple Commitment

The `SimpleCommitment` element provides a lightweight commitment variant for generators that require minimum-output enforcement without the full startup/shutdown tracking overhead. It introduces a single binary status variable per block:

$$u^{\text{sc}}_{g,s,t,b} \in \{0,1\} \qquad \forall\, g \in \mathcal{UC},\ s,t,b \tag{39}$$

with generation bounds:

$$p_{g,s,t,b} \le \overline{P}_{g,t,b}\, u^{\text{sc}}_{g,s,t,b} \tag{40}$$

$$p_{g,s,t,b} \ge \underline{P}^{d}_{g,t,b}\, u^{\text{sc}}_{g,s,t,b} \tag{41}$$

where $\underline{P}^{d}_{g,t,b}$ (`dispatch_pmin`) is the minimum output when dispatched, defaulting to the generator's $\underline{P}_g$ when unset. When the generator is offline ($u = 0$), generation is forced to zero; when online ($u = 1$), generation must lie in $[\underline{P}^{d}_{g}, \overline{P}_g]$.

Unlike the full `Commitment` element (Section III-Q), `SimpleCommitment`:

- Has no startup/shutdown variables v, w or associated costs
- Has no minimum up/down time constraints
- Has no ramp limits
- Has no no-load cost
- Has no commitment-period abstraction (binary per dispatch block)
- Does *not* require chronological stages

This lighter formulation is sufficient for daily unit commitment in many hydrothermal systems and for PLP-style inertia modeling where only the online/offline status matters. When `must_run` = true, the column is clamped to $u^{\text{sc}}_{g,s,t,b} = 1$ via lower bound $\underline{u}^{\text{sc}} = 1$. When `relax` = true or in SDDP continuous phases, the integrality is dropped and $u^{\text{sc}}_{g,s,t,b} \in [0,1]$.

S. Emissions

Emissions are modeled with three elements: a pollutant registry (CO_2, CH_4, NO_x, SO_2, particulate matter, …); emission zones (a system-wide or regional scope covering a set of pollutants with weights, e.g. GWP-100 for a CO_2-equivalent basket, and an optional price and cap); and emission sources that link generators to zones. Sources can also be derived from fuel emission factors and heat rates. There is no emission variable: each source adds to its generator's dispatch column $p_{g,s,t,b}$ the coefficient

$$\alpha^{\text{em}}_{\nu,t,b} = \mu_{z,\epsilon}\big(1 - \eta^{\text{cap}}_{g,\epsilon,t}\big)\big(f_{\nu,t} + f^{\text{up}}_{\nu,t}\big)\Delta_b \quad \text{[t/MW]}, \tag{42}$$

where $\mu_{z,\epsilon}$ is the zone weight, η^{cap} the carbon-capture fraction, and f, f^{up} the combustion and upstream rates [t/MWh]. A zone price $c^{\text{em}}_{z,t}$ adds $c^{\text{em}}_{z,t}\,\alpha^{\text{em}}_{\nu,t,b}/\Delta_b$ to the generator's cost. A zone cap adds one row per scenario and stage,

$$\sum_{b\in\mathcal{B}_t}\sum_{\nu\in z} \alpha^{\text{em}}_{\nu,t,b}\, p_{g(\nu),s,t,b} - s^{\text{cap}}_{z,s,t} \le \overline{E}_{z,t}, \tag{43}$$

whose dual is the endogenous price of the pollutant; the optional slack s^{cap} makes the cap soft. For cap-and-trade, zones draw on an allowance pool that reuses the storage balance: allocations flow in, emissions are withdrawn, auction purchases are allowed at a price, and unused allowances are banked across stages, so the cap becomes a multi-stage budget. From the bus and cap duals, GTOpt also reports marginal emission factors per bus and hour, including the emissions avoided or caused by storage operation.

T. Multi-Carrier Chains

Hydrogen, ammonia and heat are represented by carrier-typed balance nodes, storage elements and one-stage converters between any pair of carriers. A converter with efficiency η_c withdraws its input $x_{c,s,t,b}$ from the source node and injects $\eta_c x_{c,s,t,b}$ at the destination node, which models electrolysers, fuel cells, Haber–Bosch synthesis, ammonia cracking, electric boilers and concentrated-solar power blocks. Hydrogen, ammonia and thermal storage reuse the storage balance of Section III-H. Together with the LNG terminal of Section III-P, the same LP therefore co-optimizes the power system and its fuel and power-to-X chains.

U. Synchronous Inertia Requirements

As variable renewable penetration increases, maintaining sufficient synchronous inertia for frequency stability becomes a binding operational constraint. GTOpt models inertia requirements through two elements mirroring the reserve-zone formulation (Section III-N).

An `InertiaZone` $z \in \mathcal{IZ}$ specifies a minimum inertia requirement $\overline{R}^{H}_{z,t,b}$ [MWs]:

$$\sum_{p\in\mathcal{P}_z} \Phi_{p,t}\, r^{H}_{p,s,t,b} \ge \overline{R}^{H}_{z,t,b} \tag{44}$$

An `InertiaProvision` $p \in \mathcal{IP}$ links a generator to one or more zones. The provision variable $r^{H}_{p,s,t,b} \in [0, \overline{R}^{H}_{p,t,b}]$ [MW] is bounded by a downward coupling row:

$$p_{g,s,t,b} \ge r^{H}_{p,s,t,b} \tag{45}$$

The conversion factor from MW dispatch to MWs of inertia is:

$$\Phi_{p,t} = \frac{H_p \cdot S_p}{\underline{P}_g} \tag{46}$$

where H_p [s] is the machine inertia constant and S_p [MVA] is the rated apparent power. A user-specified `provision_factor` may override this formula. Shortage is penalized at the zone's `cost` [$/MWs], enabling the optimizer to trade off committed generation against inertia value.

V. Water Rights and Irrigation Agreements

Hydro-dominated power systems often operate under legally-binding irrigation agreements that partition turbine discharge into distinct rights categories—irrigation, electrical, mixed—with volume-dependent allocations that vary with reservoir level. In Chile, the Laja Agreement (1958, updated 2017) and the Maule Convention (1947) govern two of the largest hydro cascades in the national interconnected system, imposing seasonal flow obligations and multi-annual volume accounting constraints on the optimizer.

Traditional tools such as PLP encode each agreement as a monolithic Fortran module, tightly coupling the rights logic to the physical topology. GTOpt instead introduces two generic, data-driven entities—*FlowRight* and *VolumeRight*—together with the flexible *UserConstraint* mechanism, forming a *rights-domain* layer parallel to the physical hydro topology. This separation allows arbitrary irrigation agreements to be specified declaratively in JSON and parameterized AMPL-inspired scripts without modifying the solver code.

1) Flow Right: A flow right $i \in \mathcal{FR}$ represents a flow-based water right [m^3/s] that creates one LP column per block. Two operating modes are supported:

- *Fixed mode*: the extraction flow is fixed at a scheduled discharge value $\bar{\xi}_{i,s,t,b}$ (lower bound equals upper bound).
- *Variable mode*: when a maximum flow $\overline{F}_i > 0$ is specified, the optimizer decides the flow $\xi_{i,s,t,b} \in [0, \overline{F}_i]$.

