AI for Enhancing Operations Research of Agriculture and Energy

Morteza Kimiaei*
Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
Email: kimiaeim83@univie.ac.at
WWW: `http://www.mat.univie.ac.at/~kimiaei`

(*Corresponding Author)

Anshu Kumar
Department of Computer Science, Czech Technical University
Karlovo Namesti 13, 121 35 Prague 2, Czech Republic
Email: kumarans@fel.cvut.cz

Vyacheslav Kungurtsev
Department of Computer Science, Czech Technical University
Karlovo Namesti 13, 121 35 Prague 2, Czech Republic
Email: vyacheslav.kungurtsev@fel.cvut.cz

Abstract. This paper surveys optimization problems arising in agriculture, energy systems, and water-energy coordination from an operations research perspective. These problems are commonly formulated as integer nonlinear programs, mixed-integer nonlinear programs, or combinatorial set optimization models, characterized by nonlinear physical constraints, discrete decisions, and inter-temporal coupling. Such structures pose significant computational challenges in large-scale and repeated-solution settings. The survey presents a unified mathematical framework and reviews key application domains including crop selection, production planning, optimal power flow, unit commitment, hydropower scheduling, and energy market operations. General-purpose solvers and domain-specific software tools are examined with respect to their algorithmic foundations and practical limitations. Recent advances in learning-assisted optimization are also reviewed, highlighting how machine learning and reinforcement learning enhance classical solvers through warm-starting, constraint screening, branching strategies, policy approximation, and scenario generation. The paper provides a consolidated reference for scalable and reliable optimization methods in complex agriculture and energy systems.

Keywords. Mixed-Integer Nonlinear Programming; Optimal Power Flow; Unit Commitment; Stochastic Optimization; Learning-Enhanced Optimization; Reinforcement Learning; Energy Systems; Agricultural Planning.

Contents

1 Introduction

This section introduces the scope and objectives of the study, situating it at the intersection of Operations Research (OR), agriculture, and energy systems. We outline the class of optimization problems considered, motivate the need for scalable solution techniques, and frame the role of learning-based methods as enhancements to classical optimization rather than substitutes. The introduction provides the conceptual foundation required for the formal problem definitions and application-driven models developed in the subsequent sections.

1.1 Background and Motivation

OR provides the mathematical foundation for decision-making in complex systems involving logistics, scheduling, allocation of scarce resources, and networked infrastructures. In agriculture and energy systems, OR models govern decisions such as crop selection, production scheduling, water allocation, power system operation, unit commitment, and market participation. These applications inherently involve discrete decisions, nonlinear physical relationships, and combinatorial structures.

A large class of these problems can be expressed as integer or mixed-integer optimization problems. Due to their combinatorial nature, obtaining globally optimal solutions often scales exponentially with problem dimension when no exploitable structure is present. This presents severe practical limitations for real-time or large-scale applications, particularly when decisions must be recomputed repeatedly under changing external conditions.

Machine Learning (ML) and Reinforcement Learning (RL), which scale favorably with problem dimension and data availability, offer a complementary paradigm. Rather than replacing optimization models, learning-based methods can exploit repeated structure across problem instances to accelerate solution procedures while preserving feasibility and interpretability.

1.2 Problem Setting and Research Question

This work studies optimization problems that are discrete or mixed discrete–continuous in nature and arise repeatedly under varying external conditions. We begin by defining the decision space through simple bound constraints:

$$\mathbf{X} := \{x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x}\}, \tag{1}$$

where $\underline{x}, \overline{x} \in \mathbb{R}^n$ with $\underline{x} < \overline{x}$.

Integer Nonlinear Programming (INLP). The Integer Nonlinear Programming problem is formulated as

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in C_{\text{in}}, \end{aligned} \tag{2}$$

where the integer nonlinear feasible set is

$$C_{\text{in}} := \{x \in \mathbf{X} \mid g(x) = 0,\ h(x) \le 0,\ x_i \in s_i\mathbb{Z}\ \forall i \in [n]\}. \tag{3}$$

Here, f is a possibly non-convex objective, $g(x)$ and $h(x)$ denote equality and inequality constraints, and $s_i > 0$ are resolution factors. Moreover, for integers $1 \le i \le q$, we define

$$[i{:}q] := \{i, i+1, \dots, q\}.$$

In the special case $i = 1$, we recover the standard notation

$$[q] := [1{:}q] = \{1, \dots, q\}.$$

Mixed-Integer Nonlinear Programming (MINLP). When only a subset of variables is constrained to be integer-valued, the problem becomes a MINLP:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in C_{\text{mi}}, \end{aligned} \tag{4}$$

with feasible set

$$C_{\text{mi}} := \{x \in \mathbf{X} \mid g(x) = 0,\ h(x) \le 0,\ x_i \in s_i\mathbb{Z}\ \forall i \in I\}, \tag{5}$$

where $I \subseteq [n]$ indexes integer decision variables.

Combinatorial Set Optimization (CSO). Many OR problems are more naturally expressed as set selection problems. The generic CSO formulation is

$$\begin{aligned} \min \quad & f(\mathcal{S}) \\ \text{s.t.} \quad & \mathcal{S} \in \mathcal{F}, \end{aligned} \tag{6}$$

where $\mathcal{F}$ is a family of feasible subsets defined by logical, structural, or relational constraints.

Dynamic Programming (DP). A further important class of problems arising in agriculture and energy systems is naturally formulated using DP. These problems are characterized by sequential decision-making over a finite or infinite horizon, with system dynamics governed by state-transition equations. Let

$s_t \in \mathcal{S}$ denote the system state at stage t, and let $a_t \in \mathcal{A}(s_t)$ denote the control action. A generic DP formulation can be written as

$$V_t(s_t) = \min_{a_t \in \mathcal{A}(s_t)} \left\{ \ell(s_t, a_t) + V_{t+1}(s_{t+1}) \right\}, \tag{7}$$

where $\ell(s_t, a_t)$ is the stage cost and the next state s_{t+1} is given by a system dynamics equation $s_{t+1} = F(s_t, a_t)$. The function $V_t(\cdot)$ denotes the value function at stage t, encoding the optimal cost-to-go.

Dynamic programming formulations arise prominently in multi-stage scheduling, inventory control, hydropower reservoir operation, and long-horizon planning problems. While DP avoids explicit combinatorial enumeration, its computational complexity typically grows exponentially with the dimension of the state space, a phenomenon known as the curse of dimensionality. As a result, DP problems are often combined with approximation, decomposition, or learning-based methods in large-scale applications.

Research Question. The central question addressed in this work is:

> How can ML and RL be integrated with classical operations research algorithms to enhance the tractability and scalability of INLP, MINLP, CSO, and DP formulations arising in agriculture and energy systems, without compromising feasibility, correctness, or physical interpretability?

1.3 Challenges in Enhancing OR of Agriculture and Energy

Several challenges arise when enhancing OR methods for agriculture and energy applications. First, many constraints are derived from nonlinear physical laws, such as power flow equations or reservoir dynamics, which induce nonconvex feasible regions. Second, discrete operational decisions introduce combinatorial explosion, particularly in large-scale scheduling and network problems. Third, many applications involve temporal coupling and uncertainty, requiring repeated solution of structurally identical problems under varying exogenous parameters.

These challenges imply that purely data-driven methods are insufficient, as feasibility, safety, and interpretability must be maintained. Any learning-based enhancement must therefore respect the underlying optimization structure defined by (2), (4), and (6).

1.4 Scope and Methodology

The scope of this work spans the full range of optimization problems defined above, including their linear, nonlinear, deterministic, and stochastic variants. We consider both static and multi-stage formulations, as well as problems defined on lattices, graphs, and logical structures.

Our methodology follows a learning-to-optimize paradigm. Classical solvers are treated as the primary mechanism for enforcing feasibility and optimality, while ML and RL models are used to accelerate solver components, approximate policies in sequential settings, and generate realistic scenarios. The learning models exploit repeated structure across instances parameterized by exogenous variables, while the mathematical formulation remains unchanged.

1.5 Contributions

The contributions of this work are summarized as follows:

- A unified mathematical treatment of INLP, MINLP, CSO, and DP formulations arising in agriculture, energy, and water–energy systems.
- A comprehensive survey of classical optimization models and solver architectures used in these domains, highlighting their shared structural properties and computational challenges.
- A structured synthesis of learning-based enhancement strategies—including ML, RL, and generative models—as augmentations to classical OR algorithms, with explicit emphasis on preserving feasibility, correctness, and interpretability.
- An application-driven perspective that systematically connects abstract optimization formulations to representative real-world problems in agriculture, power systems, hydropower, and market operations.

Unlike existing surveys that primarily organize the literature by application domain (e.g., optimal power flow or unit commitment [1, 29, 33]) or by algorithmic paradigm (e.g., convex relaxations or learning-based methods [6, 29]), this work adopts a *structural optimization perspective*. Specifically, we view energy system decision problems as members of parametric families of large-scale, nonconvex optimization problems, and we review solution methodologies according to how they exploit structure in the feasible set, objective function, and repeated solution requirements. This viewpoint allows classical global optimization techniques, convex relaxations, and learning-enhanced methods to be discussed within a unified mathematical framework, highlighting both their theoretical relationships and their complementary roles in practice.

2 Applications of Interest

This section illustrates how the optimization models and algorithmic frameworks introduced earlier arise in a range of practical application domains. The selected applications span OR, energy systems, agriculture, and water–energy coordination, and are chosen to highlight the diversity of mathematical structures encountered in real-world decision-making problems. Despite their different physical interpretations, these applications can be expressed using a common set of optimization paradigms—including linear and nonlinear programming, mixed-integer formulations, and dynamic programming—thereby demonstrating the unifying role of optimization theory across disciplines. The formulations presented below serve both as canonical examples and as motivation for the advanced solver techniques and learning-enhanced methods discussed in subsequent sections.

2.1 OR - Background

OR is a field that uses mathematical modeling, statistical analysis, and optimization techniques to make better decisions in complex systems. It is widely applied in logistics, supply chain management, finance, and resource allocation (cf. [15]). Many OR problems can be expressed in the general form of (4), which can be reduced to NLP, CNLP, INLP, and DP problems. Here, we introduce three simple practical examples.

LP (Production Planning Problem). A company produces n different products using limited resources in the form of

$$\begin{aligned} \max \quad & Z = \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ji} x_i \leq b_j \quad \text{for } j \in [m], \\ & x_i \geq 0 \quad \text{for } i \in [n], \end{aligned}$$

where c_i is profit per unit of product i, a_{ji} is resource j needed per unit of product i, b_j is availability of resource j, and x_i is decision variable (units of product i to produce). The constraints of this LP, known as **resource capacity constraints**, ensure that production does not exceed available resource capacities. They are critical to maintaining feasibility by preventing resource overuse, thereby guaranteeing that production plans remain realistic and executable within available resource limitations.

CNLP (Portfolio Optimization with Risk). This problem allocates capital among assets to minimize risk (variance of return), while ensuring a minimum expected return, i.e.,

$$
\begin{array}{ll}
\min & Z = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i \sigma_{ij} x_j, \\
\text{s.t.} & \sum_{i=1}^{n} r_i x_i \geq R, \\
P & \sum_{i=1}^{n} x_i = 1, \\
 & x_i \geq 0 \quad \text{for all } i \in [n],
\end{array}
$$

where x_i is the proportion of investment in asset i, r_i is the expected return of asset i, σ_{ij} is the covariance of returns between assets i and j, and R is the required lower expected return. The constraints in this CNLP model serve distinct financial goals. The **expected return constraint** (the first constraint) ensures the portfolio achieves at least a minimum desired return, safeguarding investor objectives. The **budget constraint** (the second constraint) enforces that the full investment capital is allocated among the assets, maintaining completeness of the portfolio. Finally, the **non-negativity constraint** (the third constraint) prohibits short-selling by ensuring only non-negative allocations, which reflects many practical investment settings.

DP (Knapsack Problem). Given n items, each with value v_i and weight w_i, this problem chooses a subset to maximize value without exceeding total capacity W, i.e.,

$$
\begin{array}{ll}
\max & \sum_{i=1}^{n} v_i x_i \\
\text{s.t.} & \sum_{i=1}^{n} w_i x_i \leq \overline{w}, \\
 & x_i \in \{0, 1\} \quad \text{for all } i \in [n],
\end{array}
$$

where v_i is the value of item i, w_i is the weight of item i, $\overline{w}$ is the knapsack capacity, and x_i is the decision variable (include item or not). DP solves this problem by defining $V_{i,w}$ as the maximum value achievable with the first i items and total weight $\leq w$. The recursive relation is

$$
V_{i,w} = \begin{cases} 0 & \text{if } i = 0 \text{ or } w = 0, \\ V_{i-1,w} & \text{if } w_i > w, \\ \max\left(V_{i-1,w},\ V_{i-1,w-w_i} + v_i\right) & \text{otherwise.} \end{cases}
$$

The final result is $V_{n,\overline{w}}$, which gives the maximum value without exceeding the capacity. The **capacity constraint** (the first constraint) ensures that the total weight of selected items does not exceed the knapsack's weight limit, preserving feasibility. The **binary decision constraint** (the second constraint) enforces that each item is either included entirely or excluded, modeling the classic 0-1 knapsack scenario where fractional selection is not allowed.