When `consumptive = true`, the flow right subtracts from the physical junction balance, coupling the rights domain to the physical hydro topology. An optional deficit variable with penalty cost c_i^f ensures feasibility when the right cannot be fully served.

A stage-average hourly flow variable $\bar{\xi}_i^h$ is optionally created to aggregate block-level flows into the duration-weighted average required by the volume accumulation equations:

$$\bar{\xi}^h_{i,s,t} = \sum_{b \in \mathcal{B}_t} \frac{\Delta_b}{T_t} \xi_{i,s,t,b} \tag{47}$$

2) Volume Right: A volume right $o \in \mathcal{VR}$ is modeled as a virtual reservoir (following the same `StorageLP` pattern as batteries) that tracks accumulated volume [hm^3] across stages. The volume balance is:

$$\nu_{o,s,t} = \nu_{o,s,t-1} - \alpha_o \sum_{b \in \mathcal{B}_t} \Delta_b \, \xi^{\text{in}}_{o,s,t,b} \tag{48}$$

where $\nu_{o,s,t}$ is the remaining volume of right o at the end of stage t, $\xi^{\text{in}}_{o,s,t,b}$ is the input extraction rate [m^3/s], and $\alpha_o = 0.0036$ [$hm^3/(m^3/s \cdot h)$] is the flow-to-volume conversion factor.

Volume rights support inter-stage state-variable coupling for SDDP, optional demand satisfaction constraints ($\alpha_o \sum_b \Delta_b \xi^{\text{in}}_b + q^{\text{fail}} \geq D_o$), and an automatic seasonal reset mechanism: when a `reset_month` is configured, the accumulated volume is reset to the initial value at the start of each hydrological year.

3) User Constraints and Parameters: The partition logic that links flow rights together—and the proportional allocation rules of the Maule Convention—is expressed through *UserConstraints*: user-defined linear constraints written in an AMPL-inspired pseudo-script. Each constraint is a named linear (in)equality over LP variables of any system element (generators, demands, flow rights, volume rights, etc.) and can optionally restrict its domain to specific scenarios, stages, or blocks.

For the Laja agreement, the total turbine flow ("gasto turbinado") must equal the sum of all partitioned extraction categories—irrigation rights ("derechos de riego"), electrical rights ("derechos eléctricos"), mixed rights ("derechos mixtos"), and anticipated discharge ("gasto anticipado"):

$$\xi^{\text{gen}} = \xi^{\text{irr}} + \xi^{\text{elec}} + \xi^{\text{mix}} + \xi^{\text{antic}} \tag{49}$$

In the `.pampl` template this constraint reads:

```
constraint laja_particion_derechos
  "Flow partition":
 flow_right('laja_q_turbinado').flow =
   flow_right('laja_der_riego').flow
 + flow_right('laja_der_electrico').flow
 + flow_right('laja_der_mixto').flow
 + flow_right('laja_gasto_anticipado').flow;
```

Entity names follow the PLP Spanish naming conventions (e.g., `laja_der_riego` for PLP's `IQDR`, "Derecho de Riego") to maintain traceability with the reference Fortran implementation. This single UserConstraint replaces the hardcoded balance node that traditional tools implement as a dedicated data structure.

When a FlowRight has a positive `fail_cost` and a nonzero `discharge`, a deficit coupling constraint is automatically generated:

$$\xi_{i,b} + \delta_{i,b} \geq d_{i,b} \tag{50}$$

where $\delta_{i,b}$ is the deficit variable penalized in the objective. This ensures the optimizer only uses the deficit variable when the flow is genuinely constrained by junction balance or dynamic bounds.

For the Maule Convention, the ordinary reserve zone enforces a proportional split between electric (p_e) and irrigation (p_r) extraction from the same shared flow:

$$\xi^{\text{elec}}_{\text{ord}} \leq p_e \left(\xi^{\text{elec}}_{\text{ord}} + \xi^{\text{irr}}_{\text{ord}}\right) \tag{51}$$

$$\xi^{\text{irr}}_{\text{ord}} \leq p_r \left(\xi^{\text{elec}}_{\text{ord}} + \xi^{\text{irr}}_{\text{ord}}\right) \tag{52}$$

where $p_e + p_r = 1$. In GTOpt, these constraints reference the Maule FlowRights using Spanish naming (`maule_gasto_ordinario_elec` for PLP's `IQMOE`, `maule_gasto_ordinario_riego` for `IQMOR`), replacing the hardcoded Fortran proportional-split logic in PLP's `genpdmaulen.f`.

Named scalar or monthly-indexed *UserParams* (p_e, p_r, cost arrays, seasonal demand curves, etc.) are defined in the same script file and resolved automatically from the stage calendar month, enabling seasonally-varying constraints without per-stage hardcoding. Each agreement is maintained in a separate `.pampl` file (e.g., `laja_agreement.pampl`, `maule_agreement.pampl`) and loaded via the `user_constraint_files` array field in the system JSON, keeping each agreement self-contained. The `plp2gtopt` converter generates these templates automatically from the parsed Fortran agreement parameters.

TABLE II
LAJA AGREEMENT VOLUME ZONE ALLOCATION FACTORS.

Zone	**Width** [hm^3]	**Irrig.** factor	**Elec.** factor	**Mixed** factor
Base	—	570	0	30
Zone 1	1 200	0.00	0.05	1.00
Zone 2	170	0.40	0.05	0.00
Zone 3	530	0.40	0.40	0.00
Zone 4	3 682	0.25	0.65	0.00

Base values in m^3/s; zone factors in m^3/s per hm^3. Caps: 5 000 / 1 200 / 30 m^3/s.

4) Volume-Dependent Dynamic Bounds: A key feature of irrigation agreements is that the available rights depend on the current reservoir volume. For example, the Laja agreement defines four volume zones that progressively allocate rights as the reservoir fills:

$$\overline{F}_i(V) = \min\Big(c_k + m_k(V - V_k),\ \overline{F}_i^{\text{cap}}\Big) \tag{53}$$

where (V_k, m_k, c_k) are breakpoints of a piecewise-linear function and V is the current volume of the source reservoir. At each SDDP iteration, the bound is re-evaluated from the forward-pass reservoir volume and applied via `set_col_upp` on the corresponding flow or extraction column.

Table II illustrates the Laja volume zone allocation.

5) Design Advantages: The rights-domain abstraction provides three advantages over the monolithic approach used in traditional tools: (i) new agreements are specified declaratively in JSON and AMPL-inspired scripts without modifying the solver code; (ii) the same two generic entities (FlowRight and VolumeRight) together with UserConstraints express both the Laja and Maule agreements despite their very different structures (4 volume zones vs. 3 reservoir portions, multi-annual vs. seasonal reset, exact partition vs. proportional split); (iii) volume-dependent dynamic bounds integrate naturally with the SDDP forward-pass update mechanism, reusing the same piecewise-linear evaluation used for turbine efficiency. The `plp2gtopt` converter automatically transforms PLP's Fortran agreement parameters into the equivalent GTOpt JSON configuration and `.pampl` template with verified numerical equivalence.

W. Constraint Summary

Table III lists all constraint families in the formulation.

TABLE III
COMPLETE CONSTRAINT SUMMARY.

#	**Constraint**	**Eq.**	$\forall$
C1	Bus power balance	(16)	n, s, t, b
C2	Demand balance	(17)	d, s, t, b
C3	Kirchhoff voltage law	(18)	ℓ, s, t, b
C4	Generator output bounds	(1)	g, s, t, b
C5	Capacity expansion	(29)	g, t
C6	Generator profile	(31)	g, s, t, b
C7	Demand min. energy	(32)	d, s, t
C8	Line capacity	—	ℓ, s, t, b
C9	Battery SoC	(22)	e, s, t, b
C10	Converter coupling	(23)	v, s, t, b
C11	Reserve requirements	(33)	z, s, t, b
C12	Reserve–generator coupling	—	p, s, t, b
C13	Hydro water balance	(25)	j, s, t, b
C14	Turbine conversion	(26)	h, s, t, b
C15	Transformer / PST	(20)	ℓ, s, t, b
C16	Flow right averaging	(47)	i, s, t
C17	Volume right balance	(48)	o, s, t
C18	User-defined constraint	(49)	user-defined
C19	Pump power–flow coupling	(34)	h, s, t, b
C20	LNG tank volume balance	(35)	τ, s, t, b
C21	Unit commitment logic (3-bin)	(36)	g, s, t, p
C22	UC generation bounds	(38)	g, s, t, b
C23	Simple commitment variable	(39)	g, s, t, b
C24	SC generation bounds (upper)	(40)	g, s, t, b
C25	SC generation bounds (lower)	(41)	g, s, t, b
C26	Inertia zone requirement	(44)	z, s, t, b
C27	Inertia provision coupling	(45)	p, s, t, b
C28	Loss tangents (signed flow)	(21)	ℓ, s, t, b, k
C29	Emission cap	(43)	z, s, t

IV. SOLUTION METHODS

GTOpt implements three complementary solution methods, selected with the `method` option. All three solve the same formulation (Section III); they differ in how they decompose the problem across time and scenarios. The problem is organized in *scenes* (groups of scenarios) and *phases* (groups of stages); each (scene, phase) pair is an independent LP that is assembled, solved and written in parallel.

A. Monolithic LP

The default method assembles the entire multi-scenario, multi-stage problem into a single sparse LP/MIP and passes it to the solver backend. The LP is constructed in compressed sparse column (CSC) format via the `FlatLinearProblem` class.

For a system with N buses, G generators, D demands, L lines, E batteries, B blocks per stage, T stages, and S scenarios, the approximate problem size is:

TABLE IV
LP PROBLEM SIZE ESTIMATES

Quantity	Approximate count
Operational variables	$(G{+}2D{+}2L{+}3E{+}N) \times S \times T \times B$
Investment variables	$(G{+}D{+}L{+}E) \times T$
Bus balance rows	$N \times S \times T \times B$
Kirchhoff rows	$L \times S \times T \times B$ (if DC OPF)
SoC balance rows	$E \times S \times T \times B$
Capacity rows	$(G{+}D{+}L{+}E) \times T$

The monolithic method is exact (no decomposition error) and is the natural choice for daily and weekly operation, unit commitment, and deterministic expansion problems that fit in memory. It is also used, with the SDDP cuts of a longer study as end-of-horizon valuation, to solve the short-term scale of Section IV-E.

B. Stochastic Dual Dynamic Programming (SDDP)

For multi-stage stochastic problems, GTOpt implements SDDP following Pereira and Pinto [4]. The phases are solved as separate subproblems linked by state variables (reservoir volumes, storage levels, allowance banks, irrigation-right balances) and by Benders cuts that approximate the expected future cost (Algorithm 1, Fig. 4).

Algorithm 1 SDDP Algorithm in GTOpt

1: Initialize: empty cut sets $\mathcal{C}_t = \emptyset$ for all t
2: **for** iteration $k = 1, 2, \ldots, K$ **do**
3: **Forward pass**: Solve stages $t = 1, \ldots, T$ sequentially for sampled scenarios; record state variables (SoC, reservoir volumes)
4: **Backward pass**: For $t = T, \ldots, 1$, solve each stage for all aperture scenarios; compute dual variables; generate Benders cuts:

$$\alpha_t \geq \hat{\alpha}_t + \hat{\pi}_t^\top (x_t - \hat{x}_t)$$

5: Add cuts to $\mathcal{C}_t$
6: Check convergence: bound gap $\leq \epsilon$, or a stationary gap
7: **end for**

a) Benders Optimality Cuts.: The core mechanism of SDDP is the Benders optimality cut, which approximates the expected cost-to-go function as a piecewise-linear lower bound. At each backward step for stage t, the cut takes the form:

$$\alpha_t \;\geq\; \hat{z}_{t+1} + \sum_i \hat{\mu}_i \big(x_i - \hat{x}_i\big), \tag{54}$$

where α_t is the future cost variable (the cost-to-go approximation at stage t), $\hat{z}_{t+1}$ is the optimal objective value of the stage-$(t{+}1)$ subproblem evaluated at the trial point, $\hat{\mu}_i$ are the dual multipliers of the state-linking constraints (coupling successive stages through reservoir volumes, state-of-charge, etc.), and $\hat{x}_i$ are the trial state variable values from the forward pass. As iterations proceed, the accumulated cuts progressively tighten the piecewise-linear outer approximation of the true expected cost-to-go, driving the lower bound upward toward the optimum.

b) Cut Coefficient Modes.: GTOpt supports two modes for computing the cut coefficients $\hat{\mu}_i$ in (54). The `reduced_cost` mode (default) extracts dual information from the reduced costs of the LP solution, which is numerically robust when state variables are at their bounds. The `row_dual` mode instead reads the row dual values of the state-linking constraints directly. Both modes produce valid cuts; the choice can affect convergence speed depending on problem structure and solver backend.

c) Uncertainty and apertures.: Hydro inflows, renewable capacity factors and demand vary across scenarios. In the backward pass, each phase is re-solved for a set of *apertures*, i.e. alternative realizations of the uncertain data, and the resulting cuts are averaged (single cut) or kept per aperture (multi-cut). Apertures can be sorted from wettest to driest and truncated to the first N, which gives cheap early iterations and full accuracy later. Inflows can follow an AR(1) model, and an experimental Markov-chain variant shares cuts by hydrological state.