2.2 Energy Grid Optimization

The **energy grid optimization** problem aims to minimize the total cost, containing both generation and transmission costs while ensuring that the grid constraints, including power balance, generation limits, transmission limits, power flow relationships, binary facility assignments, and non-negativity are satisfied. This problem can be formulated as

$$\begin{aligned} \min \quad & f(\mathcal{S}) = \sum_{g \in G} F_g(P_g) + \sum_{l \in L} F_l(P_l) \\ \text{s.t.} \quad & \mathcal{S} \in \mathcal{F} = \{\mathcal{S} \subseteq N \mid \text{power flow and grid constraints}\}, \end{aligned}$$

where N is the set of all nodes (generation plants, consumer nodes, and transmission lines), $G \subseteq N$ is the set of generation nodes (e.g., power plants), $L \subseteq N \times N$ is the set of transmission lines connecting generation plants and consumer nodes, P_g is the power generated at generation node $g \in G$, P_l is the power flowing through transmission line $l \in L$, $F_g(P_g)$ is the generation cost function at plant $g \in G$, and $F_l(P_l)$ is the transmission cost function for line $l \in L$. Let

$$x_{gl} \in \{0, 1\}, \quad \text{for all } g \in G,\, l \in L$$

be the set of binary decision variables, where x_{gl} is true if the transmission line l is used for transmitting power from the generation plant g, and it is false otherwise. Indeed, this binary decision variable indicates whether a transmission line l is used to transmit power from the generation node g.

Let us describe the constraints that $\mathcal{F}$ contains. The first constraint is the **power balance**

$$\sum_{g \in G} P_g = \sum_{c \in C} D_c + \sum_{l \in L} P_l$$

whose goal is to ensure that the total power generated equals the total power demand plus the power transmitted, where $C \subseteq N$ is the set of consumer nodes (e.g., load centers) and D_c is the power demand at a consumer node $c \in C$. The second constraint is the **generation limits**

$$P_g^{\min} \leq P_g \leq P_g^{\max}, \quad \text{for all } g \in G$$

whose goal is to ensure that the power generated at each plant is within its specified limits ($P_g^{\min}$ and $P_g^{\max}$). The third constraint is the **transmission limits**

$$P_l^{\min} \leq P_l \leq P_l^{\max}, \quad \text{for all } l \in L$$

whose goal is to ensure that the power flow through each transmission line is within its specified limits ($P_l^{\min}$ and $P_l^{\max}$). The fourth constraint is the **power flow relationships**

$$P_l = \sum_{g \in G} \alpha_{gl} P_g - \sum_{c \in C} \beta_{lc} D_c, \quad \text{for all } l \in L$$

that express the power flow through each transmission line l in terms of the generation at plant g and the demand at consumer node c. Here α_{gl} denotes the fraction of power generated at node $g \in G$ transmitted through line $l \in L$ and β_{lc} denotes the fraction of demand at consumer node $c \in C$ supplied by transmission line $l \in L$.

The fifth constraint is the **facility allocation constraint**

$$\sum_{g \in G} x_{gl} \leq 1, \quad \text{for all } l \in L$$

whose goal is to ensure that each transmission line is used by at most one generation plant. The sixth constraint is the non-**negativity constraint**

$$P_g \geq 0, \quad P_l \geq 0, \quad \text{for all } g \in G, l \in L$$

whose goal is to ensure that power generation P_g and transmission P_l are non-negative.

2.3 Production Scheduling for Agriculture

Production scheduling in agriculture involves optimizing the allocation of resources (e.g., labor, machinery, water) to maximize crop yield, minimize costs, and meet demand. Below is an example of a MILP for Agriculture:

$$\begin{aligned} \min \quad & C = \sum_{i=1}^{n} c_i x_i + \sum_{j=1}^{m} d_j y_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_{ij} x_i + \sum_{j=1}^{m} b_{ij} y_j \geq r_j \quad \text{for all } j \in [m], \\ & x_i \in \mathbb{Z}^+, \quad y_j \in \mathbb{R}^+, \end{aligned}$$

where C is the total cost, x_i are integer decision variables like the number of workers, y_j are continuous decision variables like the amount of fertilizer, c_i and d_j are cost coefficients, a_{ij} and b_{ij} are resource usage coefficients, and r_j are resource requirements. The constraints in this model, known as **resource requirements constraints**, ensure that allocated resources such as labor, machinery, and materials meet or exceed the minimum levels necessary for production activities. They guarantee feasibility by preventing the insufficient allocation of essential inputs, thereby ensuring adequate resource availability to achieve targeted agricultural outputs.

2.4 Optimal Crop Selection

Efficient crop selection and scheduling play a crucial role in maximizing the economic return from farmland while considering resource and operational constraints. This problem can be formulated as an optimization model that determines the optimal crop mix and tool allocation to maximize the net profit of cultivation. The profit is defined as the difference between the expected revenues from selling the harvested crops and the production costs incurred during the entire sequence of operations required for cultivation [13].

The mathematical formulation of the problem is as follows:

$$
\begin{aligned}
\max \quad & \sum_{i\in[m]} \bar{r}_i \sum_{j\in[n]} a_{i,j,1} \sum_{l\in[u_j]} z_{i,1,l} \\
& - \sum_{i\in[m]} \sum_{k\in[q_i]} \sum_{j\in[n]} a_{i,j,k} \sum_{l\in[u_j]} c_{i,j,l} \sum_{t\in[s_{i,k}:f_{i,k}]} y_{i,k,l,t} \\
\text{s.t.} \quad & \\
& \sum_{i\in[m]} \sum_{j\in[n]} a_{i,j,1} \sum_{l\in[u_j]} z_{i,1,l} \le H, \\
& \left\{ \begin{aligned} & \sum_{j\in[n]} a_{i,j,k-1} \sum_{l\in[u_j]} z_{i,k-1,l} \\ & \quad - \sum_{j\in[n]} a_{i,j,k} \sum_{l\in[u_j]} z_{i,k,l} = 0 \end{aligned} \right. \qquad \forall i \in [m],\ k \in [2{:}q_i], \\
& \sum_{i\in[m]} \sum_{k\in[q_i]} \sum_{j\in[n]} a_{i,j,k} \sum_{l\in[u_j]} c_{i,j,l} \sum_{t\in[s_{i,k}:f_{i,k}]} y_{i,k,l,t} + FC = B, \\
& \left\{ \begin{aligned} & z_{i,k,l} \le \sum_{j\in[n]} a_{i,j,k}\, h_{i,j,l} \sum_{t\in[s_{i,k}:f_{i,k}]} y_{i,k,l,t}, \\ & \forall i \in [m],\ k \in [q_i],\ l \in [u_{j[i,k]}], \end{aligned} \right. \\
& \sum_{i\in I(j,t)} \sum_{k\in[q_i]} a_{i,j,k}\, y_{i,k,l,t} \le 1, \quad \forall j \in [n],\ l \in [u_j],\ t \in [0{:}T], \\
& \sum_{i\in[m]} \sum_{j\in I(j,t)} \sum_{k\in[q_i]} a_{i,j,k} \left(\sum_{l\in[u_j]} y_{i,k,l,t} \right) \le w, \quad \forall t \in [0{:}T], \\
& y_{i,k,l,t} \in \{0,1\}, \quad \forall i \in [m],\ k \in [q_i],\ l \in [u_{j[i,k]}],\ t \in [s_{i,k}{:}f_{i,k}], \\
& z_{i,k,l} \ge 0, \quad \forall i \in [m],\ k \in [q_i],\ l \in [u_{j[i,k]}].
\end{aligned}
$$

The formulation uses the following notation. The integer m denotes the total number of candidate crops, and n denotes the total number of tool types required across all agricultural operations. For each tool type $j \in [n]$, u_j represents the number of identical tools of that type available. Each crop $i \in [m]$ requires a sequence of q_i operations, indexed by $k \in [q_i]$, and $j[i,k]$ denotes the index of the tool type required for the k-th operation of crop i. The planning horizon is discrete and has length T, with time indexed by $t \in [0{:}T]$. For each crop i and operation k, the interval $[s_{i,k}, f_{i,k}]$ specifies the admissible time window during which the operation must be executed.

The parameter H denotes the total number of hectares available for cultivation. For each crop i, $\bar{r}_i$ represents the expected revenue per hectare. The binary parameter $a_{i,j,k}$ equals 1 if tool type j is required for the k-th operation of crop i, and 0 otherwise. The parameter $h_{i,j,l}$ denotes the number of hectares that tool l of type j can process per unit time when assigned to crop i, while $c_{i,j,l}$ denotes the cost per unit time of using that tool. Fixed cultivation costs are captured by FC, and B denotes the total budget available. The set $I(j,t)$ contains the crops whose operations requiring tool type j may be active at time t. The parameter w denotes the number of identical tractor machines available.

The decision variables consist of binary variables $y_{i,k,l,t}$, which take value 1 if tool l of type $j[i,k]$ is assigned to the k-th operation of crop i at time t and 0 otherwise, and continuous variables $z_{i,k,l}$, which represent the number of hectares processed by tool l for the k-th operation of crop i.

The optimization model maximizes total net profit, defined as the difference between cultivation revenues and operational costs. For each crop i, revenue is computed as the expected return per hectare multiplied by the cultivated area, which is fixed by the first operation and enforced to remain identical across all subsequent operations of that crop. The first constraint ensures that the total cultivated area does not exceed the available farmland H. The second constraint enforces inter-operation consistency by requiring that the number of hectares assigned to the first operation of a crop is equal to the number assigned to each subsequent operation, such as ploughing, seeding, and harvesting. The third constraint imposes a budget limitation, ensuring that the total variable cultivation costs together with the fixed costs FC do not exceed the available budget B; although not strictly required for feasibility, this constraint is often imposed as an equality to allow fair comparison between solutions with identical expenditure levels.

The fourth constraint links the continuous cultivated-area variables to the binary assignment variables and tool productivity, ensuring that each crop operation receives sufficient time allocation for a feasible crop–operation–tool assignment. The fifth constraint restricts each tool to be assigned to at most one operation at any given time period. The sixth constraint limits the total number of simultaneously active tool–tractor pairs at any time t to not exceed the available number w. The seventh constraint enforces the binary nature of the assignment variables, while the final constraint imposes non-negativity on the continuous variables representing cultivated hectares.

2.5 Optimal Power Flow and Unit Commitment

Optimal Power Flow (OPF) determines the most efficient way to distribute electricity in a power grid while minimizing costs and satisfying constraints. Unit Commitment (UC) determines the optimal schedule for power generation units to meet demand at minimum cost (cf. [47]).

Three main classic ML methods are used in optimal power flow and unit commitment. Namely, Quadratic Programming (QP), which is used for OPF with quadratic cost functions. MIP is used for Unit Commitment with binary variables like on/off states of turbines in power plants or when to use one source of power production, like solar, and stop using another source of power production, like hydropower. Deep Reinforcement Learning (DRL) is used for real-time OPF and UC in dynamic environments, like when the power plants and the transmission grid have multiple Internet of Things Sensors that are sending data regularly (cf. [43]).

OPF (economic dispatch). This problem is introduced as

$$
\begin{aligned}
\min \quad & C = \sum_{i=1}^{n} (c_i P_i^2 + d_i P_i + e_i) \\
\text{s.t.} \quad & \sum_{i=1}^{n} P_i = D, \\
& P_i^{\min} \leq P_i \leq P_i^{\max} \quad \text{for all } i \in [n],
\end{aligned}
$$

where C is the total power generation cost, P_i is the power output of the generator i, c_i, d_i, e_i are cost coefficients, and D is the total power demand. Moreover, $P_i^{\min}$ and $P_i^{\max}$ are the minimum and maximum power outputs of the generator i. The **demand-supply balance constraint** (the first constraint) ensures that the total power generated meets the total demand exactly, thereby maintaining the stability of the power system. The **generation limits constraint** (the second constraint) guarantees that power generation stays within safe and efficient operational limits.

OPF (AC-OPF). This problem is introduced as

$$
\begin{aligned}
\min \quad & \sum_{i\in N_g} \left(c_i P_{g,i}^2 + d_i P_{g,i} + e_i\right) \\
\text{s.t.} \quad & \\
& P_i = E_i \sum_{j\in N} \left(A_{ij}E_j - B_{ij}F_j\right) + F_i \sum_{j\in N} \left(A_{ij}F_j + B_{ij}E_j\right), \quad \forall i \in N \setminus \{s\}, \\
& Q_i = E_i \sum_{j\in N} \left(A_{ij}F_j + B_{ij}E_j\right) - F_i \sum_{j\in N} \left(A_{ij}E_j - B_{ij}F_j\right), \quad \forall i \in N_{pq}, \\
& P_{g,i}^{\min} \le P_{g,i} \le P_{g,i}^{\max}, \quad \forall i \in N_g, \\
& Q_{g,i}^{\min} \le Q_{g,i} \le Q_{g,i}^{\max}, \quad \forall i \in N_g, \\
& |S_k| \le S_k^{\max}, \quad \forall k \in N_e, \\
& Q_{c,i}^{\min} \le Q_{c,i} \le Q_{c,i}^{\max}, \quad \forall i \in N_c, \\
& T_k^{\min} \le T_k \le T_k^{\max}, \quad \forall k \in N_t, \\
& V_i^{\min} \le V_i \le V_i^{\max}, \quad \forall i \in N.
\end{aligned}
$$

In the formulation of the OPF problem, P_i and Q_i denote the active and reactive powers at node i, while P_{g_i} and Q_{g_i} represent the active and reactive power generation at node i. The voltage magnitude at node i is denoted by V_i. The quantities G_{ij} and B_{ij} are the real and imaginary parts of the (i,j)-th element of the nodal admittance matrix, whereas E_i and F_i are the real and imaginary components of the nodal voltage at node i. The coefficients a_i, b_i, and c_i represent the cost parameters of the i-th generator. The parameter N_{pq} gives the total number of PQ nodes, and N_c is the set of branches with switchable shunt capacitors or reactors. The line flow in branch i is denoted by S_i, and T_i indicates the tap setting of transformer branch i in per unit. The reactive power of the shunt capacitor or reactor at bus i is represented by Q_{c_i}. The set N_{Vpq}^{lim} contains those PQ buses at which voltages violate their prescribed limits, while N_{Qg}^{lim} represents the buses where reactive power generation exceeds its bounds, and N_{Sk}^{lim} includes the branches in which apparent power flows violate their limits. The set N denotes all buses in the system, N_t refers to transformer branches with tap settings, N_e is the set of all branches, and N_g is the set of generator buses. Finally, P_{gs} denotes the real power generation at the slack bus, expressed in per unit.