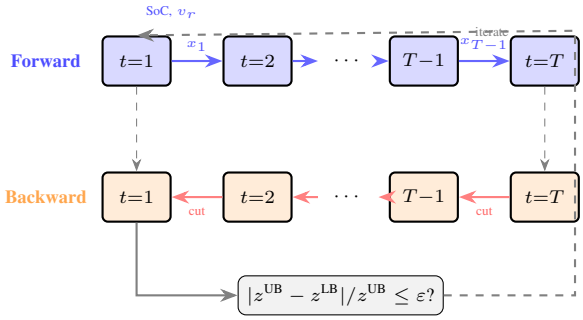

Fig. 4. SDDP forward-backward iteration: the forward pass solves stages sequentially passing state variables (battery SoC, reservoir volume); the backward pass generates Benders cuts. Convergence is checked via the upper/lower bound gap.

d) Feasibility.: Hard limits such as minimum reservoir levels or irrigation deliveries can make a forward-pass subproblem infeasible for a trial state. An elastic filter relaxes the state-linking bounds with penalized slacks, identifies the violated limits, and adds a feasibility cut to the previous phase, so that the policy learns to respect the limit instead of paying a penalty.

e) Convergence.: The upper bound is the mean cost of the forward-pass scenarios and the lower bound is the root-phase objective with all cuts. The algorithm stops when the relative gap falls below a tolerance (default 10^{-4}), or, optionally, when the gap has been stationary over a window of iterations, which accepts the irreducible gap of sampled problems. Statistical stopping rules based on the variance of the upper bound are also available.

f) Engineering for large systems.: Several features make SDDP practical on national-scale systems:

- **Parallelism**: scenes, phases and apertures are solved in parallel by an adaptive work pool that throttles dispatch on free memory.
- **Low-memory mode**: each subproblem is kept as an lz4-compressed snapshot between solves and rebuilt on demand.
- **Warm starts**: simplex bases are stored in a solver-neutral form and reused across iterations, runs and backends.
- **Cut management**: cuts are written as plain CSV after every iteration, so runs can be stopped, resumed (hot start) or extended; inactive cuts can be pruned and capped.
- **Boundary cuts**: an external future-cost function, e.g. the cuts of a longer study or of PLP, values the state at the end of the horizon.

C. Cascade Decomposition

The cascade method runs SDDP over a sequence of levels with progressively more detailed models. Each level defines its own LP options (e.g. single bus, then the full network without losses, then the network with losses) and SDDP options (iterations, apertures, tolerances). A level starts from

TABLE V
TIME SCALES AND METHODS

Scale	Horizon	Blocks	Method
Operation	hours–days	15 min–1 h	monolithic (UC)
Medium term	months–years	hours, load curve	SDDP
Planning	up to 30+ yr	representative	SDDP, cascade, investment master

the cuts of the previous level, kept when their duals remain valid and otherwise re-generated, so the expensive detailed model needs few iterations. A typical configuration uses 4 apertures in level 0, 8 in level 1 and the full set in level 2.

D. Investment Master

For expansion problems that are too large for a monolithic LP, an investment master in the style of OptGen chooses the investment modules, and SDDP evaluates the expected operating cost of each plan. Benders cuts on the investment variables return the marginal value of each module to the master.

E. One Model Across Time Scales

Because the same element definitions are used at every scale, only the time structure (stages and blocks of any duration) and the method change between studies (Table V).

The scales are coupled in two directions:

- **Optimality cuts** carry water values and storage opportunity costs downward: the future-cost function of a planning or medium-term study becomes the boundary cut set of a shorter study, and the cascade inherits the cuts of its coarser levels.
- **Feasibility cuts** enforce reservoir levels and water rights at each scale, so the policy handed from one scale to the next is also feasible.

As a result, the water value used to dispatch a reservoir on a given day is the same marginal value that the medium-term study computed for that reservoir, and the investment plan is evaluated with the same operating model that will later schedule it.

F. Solver Backends

All methods delegate LP/MIP solution to a plugin (`libgtopt_solver_*.so`) loaded at run time. The supported backends are CLP/CBC [26], HiGHS [27], SCIP, NVIDIA cuOpt (GPU), CPLEX, Gurobi and MindOpt. Without an explicit choice, the first available of CPLEX, HiGHS, MindOpt, CBC and CLP is used. Each solve reports its status and an estimate of the condition number, and per-element variable scaling, objective scaling and row/column equilibration (row-max or Ruiz) keep models with kWh-scale batteries and TWh-scale reservoirs numerically stable.

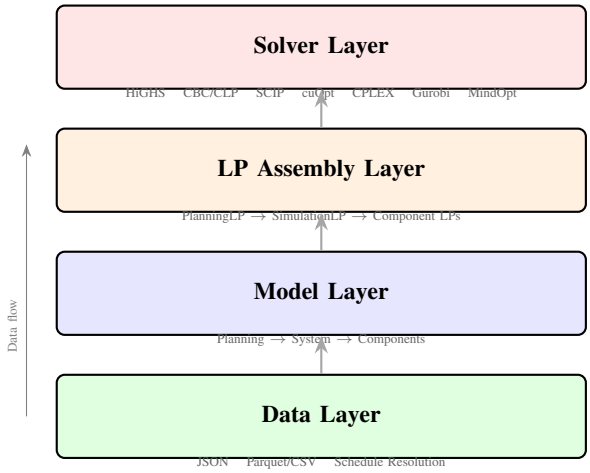

Fig. 5. GTOpt layered architecture: data ingestion, model construction, LP assembly, and pluggable solver backends.

V. SOFTWARE ARCHITECTURE

A. Design Principles

The software architecture of GTOpt is organized around four core principles:

1) **Performance**: C++26 with cache-friendly sorted flat maps for sparse-matrix assembly, parallel assembly and solution of the (scene, phase) subproblems, and compressed in-memory LP snapshots.
2) **Modularity**: Each power system component (generator, demand, line, battery, reservoir, etc.) has its own LP assembly class (`*_lp.cpp`) that independently adds variables, constraints, and objective terms to the shared `LinearProblem`.
3) **Extensibility**: New components and solver backends can be added without modifying the core LP engine. Solver backends are loaded as dynamic plugins (`libgtopt_solver_*.so`).
4) **Interoperability**: JSON configuration with native Apache Arrow/Parquet I/O enables seamless integration with Python, R, and other data science ecosystems.

B. System Architecture

Fig. 5 illustrates the layered architecture.

The software is organized into four layers:

1) **Data Layer**: JSON parsing (via DAW JSON Link, a zero-allocation header-only library), Parquet/CSV I/O (via Apache Arrow C++ SDK), and schedule resolution (`FieldSched<T>` variant type supporting scalar, vector, or filename references).
2) **Model Layer**: The `Planning` → `System` → component hierarchy that maps JSON input to C++ data structures.
3) **LP Assembly Layer**: `PlanningLP` orchestrates `SimulationLP` instances (one per scenario/stage), which in turn assemble component LP objects (`GeneratorLP`, `DemandLP`, `LineLP`, `BatteryLP`, `ReservoirLP`, etc.) into a shared `FlatLinearProblem`.

4) **Solver Layer**: The `SolverBackend` interface abstracts LP/MIP solution. Implementations are loaded as dynamic plugins from configurable search paths.