UC. This problem is formed as

$$\begin{aligned}
\min \quad & C = \sum_{t=1}^{T} \sum_{i=1}^{n} (c_i u_{it} + s_i v_{it} + f_i P_{it}) \\
\text{s.t.} \quad & \sum_{i=1}^{n} P_{it} \geq D_t \quad \text{for all } t \in [T], \\
& u_{it} \in \{0,1\}, \quad v_{it} \in \{0,1\} \quad \text{for all } t \in [T] \text{ and } i \in [n],
\end{aligned}$$

where C is the total cost over time horizon T, u_{it} is a binary variable indicating if generator i is on at time t, v_{it} is a binary variable indicating if generator i is started at time t, and P_{it} is the power output of the generator i at time t. Moreover, c_i, s_i, f_i are cost coefficients for operation, startup, and fuel, respectively, and D_t is the power demand at time t. The **demand fulfillment constraint** (the first constraint) ensures that total generation meets or exceeds demand at every time period. The **operational and startup status constraints** (the second constraint) define the binary nature of generator statuses, capturing operational decisions such as starting up or shutting down generators.

2.6 Ancillary Services and Water Pricing Hydropower Dynamic Programming

This section introduces three optimization problems of central importance to water–energy systems: **Water Pricing (WP)**, **Hydropower Optimization (HO)**, and **Ancillary Services Optimization (ASO)**. These problems address the efficient management of water resources, the determination of economically and socially sound pricing structures, the maximization of hydropower generation value under physical and environmental constraints, and the provision of essential grid stability services. All three formulations naturally exhibit an inter-temporal structure and are most naturally posed within a dynamic optimization or dynamic programming framework, in which decisions and system states evolve over a discrete planning horizon $t \in [T]$ and state-transition constraints enforce physical conservation laws. For further background and applications, see [4].

In the water pricing model, $Q_d(p)$ and $Q_s(p)$ denote, respectively, the demanded and supplied quantities of water at price p, and $C(Q_s(p))$ denotes the total cost of supplying quantity $Q_s(p)$. The resulting welfare function therefore follows the standard surplus formulation, defined as the difference between aggregate willingness-to-pay and supply cost.

In the hydropower optimization model, the generation function $g(Q_t, H_t)$ represents the electrical power produced as a function of turbine discharge Q_t and hydraulic head H_t, while the storage balance constraint enforces water conservation across time. The hydraulic head H_t is fully determined by the reservoir storage level through the functional relationship $H_t = f(S_t)$.

In the ancillary services optimization model, the cost function $c(Q_t, R_t)$ captures efficiency losses or opportunity costs associated with the simultaneous provision of energy and reserve capacity. The index set J denotes the collection of hydropower units contributing to system-wide ancillary services, and the final constraint enforces aggregate frequency regulation requirements at the system level.

WP. This aims to determine optimal tariffs for water resources, balancing economic efficiency, social equity, and environmental sustainability. Proper water pricing signals scarcity, influences water usage patterns, and ensures the allocation of water resources to their highest-valued uses. This problem is formulated as

$$\begin{array}{ll} \max_p & W(p) = \displaystyle\int_0^{Q_d(p)} D^{-1}(q)\,dq - C(Q_s(p)) \\ \text{s.t.} & Q_d(p) \le Q_s(p), \quad \text{for all } p, \\ & p_{\min} \le p \le p_{\max}, \end{array}$$

where $W(p)$ is social welfare, $D^{-1}(q)$ is inverse demand function (marginal willingness-to-pay), $C(Q_s)$ is the cost of supplying water $Q_d(p)$, $Q_s(p)$, and demand and supply at price p. Moreover, $p_{\min}$ denotes the minimum allowable price, typically reflecting policy constraints or regulatory standards ensuring affordability or maintaining social equity. $p_{\max}$ denotes the maximum allowable price, ensuring economic fairness or political acceptability, protecting consumers from excessively high pricing, which could be socially or economically disruptive. The first constraint is the **demand-supply balance constraint**, which ensures water resource sustainability by not exceeding available supply and preventing resource overconsumption, promoting long-term availability. The second constraint is the **price bound constraints**, ensuring social equity and affordability (the lower bound $p_{\min}$) and protecting consumers from excessively high costs (the upper bound $p_{\max}$).

HO. This involves determining the best operating strategy for hydropower facilities to maximize electricity generation or economic returns, given constraints such as reservoir storage limits, inflow uncertainty, environmental regulations, and downstream water usage needs. HO can be formulated as

$$\begin{array}{lll} \displaystyle\max_{Q_t, S_t} & \displaystyle\sum_{t=1}^{T} \pi_t \cdot g(Q_t, H_t) & \\ \text{s.t.} & S_{t+1} = S_t + I_t - Q_t - E_t, & \text{for all } t \in [T], \\ & S_{\min} \le S_t \le S_{\max}, & \text{for all } t \in [T], \\ & Q_{\min} \le Q_t \le Q_{\max}, & \text{for all } t \in [T], \\ & Q_t \ge Q_{\text{env}}, & \text{for all } t \in [T], \\ & H_t = f(S_t), & \text{for all } t \in [T], \end{array}$$

where Q_t is the water release at time t, S_t is the reservoir storage volume at time t, I_t is the reservoir inflow at time t, E_t is the evaporation or losses at time t, $g(Q_t, H_t)$ is the power generation function, and π_t is the electricity price at time t. The **reservoir storage balance constraint** (the first constraint) maintains the physical continuity of water and ensures conservation by accurately accounting for inflows, releases, and losses. The **reservoir storage limits constraint** (the second constraint) ensures the reservoir operates safely by preventing structural risks associated with overfilling or excessive depletion. The **generation limits constraint** (the third constraint) guarantees operational feasibility and protects power-generating equipment from potential damage or inefficiencies. The **downstream water usage constraint** (the fourth constraint) secures the minimum required environmental flows or meets downstream agricultural and municipal water needs. Lastly, the **hydraulic head definition constraint** (the fifth constraint) represents the relationship between reservoir storage levels and hydropower production capacity, ensuring accurate computation of generated electricity.

ASO. Hydropower plants can provide ancillary services—such as frequency regulation, spinning reserve, and voltage control—to ensure grid stability. ASO involves determining optimal operational adjustments of hydropower facilities to maximize grid reliability and economic returns without jeopardizing water usage obligations. This problem can be formulated as

$$
\begin{aligned}
\max_{R_t, Q_t} \quad & \sum_{t=1}^{T} [\rho_t R_t - c(Q_t, R_t)] \\
\text{s.t.} \quad & S_{t+1} = S_t + I_t - Q_t - E_t, && \text{for all } t \in [T], \\
& Q_{\min} \le Q_t \pm R_t \le Q_{\max} && \text{for all } t \in [T], \\
& 0 \le R_t \le R_{\max}, && \text{for all } t \in [T], \\
& \sum_{j \in J} R_{j,t} \ge FR_t, && \text{for all } t \in [T],
\end{aligned}
$$

where R_t is the ancillary service provided at time t, ρ_t is the market price for ancillary services at time t, $c(Q_t, R_t)$ is the cost of ancillary service adjustments, $R_{\max}$ is the maximum ancillary service capacity, FR_t is the required frequency regulation at time t, and J is the set of hydropower plants or units. The **reservoir continuity constraint** (the first constraint) ensures accurate tracking and consistent management of reservoir storage, effectively capturing water availability over time during the provision of ancillary services. The **operational discharge limits constraint** (the second constraint) safeguards the operational feasibility of the plant and protects its infrastructure by restricting allowable adjustments to the base water discharge levels necessary for ancillary service provision. The **ancillary service capacity constraint** (the third constraint) guarantees that ancillary services remain within realistic operational boundaries, dictated by the turbine capacities and grid specifications. Finally, the **system stability constraint** (the fourth constraint) ensures that sufficient frequency regulation capacity is provided to meet grid stability requirements, thus maintaining the reliability and security of the power system.

2.7 Optimization-Based Bidding Strategy for HPP in Energy and Ancillary Markets

In this subsection, we present the optimization model developed by Perekhodtsev and Lave [36], which addresses efficient bidding strategies for hydropower plants participating in joint energy and ancillary services markets. The objective of the model is to maximize the expected revenue of a hydropower producer by optimally distributing available water resources between energy generation and reserve provision, while adhering to physical and market constraints. The formulation explicitly incorporates reservoir dynamics, turbine output limits, and temporal variations in both energy and ancillary service prices.

The optimization problem to maximize the hydro profit is formulated as follows (using a discrete-time approximation over N intervals of size Δt):

$$\begin{aligned} \max_{y_k} \quad & \sum_{k=1}^{N} y_k p_k \, \Delta t \\ \text{s.t.} \quad & \\ & 0 \le y_k \le y_{\max}, && \forall k, \\ & s_{\min} \le s_k \le s_{\max}, && \forall k, \\ & s_N = s_T, \quad s_T \in [s_{\min}, s_{\max}], \end{aligned}$$

where y_k denotes the hydropower plant's output in interval k, $y_{\max}$ is the maximum generation capacity, and p_k is the energy market price at interval k. The reservoir storage s_k is constrained between $s_{\min}$ and $s_{\max}$, and the terminal storage level $s_N = s_T$ is fixed within this range.

Over any interval $[k_1, k_2] \subseteq [1, N]$ in which the reservoir constraints are non-binding, the optimization problem simplifies to:

$$\begin{aligned} \max_{y_k} \quad & \sum_{k=k_1}^{k_2} y_k p_k \, \Delta t \\ \text{s.t.} \quad & \\ & 0 \le y_k \le y_{\max}, \quad k_1 \le k \le k_2, \\ & \sum_{k=k_1}^{k_2} y_k \, \Delta t \le s_{k_1} - s_{k_2} + \sum_{k=k_1}^{k_2} e_k \, \Delta t = S_{[k_1,k_2]}, \end{aligned}$$

where $S_{[k_1,k_2]}$ is the total volume of water available for generation over the interval $[k_1, k_2]$, and e_k represents the natural inflow during interval k. Within such an interval where the storage constraints are not binding, the optimal solution results in a constant water shadow price y_k. Conversely, if the reservoir constraints are binding at time step k, then $y_k = p_k$, as shown in [17].

The outflow from the reservoir is given by:

$$f_k = y_k - e_k,$$

under the assumption that $y_{\max}$ is larger than $\sup |e_k|$, ensuring that spillage does not occur. Given an initial reservoir level s_0, the reservoir level at any time step k evolves according to the balance equation:

$$s_k = s_0 - \sum_{i=1}^{k} f_i \, \Delta t.$$

Ancillary services provision by river dams Ancillary services such as regulation and spinning reserve can be provided by hydroelectric power plants due to their fast ramping capability and operational flexibility. Consider a hydro unit that participates simultaneously in the energy, regulation, and spinning reserve markets over a time interval $[k_1, k_2]$. Let y_k denote the energy output, r_k the regulation capacity, and s_k the spinning reserve provision, priced at p_k, $p_{r,k}$, and $p_{s,k}$, respectively.

The profit maximization problem over $[k_1, k_2]$, assuming that the reservoir capacity constraint is not binding, can be formulated as

$$\begin{aligned}
\max_{y_k, r_k, s_k} \quad & \sum_{k=k_1}^{k_2} \left[y_k p_k + r_k p_{r,k} + s_k p_{s,k} \right] \Delta t \\
\text{subject to:} \quad & 0 \le y_k \le y_{\max}, \quad \forall k, \\
& 0 \le r_k \le r_{\max}, \quad \forall k, \\
& 0 \le s_k \le s_{\max}, \quad \forall k, \\
& y_k + r_k + s_k \le y_{\max}, \quad \forall k, \\
& r_k + s_k \le s_{\max}, \quad \forall k, \\
& y_k \ge r_k, \quad \forall k, \\
& \sum_{k=k_1}^{k_2} y_k \, \Delta t \le s_{k_1} - s_{k_2} + \sum_{k=k_1}^{k_2} e_k \, \Delta t = S_{[k_1, k_2]}.
\end{aligned}$$

The first three constraints enforce the physical upper and lower bounds on generation, regulation, and spinning reserve. The fourth constraint guarantees that the sum of energy production and ancillary service allocations does not exceed the total generator capacity. The fifth constraint ensures that the combined provision of regulation and spinning reserve does not exceed the plant's spinning reserve capability. The sixth constraint maintains enough headroom for downward regulation by requiring the energy output to exceed the regulation allocation. Finally, the last constraint enforces the water balance: the total water released for energy production during $[k_1, k_2]$ must be less than or equal to the available water, which includes initial storage s_{k_1}, final storage s_{k_2}, and cumulative natural inflow e_k.

In the referenced study [36], they assume $s_{\max} = y_{\max}$, reflecting the operational capability of hydro units to ramp up to their full capacity within 10-15 minutes. They also impose $r_{\max} \leq 0.5\, y_{\max}$ to account for the requirement that regulation capacity must allow for symmetric upward and downward movements. Furthermore, their analysis is based on empirical data from the New York Independent System Operator (NYISO), where it was observed that energy, regulation, and reserve prices satisfy $p(t) \geq p_r(t) \geq p_s(t) \geq 0$ for more than 95% of the operating hours in the study period.

2.8 Demand Aggregation and Virtual Power Plant Operation

Demand Aggregation (DA) and Virtual Power Plant (VPP) Operation are two critical approaches in modern energy management, enhancing operational flexibility and economic efficiency within power systems. DA combines individual consumer demands or production units into a unified, manageable energy profile, enabling effective market participation and optimized energy utilization. VPP Operation coordinates distributed energy resources to function collectively as a single, flexible power plant, optimizing performance by balancing generation, storage, and flexible demand.

DA. This involves grouping multiple distributed energy consumers or producers to create a single, manageable load or generation profile. This aggregation allows efficient energy market participation and enhanced operational flexibility. This problem can be formulated as The mathematical formulation can be stated as:

$$
\begin{aligned}
\min \quad & C = \sum_{t=1}^{T} (p_t \cdot L_t^{\text{agg}}) \\
\text{s.t.} \quad & L_t^{\text{agg}} = \sum_{i=1}^{n} l_{it}, && \text{for all } t \in [T], \\
& l_i^{\min} \leq l_{it} \leq l_i^{\max}, && \text{for all } i \in [n],\ t \in [T],
\end{aligned}
$$

where C is the total aggregated energy cost, p_t represents the energy price at time t, and L_t^{agg} is the aggregated load at time t. The decision variable l_{it} is the individual load for consumer i at time t, bounded by minimum and maximum allowable consumption $l_i^{\min}$ and $l_i^{\max}$, respectively. The **aggregation constraint** (the first constraint) ensures accurate aggregation of individual loads into a single manageable profile. The **individual load limits constraint** (the second constraint) guarantees that each consumer operates within realistic and predefined consumption limits.

VPP Operation. This coordinates diverse distributed energy resources (DERs) such as renewable generators, storage units, and demand response to operate collectively as a single flexible power plant. The goal is typically to maximize economic returns or minimize costs while satisfying technical and operational constraints. This problem can be formulated as:

$$
\begin{aligned}
\max \quad & R = \sum_{t=1}^{T} \left(p_t G_t^{\mathrm{VPP}} - c_t E_t^{\mathrm{VPP}} \right) & \\
\text{s.t.} \quad & G_t^{\mathrm{VPP}} = \sum_{j=1}^{m} g_{jt}, & \text{for all } t \in [T], \\
& E_t^{\mathrm{VPP}} = \sum_{k=1}^{s} e_{kt}, & \text{for all } t \in [T], \\
& g_j^{\min} \le g_{jt} \le g_j^{\max}, & \text{for all } j \in [m],\ t \in [T], \\
& e_k^{\min} \le e_{kt} \le e_k^{\max}, & \text{for all } k \in [s],\ t \in [T],
\end{aligned}
$$

where R is the total operational revenue, p_t represents market prices at time t, G_t^{VPP} and E_t^{VPP} are the aggregated generation and energy consumption at time t. Variables g_{jt} and e_{kt} represent generation and consumption of DER j and storage or flexible load k, respectively. The **aggregation constraints** (the first and second constraints) ensure accurate aggregation of distributed generation and flexible energy consumption. The **operational limits constraints** (the third and fourth constraints) guarantee the DERs and loads operate within their specified minimum and maximum technical limits, maintaining feasibility and reliability of the VPP.

2.9 Summary of Application Models

The following table provides a high-level overview of the optimization problems discussed in Section 2. These formulations demonstrate the diversity of mathematical structures—from simple Linear Programming to complex Mixed-Integer Nonlinear Programming—that serve as the foundation for the learning-to-optimize methods proposed in this work.

Table 1: Summary of Optimization Problems and Literature References.

Problem Domain	Model Type	Objective	Refs.
OR			
Production Planning	LP	Maximize Profit	[15]
Portfolio Optimization	CNLP	Minimize Risk/Variance	[15]
Knapsack Problem	DP/ILP	Maximize Item Value	[15]
Agriculture			
Production Scheduling	MILP	Minimize Resource Cost	[13]
Optimal Crop Selection	MINLP	Maximize Cultivation Profit	[13]
Energy Systems			
Economic Dispatch	QP/NLP	Minimize Generation Cost	[47]
AC-OPF	NLP	Minimize Total System Cost	[43]
Unit Commitment	MILP	Minimize Startup & Ops Cost	[47]
VPP/Demand Aggregation	LP	Minimize Operational Costs	[43]
Water–Energy			
Water Pricing (WP)	NLP/Integral	Maximize Social Welfare	[4]
Hydropower Opt. (HO)	DP/NLP	Maximize Energy Revenue	[4, 17]
Ancillary Services (ASO)	MILP/DP	Maximize Stability Returns	[4, 36]
Bidding Strategy	LP/NLP	Maximize Joint Revenue	[36]

While the models above capture the physical and economic complexity of agriculture and energy, their solution requires specialized software architectures capable of handling non-convexities and large-scale discretization, which are reviewed in Section 3.

3 Existing Software

While general-purpose optimization solvers provide the numerical backbone for many large-scale problems, the specific requirements of energy systems—such as physical network structures and temporal coupling—have motivated the development of specialized frameworks. The general-purpose solvers summarized in Table 2 represent the state-of-the-art in numerical optimization, each specializing in distinct mathematical structures:

- `Gurobi` and `CPLEX`: `Gurobi` [16] and `IBM ILOG CPLEX` [21] are the industry standards for Linear Programming (LP) and Mixed-Integer Linear Programming (MILP), essential for large-scale Unit Commitment (UC).
- `Ipopt`: For large-scale Nonlinear Programming (NLP), `Ipopt` [44] implements a primal–dual interior-point method, serving as the de facto choice for local AC-OPF studies.
- `BARON`: The Branch-And-Reduce Optimization Navigator [40] is designed

for non-convex Mixed-Integer Nonlinear Programs (MINLP), providing global optimality guarantees.

- HiGHS: A high-performance open-source alternative, HiGHS [20] provides robust serial and parallel solvers for LP and MILP.
- **SCIP**: The SCIP Optimization Suite [7] is a versatile framework for Constraint Integer Programming and MINLP, serving as a primary vehicle for research into ML-assisted algorithmic components.

Despite their strong theoretical foundations, these solvers often face scalability challenges when applied to large-scale, highly non-convex, or mixed discrete–continuous problems such as hydropower scheduling and integrated energy system optimization. Consequently, recent research has focused on augmenting exact classical solvers with ML techniques—rather than replacing them—to improve solver components such as branching, node selection, and cut generation. Comprehensive surveys [24] demonstrate that these learning-assisted branch-and-bound frameworks can significantly accelerate convergence while preserving global optimality guarantees, particularly for MILP and MINLP problems.

Table 2: Capabilities of existing optimization solvers for problem classes discussed in Sections 2–4

Solver	**LP**	**ILP**	**MILP**	**NLP**	**MINLP**	**CSO / DP**
Gurobi	✓	✓	✓	–	–	✓
CPLEX	✓	✓	✓	–	–	✓
HiGHS	✓	✓	✓	–	–	–
Ipopt	–	–	–	✓	–	–
BARON	✓	✓	✓	✓	✓	–
SCIP	✓	✓	✓	✓	✓	✓

The remainder of this section reviews energy-specific optimization solvers. Unlike general-purpose software, these tools embed power-system physics, uncertainty, and market mechanisms directly into their mathematical formulations, enabling scalable and reliable solutions to large-scale energy problems.

3.1 Power Flow and Grid-Centric Solvers

The following subsubsections review specialized tools designed to handle the nonlinear physics of power networks, ranging from high-fidelity transmission models to convex relaxations and distribution-level agent simulations.

3.1.1 Transmission-Level Solvers: `MATPOWER` and Related Tools

`MATPOWER` [49] is an extensively utilized open-source toolbox for steady-state power system analysis and optimization. Designed primarily for transmission-level studies, it provides native support for solving both AC and DC Optimal Power Flow (OPF) problems by explicitly embedding nonlinear power flow equations into the optimization framework. In particular, `MATPOWER` primarily addresses AC-OPF problems of the form

$$\min_{P_g, Q_g, V, \theta} \sum_{g \in \mathcal{G}} C_g(P_g), \tag{8}$$

where P_g and Q_g represent the active and reactive power outputs of generator g, V and θ denote the voltage magnitudes and phase angles at all buses, and $C_g(\cdot)$ is the generation cost function. This optimization is subject to nonlinear nodal power balance equations derived from Kirchhoff's laws, generator active and reactive capacity limits, voltage magnitude constraints, and transmission line thermal limits.

The resulting optimization problem is a large-scale, sparse nonlinear program (NLP). `MATPOWER` solves this problem using a Newton-based primal–dual interior-point method known as the `MATPOWER` Interior Point Solver (MIPS). The algorithmic foundation of MIPS is detailed in [45], which presents a primal–dual interior-point method specifically tailored for the optimal power flow dispatch. The solver employs a pure-solver implementation that avoids the overhead of commercial modeling environments, utilizing a predictor-corrector mechanism and a direct sparse linear solver to handle the KKT (Karush–Kuhn–Tucker) conditions efficiently. This specialized implementation is what allows `MATPOWER` to maintain high performance even as the network scale increases. MIPS achieves high computational efficiency by exploiting the sparsity and structured Jacobian and Hessian matrices induced by the network admittance topology, substantially reducing computational cost relative to generic NLP solvers applied to power grid problems.

The principal strength of `MATPOWER` lies in its high-fidelity representation of AC power system physics. By directly solving the full nonlinear AC-OPF formulation—rather than relying on convex relaxations—it produces solutions that are feasible with respect to voltage, power flow, and network constraints, albeit with local optimality guarantees inherent to nonconvex optimization. This level of physical accuracy makes MATPOWER an essential tool for transmission planning, congestion management, and reliability studies where precise modeling of grid behavior is critical.

3.1.2 PowerModels.jl and Convex OPF Relaxation Frameworks

PowerModels.jl [11] is a Julia-based modeling framework that generalizes OPF formulations and automatically generates multiple mathematical representations of the same physical problem. Rather than prescribing a single solution approach, PowerModels.jl enables systematic comparison of nonlinear, convex-relaxed, and hybrid formulations for AC power flow optimization.

To address the inherent non-convexity of AC power flow equations, PowerModels.jl implements several convex relaxation techniques. One widely used approach is the second-order cone programming (SOCP) relaxation, which is typically derived from the bus injection model. In this formulation, bilinear voltage product terms are relaxed into rotated second-order cone constraints, yielding a convex approximation that can be solved efficiently using modern conic solvers. While computationally attractive, this relaxation may become weak for meshed transmission networks or under heavy loading conditions.

To improve tightness, PowerModels.jl also supports the quadratic convex (QC) relaxation, which applies McCormick envelopes to bilinear voltage magnitude terms and convex relaxations to trigonometric angle constraints expressed in polar coordinates. The QC relaxation generally provides stronger bounds than SOCP, at the cost of increased problem size and computational complexity. In addition, PowerModels.jl integrates optimization-based bound tightening (OBBT) procedures, which iteratively solve auxiliary optimization problems to refine variable bounds. This bound tightening significantly improves relaxation quality and reduces optimality gaps in global OPF analysis [30].

3.1.3 Distribution System Solvers: GridLAB-D and pandapower

Distribution networks differ fundamentally from transmission systems due to radial topology, phase imbalance, and the presence of numerous heterogeneous loads and distributed energy resources. As a result, specialized solvers are required to capture these characteristics accurately.

GridLAB-D is an agent-based simulation and optimization environment designed to model large-scale distribution systems at high temporal and spatial resolution [32]. Each device—such as loads, photovoltaic units, storage systems, or controllers—is represented as an autonomous agent governed by local physical or behavioral dynamics. GridLAB-D employs quasi-static time-series (QSTS) simulation to resolve interactions across time scales ranging from seconds to years. This framework is particularly well-suited for Volt–VAR optimization, demand response coordination, and large-scale studies of distributed energy resource integration.

`pandapower` [41] is a Python-based framework that combines the power flow capabilities of `PYPOWER` [25] with modern data analysis tools. It implements accelerated Newton–Raphson algorithms with optional just-in-time compilation to achieve efficient power flow and OPF calculations. `pandapower` is designed for automated studies of balanced distribution and transmission networks, making it attractive for scenario analysis, network planning, and integration with data-driven workflows.