C. Data Model and JSON Interface

The input is specified as one or more JSON files containing three sections:

```
{
  "options": { ... },
  "simulation": {
    "block_array": [...],
    "stage_array": [...],
    "scenario_array": [...]
  },
  "system": {
    "bus_array": [...],
    "generator_array": [...],
    "demand_array": [...],
    "line_array": [...],
    "battery_array": [...],
    ...
  }
}
```

Listing 1. Top-level JSON structure

Time-series data (demand profiles, generator capacity factors, reservoir inflows) can be specified inline as JSON arrays, or as references to external Parquet or CSV files in the `input_directory`. This hybrid approach keeps the JSON configuration compact while supporting arbitrarily large time-series via columnar Parquet storage. Elements can be referenced by name or by numeric identifier. The command `gtopt -json-schema` prints the JSON Schema of the input, generated from the C++ types, for editor completion and validation. Results are written per (scene, phase) as Parquet files (snappy or zstd) or CSV, readable by pandas, DuckDB, Spark or Power BI.

D. User Constraints: PAMPL

Operating rules that are not part of the element library are written in PAMPL, an AMPL-like language read from a text file. PAMPL constraints reference element variables by name, use parameters, sums over element sets and scenario/stage/block domains, and can include state terms that cross phases in SDDP. The Chilean irrigation agreements of the Maule and Laja basins, for example, are built from water-right elements (Section III) plus PAMPL constraints, without changes to the C++ code.

E. Auditability

Every LP/MIP can be written as a standard LP file with readable row and column names, and solved or checked with any external solver. When a run is infeasible, the LP is dumped and `gtopt_check_lp` computes an irreducible infeasible subsystem. An LP *fingerprint* records the structure of the model actually solved (element classes, constraint families and their counts), so that auditors and continuous integration can verify that a run uses the intended formulation. The complete formulation, including the SDDP cut-validity conditions, is published with the source code.

F. Python Toolchain

A Python toolchain is provided for data preparation, conversion and validation:

- **igtopt**: Excel workbook ($\rightarrow$ JSON + Parquet) converter for spreadsheet-based case preparation.
- **Converters** from the models used in the region and in the literature: `plp2gtopt` (PLP, including its cuts), `plexos2gtopt` (PLEXOS inputs), `sddp2gtopt` (PSR SDDP), PSS/E, NREL Sienna, RTS-GMLC and NREL-118.
- **pp2gtopt / gtopt2pp**: pandapower ($\leftrightarrow$ gtopt) bidirectional converters for power flow model exchange.
- **gtopt_compare**: Automated cross-validation against pandapower DC OPF on IEEE test networks.
- **gtopt_diagram**: network and hydro-topology diagrams.
- **gtopt-net**: network reduction for long-term studies, with decisions mapped back onto the full network.
- **run_gtopt / gtopt_monitor**: execution management with a terminal dashboard and live SDDP convergence monitoring.
- **Validation tools**: `gtopt_check_json`, `gtopt_check_lp`, `gtopt_check_output`, `gtopt_check_solvers` for input validation and diagnostics.

G. Web and GUI Services

Two companion services provide graphical interfaces:

- **webservice**: a Next.js REST API and browser UI for submitting optimization jobs, monitoring progress and browsing results.
- **guiservice**: a Python/Flask application for creating and editing cases, viewing the network topology and plotting results in the browser.

H. Engineering Practice

The C++ library, the Python tools and the web services are covered by unit and integration test suites (more than 5 000 C++ test cases), static analysis (clang-tidy and CodeQL) and coverage measurement, all run in continuous integration on every change. The test suite includes the benchmark cases of Section VI, reference results for each case, and solver-specific tolerances where backends may return different optimal dual vertices.

VI. Validation and Benchmark Results

GTOpt is validated on a comprehensive suite of IEEE benchmark networks and derived cases. All test cases converge to optimal solutions (status 0) with zero optimality gap. Table VI summarizes the benchmark suite.

A. Test Systems and Data Sources

Table VII lists the public test systems included in the repository (directory `cases/`) or read by its converters, with their sources. Each case in the repository carries reference results that are checked in continuous integration.

TABLE VI
IEEE BENCHMARK RESULTS (SCALE_OBJECTIVE = 1000, CLP SOLVER)

Case	Type	Buses	Gen	Dem	Lines	Bat	Obj (scaled)	Blocks
Standard IEEE DC OPF (single snapshot)								
`ieee_4b_ori`	DC OPF	4	2	2	5	–	5.000	1
`ieee_9b_ori`	DC OPF	9	3	3	9	–	55.184	1
`ieee_14b_ori`	DC OPF	14	5	11	20	–	8.334	1
`ieee30b`	DC OPF	30	6	21	41	–	5.668	1
`ieee_57b`	DC OPF	57	7	42	80	–	25.016	1
24-hour dispatch with solar profiles								
`ieee_9b`	Solar+OPF	9	3	3	9	–	5400.007	24
`ieee_14b`	Solar+OPF	14	5	11	20	–	1492.283	24
Battery energy storage								
`bat_4b`	BESS+OPF	4	3	2	5	1	14.160	4
`bat_4b_24`	BESS+OPF	4	3	2	5	1	44.862	24
Capacity expansion								
`exp_gen_4b`	Gen. exp.	4	2	2	5	–	5.029	4
`exp_bat_4b`	Bat. exp.	4	3	2	5	1	12.437	4
`c0`	Dem. exp.	1	1	1	0	–	23.163	5 stages

TABLE VII
PUBLIC TEST SYSTEMS USED FOR VALIDATION AND THEIR SOURCES

System	Size	Used for	Source
IEEE 4-bus and battery variants	4 buses	DC OPF, storage, expansion	illustrative
Anderson–Fouad 9-bus	9 buses	DC OPF, solar profile	[28]
IEEE 14-, 30-, 57-bus	14–57 buses	DC OPF, pandapower cross-check	[29]–[31]
IEEE 118-bus	118 buses	DC OPF	[29], [32]
IEEE RTS-96 (24-bus)	24 buses	unit commitment with OPF	[32], [33]
RTS-GMLC	73 buses	UC and dispatch with renewables	[34]
NREL-118	118 buses	high-renewable dispatch	[35]
NREL Sienna test data	various	cases of `PowerSystems.jl`	[36]
Polish `case3375wp`	3 375 buses	large-network DC OPF	[37]
SDDP.jl examples (first steps, hydro valleys, FAST)	1–2 reservoirs	SDDP vs. published bounds	[8], [38]
PSR SDDP sample cases	multi-system	`sddp2gtopt` hydrothermal	[39]
CEN daily schedules (PLEXOS)	∼250 buses	vs. official schedule	Section VII

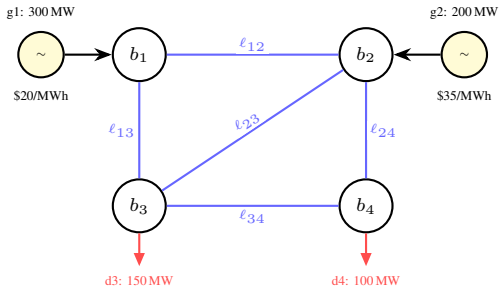

Fig. 6. IEEE 4-bus network: 2 generators, 2 demands, 5 lines forming a fully meshed topology for DC OPF validation.