3.2 Operational Scheduling and Stochastic Solvers

Moving beyond static grid snapshots, the following tools address the temporal and uncertain nature of energy systems, ranging from unit commitment to multi-stage stochastic scheduling.

3.2.1 `MOST` and Unit Commitment Solvers

The `MATPOWER` Optimal Scheduling Tool (`MOST` [48]) extends the steady-state capabilities of `MATPOWER` [49] to include multi-period scheduling, market clearing, and reliability assessment problems.`MOST` is specifically designed to address Unit Commitment (UC) and related market dispatch formulations.

`MOST` formulates UC as a time-expanded mixed-integer optimization problem of the form

$$\min \sum_{t=1}^{T} \sum_{g \in G} \left(c_g u_{g,t} + s_g v_{g,t} + f_g P_{g,t} \right),$$

where $u_{g,t}$ is the binary on/off status, $v_{g,t}$ is the binary startup decision, $P_{g,t}$ is the active power dispatch, and c_g, s_g, f_g are the fixed, startup, and variable cost coefficients, respectively. The model is subject to inter-temporal constraints including demand balance, generator ramping limits, minimum up and down times, reserve requirements, and network constraints. The resulting large-scale MILP captures the essential operational complexity of electricity markets and power system scheduling.

3.2.2 Stochastic Dual Dynamic Programming (`SDDP`) and `PySP`

Many energy optimization problems, particularly those involving hydropower, renewable generation, and long-term planning, are inherently stochastic. This uncertainty motivates the use of decomposition-based stochastic solvers.

Stochastic Dual Dynamic Programming (`SDDP`) is a specialized algorithm for multistage stochastic optimization problems with convex substructure. It is based on Bellman recursions of the form:

$$V_t(S_t) = \max_{Q_t} \left\{ \pi_t g(Q_t, S_t) + \mathbb{E}\big[V_{t+1}(S_{t+1})\big] \right\},$$

where V_t represents the cost-to-go (value) function at stage t, S_t is the state vector (e.g., reservoir storage levels), Q_t denotes the decision variables (e.g., water release or power generation), π_t is the probability of a given scenario, and $g(\cdot)$ is the immediate stage cost. The value function is approximated using cutting planes derived from dual information. By avoiding explicit enumeration of all scenarios, `SDDP` scales efficiently to long planning horizons and high-dimensional uncertainty, making it the dominant approach for large-scale hydropower scheduling [35].

The theoretical underpinnings and convergence properties of the `SDDP` algorithm are rigorously analyzed by Shapiro [37], who provides a formal framework for the method's performance in multistage stochastic settings. By characterizing `SDDP` as a sampling-based decomposition approach, Shapiro demonstrates how the algorithm constructs lower-bound approximations of the cost-to-go functions through the recursive accumulation of Benders cuts. This mathematical characterization is particularly significant for energy applications as it addresses the "curse of dimensionality" inherent in large-scale reservoir management, proving that the algorithm's complexity grows linearly with the number of stages—a crucial property for the long-horizon scheduling problems discussed herein.

3.2.3 Energy System Planning Models

Energy system planning tools such as `TIMES` [26], `MESSAGE` [28], and `OSeMOSYS` [18] are designed for long-term capacity expansion and policy analysis over multi-decade horizons [27]. Complementing these frameworks, `OSeMOSYS` provides an accessible, open-source alternative designed to lower the barrier for energy system modeling in developing regions. Unlike the more complex, data-intensive proprietary models, `OSeMOSYS` emphasizes a modular structure that allows for the rapid integration of new technologies and policy constraints, making it a standard tool for exploring long-term decarbonization pathways and resource-constrained energy planning. These models typically solve large-scale linear or mixed-integer linear programs that minimize discounted system costs subject to technology availability, emissions constraints, resource limits, and policy targets. Their strength lies in capturing cross-sector interactions and long-term investment trade-offs rather than short-term operational detail.

3.2.4 Agricultural Operations and Supply Chain Scheduling

While energy-sector software is primarily designed to address high-frequency grid stability and short-term operational constraints, agricultural scheduling is characterized by biological latency, seasonal labor availability, and strictly limited operational windows. These characteristics shift the modeling focus from continuous, physics-based dynamics to phenological processes and resource-constrained decision frameworks, as summarized in Table 3.

- **Irrigation and Resource Management:** The FAO-developed tools `CROPWAT` [38] and `AquaCrop` [39] support irrigation planning based on discrete-time soil water balance formulations driven by evapotranspiration and rainfall dynamics. A canonical formulation tracks the root-zone soil moisture deficit D_t as

$$D_{t+1} = D_t + ET_{c,t} - I_t - R_t^{\text{eff}} + DP_t,$$

  where $ET_{c,t}$ denotes crop evapotranspiration, I_t is the applied irrigation depth, R_t^{eff} is effective rainfall, and DP_t represents deep percolation losses. Irrigation decisions are constrained to maintain soil moisture within an admissible regime,

$$0 \leq D_t \leq \texttt{RAW},$$

  where `RAW` (readily available water) defines the threshold beyond which crop water stress occurs. This decision structure is analogous to energy storage management, with soil moisture acting as a state variable, albeit evolving over substantially longer time scales.

- **Farm Management Information Systems (FMIS):** Platforms such as `AgWorld` [3] and `Trimble Ag Software` [42] are widely used for operational planning and scheduling at the farm level. The underlying decision problems can be abstracted as resource-constrained scheduling models. Let $\mathcal{A}$ denote the set of agricultural activities and $\mathcal{K}$ the set of resources. A standard formulation introduces binary variables $x_{i,t} \in \{0,1\}$ indicating whether activity $i \in \mathcal{A}$ starts at time t, with resource feasibility enforced through

$$\sum_{i \in \mathcal{A}} r_{i,k} \sum_{\tau = t - d_i + 1}^{t} x_{i,\tau} \leq R_{k,t}, \quad \forall k \in \mathcal{K},\ t \in \mathcal{T},$$

  where $r_{i,k}$ is the requirement of resource k by activity i, d_i is the activity duration, and $R_{k,t}$ denotes the availability of resource k at time t. The objective typically minimizes makespan or operational cost, subject to weather-dependent field accessibility constraints.

- **Bio-Economic Planning:** For long-term, multi-period decision making, bio-economic farm planning models address large-scale optimization problems that jointly consider crop rotation, soil quality, and economic performance [22], consistent with the agricultural planning scope summarized in Table 3. A representative formulation maximizes the discounted net present value over a planning horizon Y,

$$\max \sum_{y=1}^{Y} \beta^y \sum_{c \in \mathcal{C}} \left(P_{c,y} Y_{c,y} - VC_{c,y}\right) A_{c,y}, \tag{9}$$

subject to inter-temporal crop rotation constraints

$$A_{c,y} \leq \sum_{j \in \mathcal{C}} T_{j,c} \, A_{j,y-1}, \tag{10}$$

where $A_{c,y}$ denotes the land area allocated to crop c in year y and $T_{j,c}$ encodes admissible crop succession rules. These constraints ensure agronomic feasibility while linking operational decisions to long-term soil and economic outcomes.

Table 3: Cross-Sector Mapping of Optimization Software and Applications

Sector	**Representative Tools**	**Primary Optimization Focus**
Energy	`MATPOWER`, `PowerModels.jl`, `PySP`	Physics-based constraints, grid stability, and stochastic power flow.
Agriculture	`CROPWAT`, `AgWorld`	Biological growth cycles, labor logistics, and multi-period planning.

3.3 Summary of Solver–Algorithm Mapping

Table 4 summarizes the principal energy-specific optimization frameworks discussed in this section, highlighting their underlying algorithmic paradigms and the classes of problems they are designed to address. A clear methodological stratification emerges. Physics-based solvers such as `MATPOWER` and `pandapower` focus on accurate steady-state representations of power flow, prioritizing feasibility with respect to network constraints. In contrast, `PowerModels.jl` emphasizes formulation flexibility and convex relaxation, enabling systematic exploration of tractability–accuracy trade-offs in nonconvex OPF problems.

At the distribution level, `GridLAB-D` departs from centralized optimization altogether, adopting an agent-based and quasi-static time-series paradigm that captures device-level dynamics and control interactions across multiple time scales. For inter-temporal and market-oriented problems, MOST and PySP [46] formulate large-scale mixed-integer and stochastic programs that explicitly represent commitment decisions, uncertainty, and non-anticipativity constraints. Finally, long-term planning tools such as TIMES, MESSAGE, and OSeMOSYS abstract away short-term network physics in favor of macro-scale investment and policy analysis.

This diversity of solver architectures reflects the structural heterogeneity of energy optimization problems, spanning nonlinear continuous physics, discrete operational decisions, and stochastic dynamics. While these tools are highly specialized and effective within their intended domains, their computational performance can degrade for large-scale, highly nonconvex, or real-time applications. This observation motivates recent research on ML–assisted optimization, which seeks to augment—rather than replace—these established solvers by accelerating key algorithmic components such as initialization, branching, and constraint handling.

Table 4: Energy-specific optimization frameworks and problem classes

Solver	**Algorithmic Paradigm**	**Target Problem Classes**
MATPOWER	Newton-based PD-IPM	AC/DC OPF, Feasibility
PowerModels.jl	NLP, SOCP, QC, OBBT	AC-OPF Relaxations, Benchmarking
GridLAB-D	Agent-based, QSTS	Distribution Dynamics, Volt-VAR
pandapower	Newton-Raphson PF/OPF	Automated Grid Analysis, Planning
MOST	Time-expanded MILP	Unit Commitment, Market Dispatch
SDDP	Multistage Stochastic DP	Hydropower, Stochastic Dispatch
PySP	Progressive Hedging	Stochastic UC, Market Clearing
TIMES/MESSAGE	LP/MILP (Long-horizon)	Capacity Expansion, Policy

4 ML Enhancement

The computational challenges associated with the problem classes introduced above— namely integer nonlinear programs (INLP), mixed-integer nonlinear programs (MINLP), combinatorial set optimization (CSO), and dynamic programming (DP) formulations—are particularly pronounced in large-scale energy system applications such as non-convex AC Optimal Power Flow (AC-OPF), time-expanded Unit Commitment (UC), and multistage hydropower scheduling. As discussed in the preceding sections, these problems combine nonlinear continuous physics, discrete decision spaces with lattice or graph structure, and, in many cases, inter-temporal or stochastic coupling. In particular, unit commitment induces a temporal lattice structure through binary on/off decisions across time periods, while multistage hydropower operation is naturally expressed through dynamic or stochastic dynamic programming recursions. Optimal power flow problems, by contrast, are defined on the spatial graph of the transmission network. While classical solution methods exploit problem structure to restrict otherwise intractable enumeration, their computational cost can still grow rapidly with problem size and repeated solution requirements.

Throughout this work, we denote by $\mu \in \mathbb{R}^d$ the vector of exogenous parameters (e.g., weather data, price signals, demand profiles, renewable generation levels, water inflows, fuel prices, or network states) that characterize a specific optimization instance. Accordingly, the objective function $f(x;\mu)$ and the constraint functions $g(x;\mu)$ and $h(x;\mu)$ are understood to be parametrically dependent on μ.

In practical energy and agricultural system operation and planning, the same underlying mathematical formulations—such as the INLP and MINLP problems defined in (2) and (4), or the CSO formulation in (6)—are solved repeatedly under varying realizations of μ. Although each realization induces a distinct numerical instance, the algebraic structure of the feasible sets C_{in}, C_{mi}, and $\mathcal{F}$ remains unchanged. This repeated solution setting motivates the use of learning-based methods, which treat μ as an input feature vector while exploiting structural regularities across instances.

The diverse optimization paradigms reviewed in this work—ranging from exact mathematical programming solvers to learning-enhanced heuristics, reinforcement learning, and generative models—exhibit complementary strengths and limitations. Classical solvers provide strong feasibility and optimality guarantees but may struggle with scalability and repeated-solution demands. Learning-in-the-loop methods preserve solver correctness while accelerating critical algorithmic components in parametric settings. Reinforcement learning and generative approaches, by contrast, trade formal guarantees for flexibility and scalability in long-horizon or highly uncertain environments. Understanding these trade-offs is essential for selecting appropriate solution architectures in large-scale energy and agricultural applications.

This repeated-solution setting has motivated the emergence of a *learning-to-optimize* paradigm, in which ML and RL are employed to augment classical optimization algorithms rather than replace them [19]. From this perspective, learning methods are used to extract latent regularities across families of INLP, MINLP, and CSO instances—such as recurring active constraints, typical integer patterns, or frequently optimal subsets—that are not explicitly encoded in the mathematical formulation but nonetheless govern solver behavior.

Rather than predicting solutions in isolation, learning-to-optimize approaches intervene at specific stages of established algorithms, including warm-start initialization for nonlinear solvers, branching and node selection in branch-and-bound for mixed-integer problems, cut selection and bound tightening in relaxation-based methods, or policy approximation in multistage stochastic dynamic programming. By guiding these algorithmic components using information learned from previously solved instances with similar values of μ, learning-assisted methods aim to preserve the feasibility and optimality guarantees of the original INLP, MINLP, and CSO formulations, while significantly reducing the computational effort required to navigate large discrete spaces, complex nonlinearities, and long planning horizons.

In this sense, ML serves as a data-driven mechanism for uncovering and exploiting structural regularities that are implicit in the combinatorial and algebraic foundations of the optimization problems themselves, thereby complementing the classical emphasis on structure-driven algorithm design developed in this section.

4.1 Software with Learning in the Loop

The computational challenges associated with integer nonlinear programs (INLP), mixed-integer nonlinear programs (MINLP), and combinatorial set optimization (CSO) problems—particularly in large-scale energy system applications such as non-convex AC Optimal Power Flow (AC-OPF) and time-expanded Unit Commitment (UC)—have motivated the integration of ML components directly into classical optimization software. In this *learning-in-the-loop* paradigm, learning-based models augment specific stages of established solvers rather than replacing the underlying mathematical formulations or algorithms [19].

In practical operation and planning, the same INLP, MINLP, and CSO formulations defined in (2), (4), and (6) are solved repeatedly under varying exogenous conditions, denoted by μ, such as changes in demand, renewable generation, inflows, prices, or network states. While each realization of μ yields a distinct numerical instance, the algebraic structure of the feasible sets C_{in}, C_{mi}, and $\mathcal{F}$ remains unchanged. Learning-in-the-loop methods exploit this repetition to extract latent regularities that govern solver behavior.

Throughout this section, ϕ, ψ, and θ are used to denote the learnable weights and biases of the respective neural network architectures.

Neural warm-starting for nonlinear programming. Interior-point solvers for nonlinear programming, such as those used for AC-OPF in `MATPOWER` or `Ipopt`, generate a sequence of iterates $\{x^{(k)}\}$ converging to a Karush–Kuhn–Tucker (KKT) point. Their convergence rate depends critically on the proximity of the initial iterate $x^{(0)}$ to the optimal solution $x^\star$. In parametric settings where successive instances differ primarily through slowly varying μ, this sensitivity can be exploited via data-driven initialization.

The `Ipopt` solver, based on the implementation described in [44], utilizes a primal–dual interior-point filter line-search algorithm. This approach is particularly effective for large-scale nonlinear programs (NLPs) because it does not require a penalty parameter for the objective function, instead using a "filter" to ensure progress toward both feasibility and optimality. Its robustness in handling non-convexities has made it the de facto standard for solving the physics-heavy equations of AC-OPF in a local optimization context.

A neural network f_ϕ is trained offline to approximate the parametric solution mapping

$$x^\star(\mu) = \operatorname*{argmin}_x \{f(x;\mu) \mid g(x;\mu) = 0,\ h(x;\mu) \le 0\}.$$

At run time, the predicted solution

$$x^{(0)} = f_\phi(\mu)$$

is supplied as a warm start to the interior-point solver. Training data $\{(\mu_i, x_i^\star)\}_{i=1}^N$ are generated using high-fidelity solvers, and the loss function combines regression accuracy with feasibility penalties:

$$\mathcal{L}(\phi) = \mathbb{E}_\mu\Big[\|f_\phi(\mu) - x^\star(\mu)\|_2^2 + \lambda\|\max(0, h(f_\phi(\mu);\mu))\|_2^2 + \eta\|g(f_\phi(\mu);\mu)\|_2^2\Big].$$

Here, $\lambda > 0$ and $\eta > 0$ are penalty coefficients that balance regression accuracy against satisfaction of inequality and equality constraints, respectively. Such warm-starts significantly reduce iteration counts and factorization costs in AC-OPF solvers [5,19]. Moreover, ϕ represents the vector of learnable weights and biases of the neural network f_ϕ, while x denotes the optimization decision vector. The functions $g(x;\mu)$ and $h(x;\mu)$ represent the equality and inequality constraints of the optimization problem, respectively. The coefficients λ and η are strictly positive penalty hyperparameters that weight the relative importance of feasibility satisfaction against regression accuracy during training.

Active constraint screening and set reduction. In large-scale energy optimization problems, only a small subset of inequality constraints is active at optimality. Let $\mathcal{E}$ denote the full set of candidate inequality constraints. A classifier C_ψ is trained to estimate

$$C_\psi(\mu)_i \approx \Pr\big(h_i(x^\star(\mu);\mu) = 0\big), \quad i \in \mathcal{E}.$$

Given a threshold τ, the predicted active set

$$\mathcal{A}_{\text{pred}} = \{i \in \mathcal{E} \mid C_\psi(\mu)_i \geq \tau\}$$

is used to restrict the solver to a reduced constraint set, with lazy enforcement preserving feasibility; this means that constraints predicted to be inactive are initially omitted but are checked a posteriori and reintroduced if violations occur. This reduces problem dimension and accelerates NLP and MINLP solvers [19]. Here, i serves as an index for individual constraints within the candidate set $\mathcal{E}$, and ψ denotes the parameters of the classification network C_ψ. The threshold $\tau \in [0, 1]$ is a hyperparameter that determines the aggressiveness of the set reduction; higher values of τ result in smaller, more computationally efficient reduced sets at the risk of potentially excluding active constraints.

Learning-assisted branching and node selection for MIP. For mixed-integer problems such as UC, solved via branch-and-bound, the dominant cost arises from exploring the search tree $\mathcal{T}$. At each node $n \in \mathcal{T}$, a relaxation provides a lower bound L_n, while the incumbent solution yields an upper bound U. Learning-assisted approaches introduce a policy π_θ that maps node features $\phi(n)$—derived from fractional variables, dual information, and bound gaps—to branching or node-selection decisions,

$$\pi_\theta(\phi(n)) \in \mathcal{I}.$$

Here, $\mathcal{I}$ denotes the index set of integer decision variables eligible for branching at node n. Training uses data from solved instances, either via imitation of strong branching or RL. Classical safeguards remain intact, ensuring that global optimality guarantees of the MILP or MINLP formulation are preserved [19]. In this context, θ represents the parameters of the policy network π_θ, while the feature mapping $\phi(n)$ extracts node-specific information such as variable integrality gaps and dual values to inform the decision-making process.

Physics-guided neural networks for feasibility enhancement. Physics-guided neural networks embed power system constraints directly into learning objectives, producing near-feasible approximations for rapid screening or solver initialization. Let $F(x;\mu) = 0$ denote the AC power flow equations and $h(x;\mu) \leq 0$ operational limits. A neural network f_ϕ predicts $\hat{x} = f_\phi(\mu)$ and minimizes

$$\mathcal{L}(\phi) = \mathbb{E}_\mu\Big[\|f_\phi(\mu) - x^\star(\mu)\|_2^2 + \alpha\|F(f_\phi(\mu);\mu)\|_2^2 + \beta\|\max(0, h(f_\phi(\mu);\mu))\|_1\Big].$$

Such models improve feasibility and generalization relative to purely data-driven regressors, while remaining complementary to classical solvers [5, 19]. The parameters α and β serve as regularization weights that penalize violations of the physical power flow residuals and operational limits, respectively. By adjusting these weights, the training process can be tuned to prioritize physical consistency over pure data-driven regression.

Learning-Based Prediction and Confidence-Aware Solver Integration. Early learning-in-the-loop approaches for AC-OPF focused on directly predicting optimal operating points using supervised learning, with the explicit goal of accelerating repeated nonlinear solves. A representative example is the work of Fioretto et al. [14], which trains deep neural networks to approximate the parametric solution map

$$\mu \;\mapsto\; x^\star(\mu)$$

for AC-OPF instances. Rather than replacing the optimization problem, the learned predictor is used to provide high-quality initializations and feasibility screening for classical solvers.

Building on this idea, Chen and Zhang [10] introduce *control-confidence sets* to explicitly quantify the uncertainty of learning-based OPF predictions. In their framework, the neural network outputs both a candidate solution $\hat{x}(\mu)$ and a confidence region within which the true optimal solution is guaranteed to lie with high probability. Classical optimization solvers are then invoked only when the predicted confidence set fails to certify feasibility or optimality.

Mathematically, this approach augments the warm-start paradigm by learning a set-valued mapping

$$\mu \;\mapsto\; \mathcal{X}_{\text{conf}}(\mu) \quad \text{with} \quad x^\star(\mu) \in \mathcal{X}_{\text{conf}}(\mu)$$

with high probability. This guarantees that solver acceleration does not compromise feasibility, thereby preserving the correctness of the underlying INLP formulation while significantly reducing average solution time. These works exemplify how learning models can be embedded into solver pipelines in a provably safe manner, aligning closely with the learning-in-the-loop philosophy adopted in modern power system optimization software.

4.2 Learning to Optimize and Heuristic Acceleration

The integration of data-driven parameterization into the combinatorial structure of optimization problems in energy and agriculture can be formalized using the methodological framework of Bengio et al. [6]. In this framework, a family of optimization problems is modeled as a distribution $\mathcal{D}$ over instances. Each instance $\mathcal{I} \sim \mathcal{D}$ is characterized by a triplet $(f, \mathcal{C}, \mu)$, where f denotes the objective function, $\mathcal{C}$ the feasible set, and μ a vector of instance-specific parameters such as demand profiles or weather conditions.

Within this setting, statistical models are employed to approximate computationally expensive decision rules that arise repeatedly during the solution process. A prominent example is branching-variable selection in Branch-and-Bound (B&B) algorithms for mixed-integer optimization. At any node n of the search tree $\mathcal{T}$, the solver must select a branching action $a \in \mathcal{A}(n)$ from the set of fractional candidate variables. While Full Strong Branching (FSB) produces high-quality decisions by explicitly evaluating candidate branches, its computational cost is prohibitive for large-scale instances.

Heuristic acceleration formulates branching as a mapping $\pi_\theta : \Phi(n) \to a$, where $\Phi(n)$ denotes a feature representation of the node state, including local bounds, objective coefficients, and constraint activity. The parameters θ are determined by minimizing a loss function that approximates the expert FSB decision $a^\star(n)$:

$$\mathcal{L}(\theta) = \sum_{\mathcal{I} \sim \mathcal{D}} \sum_{n \in \mathcal{T}_\mathcal{I}} \mathrm{loss}(\pi_\theta(\Phi(n)), a^\star(n))\,.$$

Approximating this mapping using graph-based or feedforward function classes yields branching decisions that are comparable in quality to FSB while maintaining a computational cost similar to classical heuristics such as pseudocost branching. Since these components are embedded within the B&B framework, feasibility and global optimality guarantees of the solver are preserved. Such approaches have been applied successfully to mixed-integer structures arising in unit commitment and multi-period agricultural planning.

Beyond heuristic acceleration, recent work has developed a more formal treatment of parameter selection within iterative optimization algorithms. Ochs et al. [31] formulated algorithm parameter tuning as a bilevel optimization problem of the form

$$\min_{\theta \in \Theta} \mathbb{E}_{\mathcal{I} \sim \mathcal{D}}\big[\mathcal{L}_{\mathrm{outer}}(x_T(\theta; \mathcal{I}), \mathcal{I})\big] \quad \text{s.t.} \quad x_{t+1} = \mathcal{F}_\theta(x_t, \mathcal{I}),\ t \in [0 : T-1],$$

where $\mathcal{F}_\theta$ denotes the update mapping of an iterative optimization algorithm, x_t is the iterate at step t, and T is a fixed unrolling horizon. The outer loss $\mathcal{L}_{\mathrm{outer}}$ measures solution quality or convergence speed. By differentiating the unrolled dynamics $\{x_t\}_{t=0}^T$ with respect to θ, gradients or subgradients can be computed even when the underlying problem is nonsmooth, enabling systematic tuning of step sizes, preconditioners, or proximal parameters while preserving the structure of the original solver.

Subsequent work by Fadili, Sucker, and Ochs [12] established a PAC-Bayesian framework for analyzing parameterized optimization algorithms. In this setting, an algorithm is modeled as a random variable $\theta \sim Q$ drawn from a posterior distribution over a hypothesis space Θ. For bounded loss functions $\ell(\theta, \mathcal{I}) \in [0, 1]$, the expected population performance satisfies, with probability at least $1 - \delta$ over the draw of a training sample $\mathcal{S}$,

$$\mathbb{E}_{\theta \sim Q,\, \mathcal{I} \sim \mathcal{D}}[\ell(\theta, \mathcal{I})] \;\leq\; \mathbb{E}_{\theta \sim Q,\, \mathcal{I} \sim \mathcal{S}}[\ell(\theta, \mathcal{I})] + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log \frac{1}{\delta}}{2|\mathcal{S}|}},$$

where P is a prior distribution over Θ and $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. This bound characterizes the trade-off between empirical performance on the training set and generalization to unseen problem instances.

Related results further demonstrate that restricting the hypothesis space Θ to algorithm classes that satisfy structural invariances—such as equivariance under affine or rotational transformations—reduces effective model complexity and improves generalization across heterogeneous problem families [8]. Together, these contributions provide a rigorous foundation for incorporating data-driven parameter selection into optimization algorithms while maintaining stability and robustness.

4.3 Ensuring Feasibility via Differentiable Layers

A persistent challenge in applying RL to physical systems is ensuring that control actions satisfy hard operational and physical constraints. Following the framework introduced by Agrawal et al. [2], optimization problems can be embedded as differentiable layers within a policy network. This is achieved by viewing a constrained optimization problem as a mapping $f(\theta)$ from parameters θ to an optimal solution $x^\star$.