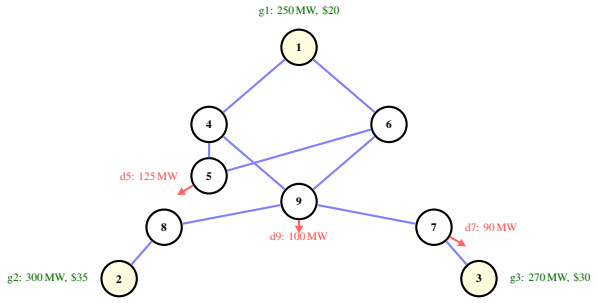

Fig. 7. IEEE 9-bus network: three generation areas (buses 1–3) serve three demands (buses 5, 7, 9) through a meshed 9-line transmission network.

B. IEEE 4-Bus Network

The IEEE 4-bus test case consists of a fully meshed network with 4 buses and 5 transmission lines (Fig. 6). Two generators (g1: 300 MW at $20/MWh, g2: 200 MW at $35/MWh) supply two demands (d3: 150 MW, d4: 100 MW).

Under DC OPF with Kirchhoff constraints, the cheaper generator g1 dispatches 250 MW and g2 idles. The scaled objective value is $5.0 = 250 \times 20/1000$. Power routes through parallel paths (b1→b3 and b1→b2→b3) according to the relative reactances.

C. IEEE 9-Bus Network

The Anderson–Fouad 9-bus system [28] has three generation areas connected by a meshed transmission network (Fig. 7). Generators g1 (250 MW, $20/MWh), g2 (300 MW, $35/MWh), and g3 (270 MW, $30/MWh) serve demands at buses 5, 7, and 9 (total 315 MW).

In the single-snapshot case (`ieee_9b_ori`), economic dispatch follows merit order constrained by network topology. The 24-hour variant (`ieee_9b`) replaces g3 with a solar plant ($0/MWh) whose output follows a daily capacity profile peaking at noon. During daylight hours, zero-cost solar generation displaces g2; at night, g2 carries the load.

D. IEEE 14-Bus Network

The 14-bus system has 5 generators, 11 demands, and 20 transmission lines. The constrained variant (`ieee_14b`) tightens limits on lines ℓ_{1-2} and ℓ_{1-5}, forcing both to saturate simultaneously at the evening demand peak. This validates that shadow prices (Lagrange multipliers of the binding Kirchhoff constraints) are computed correctly.

E. IEEE 30-Bus and 57-Bus Networks

The 30-bus [30] and 57-bus [31] systems test scalability. The 30-bus case has 6 generators, 21 demands, and 41 lines; the 57-bus case has 7 generators, 42 demands, and 80 lines. Both converge to optimal solutions in under one second with the CLP solver.

F. Battery Storage Cases

The `bat_4b` case augments the 4-bus network with a 200 MWh battery at bus 3 (60 MW charge/discharge, 95% round-trip efficiency) and a solar generator with a 4-period capacity profile. The battery charges during solar peak and discharges during evening demand, reducing total system cost compared to the no-storage baseline.

The 24-hour variant (`bat_4b_24`) demonstrates full daily charge/discharge cycling with state-of-charge tracking across all 24 blocks.

G. Capacity Expansion Cases

Three expansion cases validate the investment planning formulation:

- `exp_gen_4b`: Generator g1 can be expanded by up to 290 MW at $1000/year annualized cost. The optimizer adds capacity to avoid dispatching expensive g2.
- `exp_bat_4b`: Battery expansion with 200 MWh modules at $1000/year. The optimizer installs storage to arbitrage solar/thermal cost differentials.
- `c0`: Multi-stage demand expansion (5 stages, 10% discount rate). Demand grows through 20 MW modules at $8760/year. The optimizer balances discounted investment cost against increasing served demand.

H. Cross-Validation with pandapower

The `gtopt_compare` tool performs automated cross-validation against pandapower [22] DC OPF on all IEEE test networks (4, 9, 14, 30, 57 buses). For each case, both tools solve the same network data and compare:

- Objective function value (total generation cost)
- Generator dispatch (MW per generator)
- Bus angles (radians)
- Line flows (MW)
- Locational marginal prices (bus balance duals)

Results match to within solver tolerance ($< 10^{-6}$ relative error) for all standard IEEE cases, confirming the correctness of the DC OPF implementation in GTOpt.

I. SDDP Against Published Benchmarks

The SDDP implementation is checked in continuous integration against published reference values:

- **SDDP.jl *first steps*** [8]: a three-stage hydrothermal problem with stagewise-independent inflows and a published lower bound of 8333.33, confirmed in-repo by the 27-leaf extensive form. Every iterate of GTOpt is a valid lower bound and the sequence is monotone. With the default aperture backward pass it converges to 8055.6, 3.3% below the exact value, because the cuts are built from the resampled apertures rather than from the exact expectation. Depending on the dual vertex returned by the LP solver, it can also reach the exact value.
- **SDDP.jl *Markov uncertainty*** and *generation expansion*: checked against extensive-form and MIP oracles.
- **Infanger–Morton dual sharing**: cut-for-cut agreement of the shared-dual backward pass with the standard one where the theory guarantees it.

In addition, the equivalence between the monolithic LP and SDDP is tested on multi-stage cases: with the same data, the converged SDDP bounds match the monolithic optimum within the convergence tolerance.

VII. Comparison with the Chilean Daily Schedule

The Chilean system operator, the Coordinador Eléctrico Nacional (CEN), computes the daily operation schedule of the national system (SEN) with PLEXOS. For each schedule, the input data and the solution (dispatch, line flows, losses and marginal costs per bus and hour) are available. This makes it possible to compare an independent model against the official one on the full national network.

A. Setup

The comparison uses the CEN schedules of 14 days between October 2025 and May 2026, plus additional days for specific checks. Each case has a horizon of seven days at hourly resolution, with about 250 buses and 315 lines, hydro cascades, batteries, unit commitment and reserves. The PLEXOS inputs are converted with `plexos2gtopt` in a mode that follows the PLEXOS conventions as closely as possible, and solved by GTOpt with the monolithic method (MIP gap 0.5%). The first day of each horizon, which is the day actually scheduled, is compared bus by bus and hour by hour. The comparison covers results only, not computing times.

The CEN has used two loss models: a piecewise-linear model (in 2021) and a generalized, single-pass loss model in the recent schedules. GTOpt is run with two of the models of Section III-G:

- **piecewise**: the traditional piecewise model with free losses ($\varepsilon_\ell = 0$);
- **tangent**: the tangent outer approximation with a loss price $\varepsilon_\ell = 1$ $/MWh.

TABLE VIII
LP LOSSES VERSUS PHYSICAL LOSSES OVER THE HORIZON (MEDIAN AND RANGE OVER 14 CASES)

	Piecewise, free	Tangent, ε=1
LP / physical losses	2.50 (1.05–3.78)	1.03 (1.00–1.19)
Excess over physical	152%	5%

B. Losses

Table VIII compares the losses of the LP with the physical losses $R_\ell f_\ell^2/V_\ell^2$ evaluated at the LP flows, over the seven-day horizon of the 14 cases. With free piecewise losses, the LP losses are 2.5 times the physical value in the median case. The excess comes from the fictitious-sink mechanism of Section III-G: filling expensive segments first and circulating flow in both directions when the sum of the two end-bus duals is negative. On the week of 7 April 2026, for example, the piecewise model loses 91 GWh more than the physical losses, 65 GWh of them in hours with $\pi_{a_\ell} + \pi_{b_\ell} < 0$; the tangent model loses 6 GWh more, and spills the surplus renewable energy instead.