Specifically, consider a convex optimization problem in the form:

$$x^\star(\theta) = \underset{x}{\operatorname{argmin}} \quad f(x, \theta) \quad \text{s.t.} \quad g(x, \theta) \leq 0, \quad h(x, \theta) = 0 \tag{11}$$

Using the theory of disciplined parametrized programming (DPP), these layers expose the solution map as a differentiable operator. During the backward pass of the neural network, the gradient of the loss $\mathcal{L}$ with respect to the input parameters θ is computed by differentiating through the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial x^\star} \frac{\partial x^\star(\theta)}{\partial \theta} \tag{12}$$

The term $\frac{\partial x^\star}{\partial \theta}$ is obtained via implicit differentiation of the KKT conditions at the optimal point. This construction enables gradients to be propagated through the optimization solver during training, thereby guiding the learning process toward feasible decisions.

In effect, the RL agent learns a policy $\pi_\phi(s)$ that outputs raw parameters θ. These parameters are then passed through the differentiable layer to produce an action $a = x^\star(\theta)$ that is guaranteed to be a projection onto the feasible set defined by system constraints, such as power flow limits in energy networks or water balance and capacity constraints in irrigation systems. Consequently, feasibility with respect to the underlying physical model is enforced by construction rather than through penalties or post hoc correction.

4.4 Application: Feasibility-Optimized AC-OPF

In the specific context of power systems, Pan et al. [34] demonstrate the utility of a "prediction-and-reconstruction" approach known as *DeepOPF*. To solve the non-convex AC-OPF problem efficiently, the framework exploits the physical structure of the power flow equations to reduce the dimensionality of the learning task.

Mathematically, let the optimization vector x be partitioned into independent variables x_{ind} (e.g., active power generations P_g and voltage magnitudes $|V_g|$ at generator buses) and dependent variables x_{dep} (e.g., reactive power generations Q_g and voltage angles θ). The AC-OPF equality constraints, representing the nodal power balance, can be viewed as a mapping $x_{\text{dep}} = \Psi(x_{\text{ind}}, \mu)$. The DeepOPF approach proceeds as follows:

1. **Prediction:** A deep neural network f_ϕ is trained to predict only the independent variables:
$$\hat{x}_{\text{ind}} = f_\phi(\mu) \tag{13}$$
2. **Reconstruction:** The dependent variables $\hat{x}_{\text{dep}}$ are obtained by solving the power flow equations $\Psi(\hat{x}_{\text{ind}}, \mu) = 0$ using traditional numerical methods (e.g., Newton-Raphson).

This hybrid approach ensures that the high-dimensional power-balance equality constraints are strictly satisfied by construction. Furthermore, by predicting only x_{ind}, the number of output neurons in the DNN is significantly reduced, which decreases training complexity and the volume of required data. To handle the remaining inequality constraints (e.g., thermal limits on lines), the training loss function incorporates a penalty term based on a zero-order gradient estimation to adjust the predicted $\hat{x}_{\text{ind}}$ toward the feasible region:

$$\mathcal{L}(\phi) = \mathbb{E}_\mu \left[\|\hat{x}_{\text{ind}} - x^\star_{\text{ind}}\|^2 + \lambda \sum_j \max(0, h_j(\hat{x}, \mu)) \right] \tag{14}$$

where h_j represents the operational inequality constraints. This ensures that the model remains computationally efficient while providing high-quality, physically consistent solutions for real-time grid operations.

4.5 RL as Solution Technique

In contrast to learning-in-the-loop methods, RL is employed as an approximate solution technique for multi-stage and sequential energy optimization problems, such as hydropower reservoir operation and long-horizon market bidding. These problems are traditionally formulated using **Dynamic Programming (DP)** or **Stochastic Dual Dynamic Programming (SDDP)**, which rely on the recursive Bellman equation to find the optimal value function $V_t(s_t)$:

$$V_t(s_t) = \min_{a_t \in \mathcal{A}} \{r(s_t, a_t) + \gamma \mathbb{E}[V_{t+1}(s_{t+1})]\}. \tag{15}$$

While DP provides exact optimal policies, it suffers from the "curse of dimensionality" as the state space $\mathcal{S}$ grows. RL addresses this by using deep neural networks to approximate $V(s)$ or the policy $\pi(s)$, effectively acting as a *Neural DP* solver that scales to complex systems. RL formulates these problems as a Markov Decision Process (MDP) defined by the quintuple:

$$(\mathcal{S}, \mathcal{A}, p, r, \gamma),$$

where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, respectively. At any time t, the state $s_t \in \mathcal{S}$ encodes system conditions (e.g., reservoir levels, current demand), while the action $a_t \in \mathcal{A}$ represents a control decision (e.g., water discharge, power setpoints). The state transition probability kernel p defines the system dynamics, and r denotes the reward function capturing operational profit or cost.

In this MDP context, the state s_t is a composite vector $s_t = (x_t, \mu_t)$, where x_t represents internal system variables (e.g., current reservoir levels) and μ_t represents the exogenous parameters (e.g., forecast inflows or demand) previously defined in Section 4.1.

The discount factor $\gamma \in [0, 1)$ determines the present value of future rewards, ensuring the convergence of the infinite reward sum. In this context, the notation $a_t \mid s_t$ denotes that the action selected is conditioned on the observed state at time t, and $\pi_\theta(a_t|s_t)$ represents the stochastic policy parameterized by the learnable weights θ, which is optimized to maximize the expected cumulative reward. Policy-based methods parameterize a stochastic policy $\pi_\theta(a|s)$ and optimize

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right],$$

with gradients

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\, \hat{A}_t\right].$$

Here, $\hat{A}_t$ denotes an estimator of the advantage function, measuring the relative quality of action a_t compared to a baseline value at state s_t.

Unlike classical DP, which requires an exhaustive sweep of discretized state spaces, RL utilizes stochastic policies π_θ and entropy-regularized objectives to facilitate exploration of non-convex regions, significantly improving sample efficiency in high-dimensional hydropower systems.

A key advantage of RL is its ability to incorporate nonlinear and nonconvex physical relationships directly into the system dynamics. For example, hydropower production may follow

$$P = \eta(Q, H)\, \rho g H Q,$$

without linearization, where $\eta(Q, H)$ denotes turbine efficiency as a function of discharge Q and hydraulic head H, ρ is the density of water, and g is the gravitational acceleration constant. From a theoretical standpoint, RL policies approximate solutions to Bellman equations, trading optimality guarantees for scalability and modeling flexibility. Consequently, RL is often deployed in hybrid architectures that combine policy learning with optimization-based feasibility layers [19, 23].

Relation to Learning-Based OPF and Sequential Decision Policies. While RL is primarily used for sequential and multi-stage decision problems, its conceptual relationship to learning-based OPF methods is increasingly recognized. In particular, learning-based OPF predictors such as those proposed by Fioretto et al. [14] and Chen and Zhang [10] can be interpreted as approximations of one-step optimal control policies conditioned on the system state.

From this perspective, static AC-OPF can be viewed as a single-stage Markov Decision Process with deterministic transitions, whereas multi-period OPF, unit commitment, and hydropower scheduling naturally induce multi-stage MDPs. RL generalizes these approaches by learning policies

$$\pi_\theta(a_t \mid s_t)$$

that map system states directly to control actions over time, without explicitly solving a sequence of optimization problems at each stage.

However, unlike learning-in-the-loop OPF methods, RL does not generally provide feasibility or optimality guarantees for each action. As a result, recent research emphasizes hybrid architectures in which RL policies generate candidate actions that are subsequently filtered or corrected by optimization-based feasibility layers. In these hybrid architectures, the RL policy $\pi_\theta(s_t)$ acts as a proposal distribution. The candidate action $\hat{a}_t$ is passed to a projection operator $\Pi_{\mathcal{C}}(\hat{a}_t)$—often implemented as a differentiable layer as described in Section 4.3—which solves a local quadratic program to find the nearest feasible action $a_t \in \mathcal{C}$ before the reward $r(s_t, a_t)$ is calculated. This mirrors the role of confidence-aware learning in static OPF and reinforces the complementary relationship between RL and classical stochastic optimization techniques such as `SDDP`.

4.6 Role of Generative AI

Beyond solver acceleration and policy learning, generative artificial intelligence (AI) plays a complementary role in energy optimization workflows by supporting scenario simulation, model exploration, and solution proposal. Unlike learning-in-the-loop methods or RL policies, generative models are not embedded directly within the optimization algorithm. Instead, they operate at the interface between data, models, and decision-making, providing structured inputs that enhance stochastic optimization, robustness analysis, and human-in-the-loop design.

Scenario generation for stochastic optimization. Many energy optimization problems introduced earlier—such as stochastic unit commitment, hydropower scheduling, and energy market bidding—depend critically on exogenous uncertain parameters, collectively denoted by μ (e.g., demand trajectories, renewable generation, inflows, and prices). Classical stochastic optimization approaches assume that μ follows a known probability distribution or is represented through a finite scenario tree. Generative models provide a data-driven mechanism to approximate this distribution.

Formally, let $\mu \in \mathbb{R}^d$ denote the vector of uncertain parameters. A generative model parameterized by θ seeks to learn an approximation $\hat{p}_\theta(\mu)$ to the unknown data-generating distribution $p(\mu)$ from historical observations $\{\mu_i\}_{i=1}^N$. Sampling from $\hat{p}_\theta$ yields synthetic scenarios

$$\mu^{(s)} \sim \hat{p}_\theta(\mu), \quad \forall s \in [S],$$

which can be used as inputs to stochastic optimization problems such as `SDDP` or scenario-based MILP formulations. Compared to handcrafted scenario trees, generative models can capture complex temporal correlations, non-Gaussian behavior, and extreme events, thereby improving stress testing and robustness evaluation [19, 23].

Model and formulation proposal. Generative AI can also assist in proposing candidate optimization models, relaxations, or constraint formulations by learning from collections of previously solved problem instances. Let $\mathcal{M}$ denote a space of admissible optimization models or formulations (e.g., different OPF relaxations, constraint subsets, or approximation levels). Given historical pairs $\{(\mu_i, \mathcal{M}_i)\}$, where $\mathcal{M}_i$ denotes the formulation that performed well under conditions μ_i, a generative model can propose new candidates

$$\mathcal{M}^{\text{prop}} \sim q_\theta(\mathcal{M} \mid \mu),$$

which can then be evaluated and validated using classical solvers. This approach does not alter the mathematical structure of the optimization problem but accelerates model selection and exploration in large design spaces, particularly in planning and exploratory studies.

Solution proposal and human-in-the-loop decision support. Finally, generative models can be used to propose candidate solutions or operating points that serve as initial conditions or reference policies. For example, a generative model may learn an approximate distribution over feasible or near-optimal solutions,

$$x \sim q_\theta(x \mid \mu),$$

which can be used to initialize optimization solvers, guide scenario analysis, or support operator decision-making. Importantly, these proposals are always subject to verification and refinement through exact optimization or feasibility checks, ensuring that operational constraints and safety requirements are respected.

In all these roles, generative AI functions as a decision-support and model-discovery layer rather than a replacement for mathematical optimization. Its primary contribution lies in accelerating scenario exploration, enriching uncertainty representations, and facilitating interaction between domain experts and complex optimization software. When combined with learning-in-the-loop solvers and RL policies, generative AI further enhances the efficiency, robustness, and adaptability of large-scale energy optimization workflows [19, 23].

Generative Models for Renewable Scenario Construction. A central application of generative AI in energy optimization is the construction of realistic stochastic scenarios for uncertain renewable generation. Chen et al. [9] demonstrate that generative adversarial networks (GANs) can be trained to learn the joint spatiotemporal distribution of renewable power outputs directly from historical data, without relying on explicit probabilistic assumptions.

Formally, let μ_t denote a time-indexed vector of renewable generation across multiple locations. A generative model learns an implicit distribution $\hat{p}_\theta(\mu_{1:T})$ such that samples

$$(\mu_1^{(s)}, \dots, \mu_T^{(s)}) \sim \hat{p}_\theta$$

exhibit the same temporal dynamics, spatial correlations, and extreme-event behavior as historical observations. The integer d defines the dimensionality of the uncertain parameter space, while θ represents the parameters of the generative model $\hat{p}_\theta$. The index s denotes a specific realization out of the total number of sampled scenarios S, and the subscript $1:T$ in $\mu_{1:T}$ denotes the temporal sequence of parameters over a time horizon of T steps. These generated scenarios can be directly embedded into stochastic optimization frameworks, including stochastic unit commitment, `SDDP`-based hydropower scheduling, and robust planning models.

Compared to traditional scenario generation techniques based on parametric time-series models or copulas, generative models scale naturally to high-dimensional settings and capture nonlinear dependencies across space and time. Importantly, the generated scenarios remain external inputs to the optimization problem, ensuring that feasibility and optimality are still enforced by classical solvers. In this sense, generative AI enhances the expressiveness and realism of uncertainty modeling while preserving the mathematical rigor of the underlying INLP, MINLP, and CSO formulations.

5 Current Challenges, Limitations, and Future Directions

Despite substantial progress in optimization theory and solver technology, significant challenges remain in the practical deployment of OR methods for large-scale agriculture, energy, and water–energy systems. These challenges arise primarily from the interaction of nonlinear physical laws, discrete decision structures, inter-temporal coupling, and uncertainty, all of which are central features of the INLP, MINLP, and CSO formulations surveyed in this work.

5.1 Scalability and Computational Limitations

A fundamental challenge is scalability. While state-of-the-art solvers such as `SCIP`, `Ipopt`, `BARON`, and energy-specific tools including `MATPOWER` and `MOST` provide robust solution frameworks, their computational performance can deteriorate rapidly as problem size, time horizon, or combinatorial complexity increases. This limitation is particularly acute in applications that require repeated solution under varying exogenous conditions, such as real-time grid operation, rolling-horizon scheduling, and stochastic hydropower planning.

From a practical standpoint, these scalability limitations constrain the deployment of exact or high-fidelity optimization models in time-critical settings. As a result, operators often rely on simplified formulations, restricted horizons, or heuristic approximations, potentially sacrificing optimality or robustness in exchange for computational tractability.

5.2 Nonconvexity and Global Optimality

Nonconvexity remains another persistent obstacle. Problems such as AC Optimal Power Flow, hydropower generation with nonlinear efficiency characteristics, and reservoir dynamics induce feasible regions with multiple local optima. Although convex relaxations, decomposition techniques, and bound-tightening methods can provide useful approximations and bounds, guaranteeing global optimality or tight feasibility in large-scale settings remains computationally demanding. Balancing solution accuracy with tractability therefore continues to be a central concern in both theory and practice.

In many operational contexts, locally optimal solutions are acceptable or even unavoidable; however, the lack of global optimality certificates complicates performance assessment, sensitivity analysis, and policy evaluation. This limitation highlights an inherent trade-off between theoretical rigor and operational feasibility in complex energy and agricultural systems.

5.3 Inter-Temporal Coupling and Uncertainty

Inter-temporal and stochastic coupling further complicate solution procedures. Dynamic constraints linking decisions across time periods, as encountered in unit commitment, reservoir operation, and ancillary service provision, substantially expand the effective decision space. While approaches such as dynamic programming, stochastic dual dynamic programming, and scenario-based MILP formulations provide systematic solution frameworks, they are sensitive to dimensionality and scenario growth, necessitating careful model design and approximation.

Moreover, uncertainty modeling itself introduces additional limitations. Scenario sets may fail to capture rare but impactful events, while overly rich scenario representations can render problems computationally intractable. Establishing a principled balance between uncertainty representation, robustness, and computational efficiency remains an open methodological challenge.

5.4 Limitations of Learning-Enhanced Optimization

Within this context, learning-enhanced optimization offers several promising directions for future research, but also introduces new limitations that merit careful consideration. Learning-based components rely on historical data and implicit assumptions about stationarity and representativeness. When operating conditions deviate significantly from the training distribution—due to extreme weather, structural system changes, or rare contingencies—the performance benefits of learned heuristics may degrade, for example by producing ineffective warm-starts or misleading branching decisions that increase solver runtimes.

Furthermore, while many learning-enhanced methods preserve feasibility by embedding predictions within classical solver frameworks, their impact on convergence behavior, robustness, and worst-case performance is not yet fully understood. In particular, aggressive constraint screening or biased branching policies may interact nontrivially with solver heuristics, complicating validation, interpretability, and certification in safety-critical applications such as power system operation and water resource management.

5.5 Future Research Directions

Despite these limitations, learning-enhanced optimization offers several promising directions for future research. One important avenue is the deeper integration of learning-based components into solver architectures, including adaptive branching, constraint management, and bound-tightening strategies that exploit repeated problem structure without modifying the underlying mathematical models. Ensuring that such enhancements preserve feasibility, optimality guarantees, and solver robustness remains a key methodological requirement.

Another important direction concerns the principled treatment of uncertainty. Data-driven scenario construction and generative modeling approaches provide new opportunities to improve the realism and efficiency of stochastic optimization, particularly in renewable energy forecasting and hydrological inflow modeling. Future research should focus on establishing systematic links between scenario quality, computational performance, and decision robustness.

Finally, the interaction between approximate decision policies and exact optimization methods warrants further investigation. Hybrid architectures that combine sequential decision policies with optimization-based feasibility correction appear particularly well-suited for long-horizon and real-time applications. Developing theoretical guidelines and empirical benchmarks for when and how such hybrid methods should be deployed remains an open research question.

Overall, advancing optimization methods for agriculture and energy systems will require continued progress at the intersection of mathematical modeling, algorithm design, and data-driven enhancement. The formulations, solver frameworks, and application domains reviewed in this survey provide a structured foundation for addressing these challenges while maintaining the rigor and reliability required for operational deployment.

6 Conclusion

This work has presented a unified treatment of optimization problems arising in agriculture, energy, and water–energy systems through the lens of OR. By formalizing a broad class of applications within integer nonlinear, mixed-integer nonlinear, and combinatorial optimization frameworks, the paper highlights the common mathematical structures that underlie diverse real-world decision problems.

Building on this foundation, the study reviewed state-of-the-art optimization software and solver architectures, emphasizing their strengths and limitations when applied to large-scale, nonconvex, and inter-temporal problems. The analysis demonstrates that while classical solvers remain indispensable for enforcing feasibility and optimality, their computational performance can be significantly enhanced by exploiting problem repetition and structural regularities.

The learning-enhanced methodologies discussed in this work illustrate how data-driven techniques can be integrated into existing optimization pipelines to accelerate convergence, improve scalability, and support real-time decision making, without altering the underlying mathematical formulations. Together, these results establish a principled pathway for advancing OR methods in complex agriculture and energy systems, while preserving the rigor, interpretability, and reliability required for practical deployment.

Acknowledgements This work received funding from the National Centre for Energy II (TN02000025).

References

[1] Idriss Abdou and Mohamed Tkiouat. Unit commitment problem in electrical power system: A literature review. *International Journal of Electrical & Computer Engineering (2088-8708)*, 8(3), 2018.

[2] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[3] AgWorld (a Semios company). AgWorld farm management software. https://www.agworld.com, 2023. Accessed: 2025-03-01.

[4] Amr Ayad. *Stochastic Dynamic Programming Optimization Model for Operations Planning of a Multireservoir Hydroelectric System.* Master of ap-

plied science (m.a.sc.) thesis, University of British Columbia, Vancouver, Canada, January 2018.

[5] Kyri Baker. Emulating AC optimal power flow solvers with neural networks. *IEEE Transactions on Power Systems*, 37(6):4950–4953, 2022.

[6] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[7] Ksenia Bestuzheva et al. The SCIP optimization suite 8.0. *ACM Transactions on Mathematical Software*, 49(2):1–14, 2023.

[8] Camille Castera and Peter Ochs. From learning to optimize to learning optimization algorithms. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan, editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 1792–1800. PMLR, 03–05 May 2025.

[9] Yize Chen, Yishen Wang, Daniel Kirschen, and Baosen Zhang. Model-free renewable scenario generation using generative adversarial networks. *IEEE Transactions on Power Systems*, 36(3):1546–1556, 2021.

[10] Yize Chen and Baosen Zhang. Learning to solve AC-OPF via control-confidence sets. *IEEE Transactions on Power Systems*, 37(3):2213–2223, 2022.

[11] Carleton Coffrin, Russell Bent, Kaarthik Sundar, Yeesian Ng, and Miles Lubin. Powermodels.jl: An open-source framework for exploring power flow formulations. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–8, June 2018.

[12] Jalal M Fadili, Michael Sucker, and Peter Ochs. Learning-to-optimize with pac-bayesian guarantees: Theoretical considerations and practical implementation. *Journal of Machine Learning Research*, 2025.

[13] Carlo Filippi, Renata Mansini, and Elisa Stevanato. Mixed integer linear programming models for optimal crop selection. *Computers & Operations Research*, 81:26–39, 2017.

[14] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. Predicting AC optimal power flow outcomes with deep learning. *IEEE Transactions on Power Systems*, 35(4):3078–3089, 2020.

[15] Claudio Gambella, Bissan Ghaddar, and Joe Naoum-Sawaya. Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807–828, 2021.

[16] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2024.

[17] Anthony Horsley and Andrew J Wrobel. Efficiency rents of storage plants in peak-load pricing, ii: hydroelectricity. 1999.

[18] Mark Howells et al. OSeMOSYS: The open source energy modeling system. *Energy Policy*, 39(10):5850–5870, 2011.

[19] Xuguang Hu, Haiwang Zhong, Guanglun Zhang, Yiliu He, Xiaowei Wang, and Tianjiao Pu. Review of learning-assisted power system optimization. *CSEE Journal of Power and Energy Systems*, 7(2):221–231, 2021.

[20] Qi Huangfu and Julian A. J. Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018.

[21] IBM. *IBM ILOG CPLEX Optimization Studio*, 2021. https://www.ibm.com/products/ilog-cplex-optimization-studio.

[22] Sander Janssen and Martin K. van Ittersum. Assessing farm innovations and responses to policies: A review of bio-economic farm models. *Agricultural Systems*, 94(3):622–636, June 2007.

[23] Hooman Khaloie, Máté Dolányi, Jean-François Toubeau, and François Vallée. Review of machine learning techniques for optimal power flow. *Applied Energy*, 377:124430, 2025.

[24] Morteza Kimiaei, Vyacheslav Kungurtsev, and Brian Olimba. Machine learning algorithms for improving exact classical solvers in mixed integer continuous optimization, 2025.

[25] Richard Lincoln. PYPOWER: A Python port of MATPOWER, 2010.

[26] R Loulou, G Goldstein, A Kanudia, A Lettila, U Remme, and K Noble. Documentation for the times model part i-concepts and theory. *Energy Technology Systems Analysis Programme*, 2016.

[27] Richard Loulou, Uwe Remne, Amit Kanudia, Antti Lehtila, and Gary Goldstein. Documentation for the TIMES model: Part I–IV. Technical report, Energy Technology Systems Analysis Program (ETSAP), International Energy Agency (IEA), 2016. Part I: Theoretical Approach; Part II: Reference Manual; Part III: User Note; Part IV: GAMS Control Statements.

[28] Sabine Messner and Leo Schrattenholzer. Message–macro: linking an energy supply model with a macroeconomic module and solving it iteratively. *Energy*, 25(3):267–282, March 2000.

[29] Daniel K Molzahn, Ian A Hiskens, et al. A survey of relaxations and approximations of the power flow equations. *Foundations and Trends® in Electric Energy Systems*, 4(1-2):1–221, 2019.

[30] Harsha Nagarajan, Emre Yamangil, Russell Bent, Carleton Hijazi, and Pascal Van Hentenryck. Tightening QC relaxations of AC optimal power flow through improved linear convex envelopes. *IEEE Transactions on Power Systems*, 34(2):924–933, 2019.

[31] Peter Ochs, René Ranftl, Thomas Brox, and Thomas Pock. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 654–665. Springer, 2015.

[32] Pacific Northwest National Laboratory. GridLAB-D user's manual. Technical report, Pacific Northwest National Laboratory (PNNL), Richland, WA, 2018. Available at readthedocs.io.

[33] Narayana Prasad Padhy. Unit commitment-a bibliographical survey. *IEEE Transactions on power systems*, 19(2):1196–1205, 2004.

[34] Xianda Pan, Tianyang Zhao, Minghua Chen, and Steven Zhang. Deepopf: A feasibility-optimized deep learning approach for optimal power flow. *IEEE Transactions on Power Systems*, 37(1):679–690, 2022.

[35] Mario V. F. Pereira and Leontina M. V. G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1):359–375, May 1991.

[36] Dmitri Perekhodtsev and Lester B Lave. Efficient bidding for hydro power plants in markets for energy and ancillary services. 2006.

[37] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[38] Martin Smith et al. *CROPWAT: A computer program for irrigation planning and management*, volume 46. Food & Agriculture Org., 1992.

[39] Pasquale Steduto, Theodore C. Hsiao, Dirk Raes, and Elias Fereres. Aquacrop—the fao crop model to simulate yield response to water: I. concepts and underlying principles. *Agronomy Journal*, 101(3):426–437, May 2009.

[40] Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

[41] Leon Thurner, Alexander Schelermann, Jan Kaschek, Tanja Nettmann, and Martin Braun. pandapower—an open-source Python tool for convenient modeling and analysis of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, Nov 2018.

[42] Trimble Inc. Trimble agriculture software solutions. https://agriculture.trimble.com, 2023. Accessed: 2025-03-01.

[43] Pascal Van Hentenryck. Machine learning for optimal power flows. page 62–82, October 2021.

[44] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[45] H. Wang, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas. On computational issues of market-based optimal power flow. *IEEE Transactions on Power Systems*, 22(3):1185–1193, 2007.

[46] Jean-Paul Watson, David L. Woodruff, and William E. Hart. Pysp: modeling and solving stochastic programs in python. *Mathematical Programming Computation*, 4(2):109–149, March 2012.

[47] Yafei Yang and Lei Wu. Machine learning approaches to the unit commitment problem: Current trends, emerging challenges, and new strategies. *Electric Power Systems Research*, 189:106741, 2021.

[48] R. D. Zimmerman and C. E. Murillo-Sánchez. *MATPOWER Optimal Scheduling Tool (MOST) User's Manual, Version 1.3.1*, 2025. [Online]. Available: https://matpower.org/docs/MOST-manual-1.3.1.pdf.

[49] Ray D. Zimmerman, Carlos E. Murillo-Sánchez, and Robert J. Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.