The CEN's own solutions show the same trade-off. In the two 2021 schedules solved with the piecewise model, the PLEXOS losses during the hours with zero marginal cost are five times the secant approximation. The fictitious losses amount to 8.7 and 16.1 GWh, i.e. 4.9% and 8.6% of the daily demand, and the prices stay at exactly zero, never negative. In the recent schedules, solved with the generalized loss model, the fictitious losses disappear but the surplus has no sink left. Marginal costs then become negative: in most of these schedules, slightly negative across the whole system during solar hours, and on one day down to -132 \$/MWh in a congested pocket of the northern grid. Under surplus, a loss model therefore faces a dilemma: either fictitious losses with zero prices, or physical losses with negative prices. The tangent model with a small loss price chooses the latter, consistently with the CEN's current practice.

C. Marginal Costs

The first comparison of marginal costs showed a moderate agreement (median correlation 0.39 and mean absolute error 21 \$/MWh over all buses, 11 \$/MWh at the reference bus, Polpaico 220 kV). Its largest deviations were deeply negative prices at a small load bus in the north, down to -891 \$/MWh, where PLEXOS never goes below -1.3 \$/MWh. Inspecting the LP traced them to the conversion of 173 lines whose limits PLEXOS does not enforce: PLEXOS operates some of them well above their rating (e.g. 190 MW on a 100 MW transformer), whereas the converter had kept the rating as a hard limit. The congested chain created an isolated pocket with negative prices. Converting these lines as PLEXOS treats them (free up to three times the rating, penalized up to six times) removes the artifact (Table IX).

With the tangent loss model and the extended lines, the correlation on 7 April reaches 0.72 and the mean absolute error 9.6 \$/MWh; the lowest price in the north is -8.1 \$/MWh against -13.9 \$/MWh in PLEXOS. The remaining differences come mainly from three sources:

- **Constraints not yet converted.** The -132 \$/MWh pocket of 25 May is not reproduced by any configuration (GTOpt gives 0 to $+9$ \$/MWh at those buses). It is probably set by an operator-defined constraint or flowgate of the PLEXOS model that is not yet active in the conversion.
- **Conversion of model features.** Several PLEXOS constructs (e.g. inflexible gas contracts, storage end conditions, reserve provisions of new batteries) have no one-to-one equivalent, and each conversion choice can move the marginal unit.
- **Degeneracy.** With many units at equal cost and abundant zero-cost energy, the optimal dispatch and duals are not unique, and two exact solvers can legitimately report different marginal costs.

This comparison is ongoing work. Its value is less in the agreement figures than in making the differences explainable. With an open model whose formulation and LP can be inspected, each discrepancy can be traced to a modeling choice in the converter, in GTOpt or in the official model. The artifact above, for instance, was found and removed by reading the duals of a few saturated lines in the LP.

TABLE IX
HOURLY MARGINAL COSTS OF DAY 1, GTOPT VS. PLEXOS (CEN), PIECEWISE LOSSES: CORRELATION AND MEAN ABSOLUTE ERROR [\$/MWH] OVER ALL BUSES, WITH UNENFORCED LINES AS HARD LIMITS (STRICT) AND AS PLEXOS TREATS THEM (EXTENDED)

	Strict		Extended	
Day	Corr.	MAE	Corr.	MAE
2026-04-07	0.33	20.9	0.69	11.3
2026-04-22	0.18	37.1	0.69	19.0
2026-05-25	0.32	45.9	0.49	43.3

VIII. CONCLUSIONS

This paper has presented GTOpt, an open-source optimizer for planning and operating energy systems. The main conclusions are:

1) **One model across time scales.** A single formulation of electricity, water, fuels and LNG, hydrogen, ammonia and heat is used from 15-minute operation to 30-year planning. The scales are coupled through SDDP optimality cuts (water values and storage opportunity costs) and feasibility cuts (reservoir and water-right limits), so that the results of the different horizons are consistent by construction.
2) **A broad formulation.** The model covers the network with Kirchhoff constraints, switching and line losses, storage and hydro cascades with water rights, unit commitment, reserves and inertia, multi-carrier chains, emissions with caps and banking, and expansion of generation, transmission, storage and demand. Operating rules beyond the element library are added through PAMPL.

3) **Methods for large systems.** The monolithic LP/MIP, SDDP with apertures, multi-cut and feasibility cuts, the multi-level cascade and the investment master, together with parallel subproblems, compressed snapshots and solver-neutral warm starts, make national-scale studies tractable on a single server.
4) **Validated and auditable.** GTOpt reproduces IEEE benchmarks and pandapower DC OPF to solver tolerance and produces valid bounds on published SDDP benchmarks. On the Chilean daily schedule, the comparison with the CEN's PLEXOS solutions shows that the line-loss model drives the largest differences, and that a tangent loss model with a small loss price removes fictitious losses. Exportable LP files, structural fingerprints and a published formulation make each result traceable.

A. Future Work

GTOpt is the core of a larger set of modules being developed with the user community under the GTOpt Community organization:

- **Bid-based markets** (`gtopt-bidmark`): clearing offers instead of audited costs, in line with the transition of the Chilean power system. The goal is one dispatch chain from medium-term water values to day-ahead and intra-day scheduling, with co-optimized energy and ancillary services, nodal prices, capacity and long-term energy products, financial transmission rights, and a market simulator to test designs and bidding behaviour.
- **An observatory of the Chilean market** (`gtopt-obs-cl`): tools to observe, reproduce and analyse the daily dispatch of the national system, building on the comparison of Section VII.
- **Regional cases** (`gtopt-latam`): short- and long-term cases of the power systems of Latin America, as open reference data for validation and studies.
- **Closing the remaining gaps with the official schedule**: a systematic attribution of the marginal-cost differences of Section VII to network, conversion and degeneracy effects, and recommendations on loss modeling under surplus.
- **Exact-expectation SDDP**: a backward pass that evaluates the exact expectation over the uncertainty tree, alongside the aperture approximation, and stronger cuts for integer decisions.
- **AC power flow**: second-order cone and linear approximations of reactive power and voltage for operation studies.
- **GPU and machine learning**: wider use of GPU solvers, and learned warm starts, scenario reduction and cut selection for SDDP.

GTOpt is intended to support reproducible research and transparent studies of power systems. Its source code, formulation and test cases are public, and contributions of networks, case studies and code are welcome.

Code and Data Availability

GTOpt is developed by the GTOpt Community at https://github.com/gtopt-community/gtopt and is released under the GNU Affero General Public License v3.0 or later, with an exception for solver plugins; a commercial licence is also available. The repository contains the source code, the complete LP/MIP formulation, the test cases used in Section VI and the tools to convert and compare the cases of Section VII.

Acknowledgments

The authors thank the Coordinador Eléctrico Nacional (CEN) for publishing the input data and solutions of the daily schedule used in Section VII.

References

[1] S. Lumbreras and A. Ramos, "The new challenges to transmission expansion planning. survey of recent practice and literature review," *Electr. Power Syst. Res.*, vol. 134, pp. 19–29, 2016.
[2] R. Hemmati, R.-A. Hooshmand, and A. Khodabakhshian, "Comprehensive review of generation and transmission expansion planning," *IET Gener. Transm. Distrib.*, vol. 7, no. 9, pp. 955–964, 2013.
[3] M. Mahdavi, C. S. Antunez, M. Ajalli, and R. Romero, "Transmission expansion planning: Literature review and classification," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3129–3140, 2019.
[4] M. V. F. Pereira and L. M. V. G. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Math. Programming*, vol. 52, pp. 359–375, 1991.
[5] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for power system analysis," *J. Open Res. Softw.*, vol. 6, no. 1, p. 4, 2018.
[6] J. D. Jenkins and N. A. Sepulveda, "Enhanced decision support for a changing electricity landscape: The GenX configurable electricity resource capacity expansion model," MIT Energy Initiative, Working Paper, 2017.
[7] IDAES Consortium, "IDAES-GTEP: Flexible environments for generation and transmission expansion planning," https://github.com/IDAES/idaes-gtep, 2024.
[8] O. Dowson and L. Kapelevich, "SDDP.jl: A Julia package for stochastic dual dynamic programming," *INFORMS J. Comput.*, vol. 33, no. 1, pp. 27–33, 2021.
[9] M. d. P. Buitrago Villada, C. E. García Bujanda, E. Sierra Baeza, and M. Matus A., "Optimal expansion and reliable renewable energy integration in long-term planning using FESOP," in *Proc. IEEE Kansas Power and Energy Conf. (KPEC)*, 2022, pp. 1–6.
[10] E. Pereira-Bonvallet, S. Püschel-Løvengreen, M. Matus, and R. Moreno, "Optimizing hydrothermal scheduling with non-convex irrigation constraints: Case on the Chilean electricity system," *Energy Procedia*, vol. 87, pp. 132–140, 2016.
[11] R. Romero, A. Monticelli, A. Garcia, and S. Haffner, "Test systems and mathematical models for transmission network expansion planning," *IEE Proc.—Gener. Transm. Distrib.*, vol. 149, no. 1, pp. 27–36, 2002.
[12] B. Stott, J. Jardim, and O. Alsac, "DC power flow revisited," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, 2009.
[13] S. Frank, I. Steponavice, and S. Rebennack, "Operations research in optimal power flow: A guide to recent and emerging developments," *Eur. J. Oper. Res.*, vol. 300, no. 3, pp. 893–914, 2022.
[14] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Found. Trends Electr. Energy Syst.*, vol. 4, no. 1–2, pp. 1–221, 2019.
[15] A. Shapiro, "Analysis of stochastic dual dynamic programming method," *Eur. J. Oper. Res.*, vol. 209, no. 1, pp. 63–72, 2011.
[16] A. B. Philpott and V. L. de Matos, "Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion," *Eur. J. Oper. Res.*, vol. 218, no. 2, pp. 470–483, 2012.
[17] G. Leclercq, M. F. Anjos, and D. Aloise, "Batch learning SDDP for long-term hydrothermal planning," *IEEE Trans. Power Syst.*, vol. 38, no. 5, pp. 4845–4857, 2023.
[18] J. Fletcher, T. W. K. Mak, and C. T. Maravelias, "Stochastic dual dynamic programming for optimal power flow problems under uncertainty," *Eur. J. Oper. Res.*, 2024.
[19] R. S. Go, F. D. Munoz, and J.-P. Watson, "Assessing the economic value of co-optimized grid-scale energy storage investments in supporting high renewable portfolio standards," *Appl. Energy*, vol. 183, pp. 902–913, 2016.

[20] A. W. Frazier, A. Botterud, S. Ericson, and L. Markel, "Integrated expansion planning of electric energy generation, transmission, and storage for handling high shares of wind and solar power generation," *arXiv preprint*, 2023, arXiv:2310.05811.
[21] Sandia National Laboratories, "QuESt planning: Open-source capacity expansion planning tool for energy storage systems," https://www.sandia.gov/ess/, 2024.
[22] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower — an open-source Python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, 2018.
[23] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of AC power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.
[24] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Integrating Renewables in Electricity Markets: Operational Problems*. Springer, 2013.
[25] B. Knueven, J. Ostrowski, and J.-P. Watson, "On mixed-integer programming formulations for the unit commitment problem," *INFORMS Journal on Computing*, vol. 32, no. 4, pp. 857–876, 2020.
[26] J. J. Forrest and R. Lougee-Heimer, "CBC user guide," in *Emerging Theory, Methods, and Applications*. INFORMS, 2005, pp. 257–277.
[27] Q. Huangfu and J. A. J. Hall, "Parallelizing the dual revised simplex method," *Math. Programming Comput.*, vol. 10, no. 1, pp. 119–142, 2018.
[28] P. M. Anderson and A. A. Fouad, *Power Systems Control and Stability*, 2nd ed. Wiley–IEEE Press, 2002.
[29] R. Christie, "Power systems test case archive," University of Washington, Tech. Rep., 1993, https://labs.ece.uw.edu/pstca/.
[30] "IEEE 30 bus test case," University of Washington, Tech. Rep., 1993, https://labs.ece.uw.edu/pstca/pf30/pg_tca30bus.htm.
[31] "IEEE 57 bus test case," University of Washington, Tech. Rep., 1993, https://labs.ece.uw.edu/pstca/pf57/pg_tca57bus.htm.
[32] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris *et al.*, "The power grid library for benchmarking AC optimal power flow algorithms," 2019, https://github.com/power-grid-lib/pglib-opf.
[33] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh, "The IEEE reliability test system-1996," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1010–1020, 1999.
[34] C. Barrows, A. Bloom, A. Ehlen, J. Ikäheimo, J. Jorgenson, D. Krishnamurthy, J. Lau, B. McBennett, M. O'Connell, E. Preston, A. Staid, G. Stephen, and J.-P. Watson, "The IEEE reliability test system: A proposed 2019 update," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 119–127, 2020.
[35] I. Peña, C. B. Martinez-Anido, and B.-M. Hodge, "An extended IEEE 118-bus test system with high renewable penetration," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 281–289, 2018.
[36] J. D. Lara, C. Barrows, D. Thom, D. Krishnamurthy, and D. Callaway, "PowerSystems.jl — a power system data management package for large scale modeling," *SoftwareX*, vol. 15, p. 100747, 2021.
[37] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.
[38] L. Cambier, "FAST: Finally an SDDP toolbox," https://github.com/leopoldcambier/FAST, 2017.
[39] PSR, "PSRClassesInterface.jl: sample SDDP cases," https://github.com/psrenergy/PSRClassesInterface.jl, 2024.