On the Use of AI Planning for Water Management of the Red River Basin in Vietnam

Anonymous submission

Abstract

The management of reservoir-based hydrological systems is challenging due to complex dynamics and strong interdependencies. Traditional rule-based policies, derived from hydrological models, expert experience, and regulatory standards, are often too rigid, leading to inefficient use of water resources and limited ability to react to unexpected events such as off-season floods. We investigate how AI planning can support water management through the lens of numeric planning. The study focuses on the Hoa Binh reservoir, a key component of the Red River Basin in Vietnam, under hydrologically challenging dry-season conditions. The proposed formulation captures the complex system dynamics by defining actions that directly control the amount of water released on a daily basis. We first provide a fully domain-independent planning setting that makes no use of problem-specific information and observe that planners struggle to capture storage-dependent constraints, cumulative flows, and downstream interactions. Motivated by these limitations, we then integrate problem-specific structure and control knowledge into the domain-independent formulation to guide the planner toward higher-quality operating policies. Results indicate that planning can efficiently explore the decision space and generate plans of competitive quality compared to historical operation while achieving stricter adherence to operational constraints, particularly when enhanced with control knowledge.

Introduction

Large bodies of water play a pivotal role in ensuring the availability of water, energy, and food for society. Whether natural or human-made, these reservoirs serve as vital storage and distribution hubs for various purposes such as irrigation, hydropower generation, flood protection, and drinking water supply (Castelletti, Pianosi, and Soncini-Sessa 2008). However, their capacity often falls short of meeting the increasing demands of an ever-growing population and rapid urbanisation, exacerbating the challenge. The World Water Development Report of the UN highlights a concerning trend: water usage has been steadily increasing at a rate of approximately 1% annually since the 1980s, with projections suggesting this trajectory will persist until 2050 (Uhlenbrook, Connor et al. 2019). Adding to this complexity is the intensifying variability in precipitation patterns and the increased frequency and intensity of droughts attributed to climate change.

Traditional water management systems leverage rule-based policies derived from hydrological models and management rules based mainly on past experience and imposed standards. Yet, their inflexible nature often leads to suboptimal utilisation of water resources and an inability to adapt to unforeseen climatic events such as off-season floods. Consequently, there is a pressing need for more efficient and adaptive water management strategies, particularly in regions vulnerable to water stress, such as the Global South (Hien et al. 2020). In this study, we explicitly concentrate on the dry season (September–May) in Vietnam, the period with the tightest minimum-flow requirements and the lowest inflows. All planning instances, constraints, and evaluations are constructed exclusively for this seasonal window. The dry season scenario provides a challenging, yet highly relevant testbed for evaluating the feasibility of planning-based gate control. To address the limitations of fixed rule-based operation, we investigate AI planning for deriving operational gate-control policies by searching over feasible release schedules under explicit constraints and objectives. This choice is motivated by prior success of AI planning in structurally similar control domains, including urban traffic control and power-system voltage control (McCluskey and Vallati 2017; Vallati et al. 2016; Piacentini et al. 2015).

This paper is primarily a case study of applying an off-the-shelf numeric planner to a real-world reservoir operation problem, with domain-specific modelling choices together with iterative schemes aimed at improving plan quality in an anytime fashion. We focus on the daily operation of the Hoa Binh hydropower reservoir in the Red River Basin in Vietnam and build on a previously validated PDDL-based simulation model of the basin (Anonymous 2025). Compared to that simulation model, our contribution lies in: (i) formulating a control-oriented planning problem for the Hoa Binh reservoir; (ii) comparing a domain-independent search approach with a domain-dependent formulation based on iterative requirement tightening; and (iii) providing an empirical analysis of the behaviour of these approaches against the historical decisions of expert operators.

The paper is organised as follows. We first present the case study, introduce the necessary planning background, and describe our planning-based approach to water-resources management through automated planning. We then report the experimental analysis and discuss related

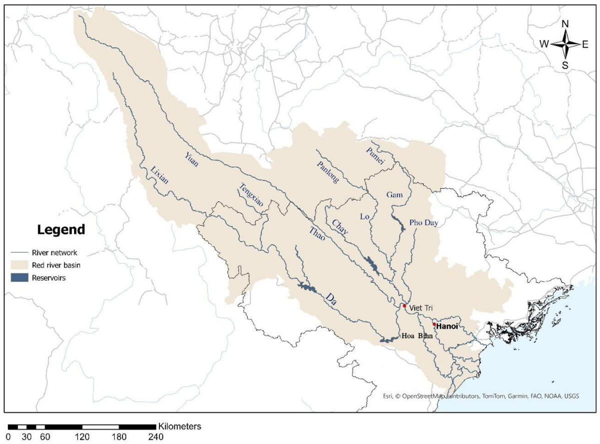

Figure 1: The Red River Basin in its entirety: from its springs in China to its mouth in the Gulf of Tonkin, Vietnam

work. Finally, we conclude the paper and outline directions for future developments.

Background

In this section we provide the general water management problem, and the numeric planning formalism that we use to tackle it.

Water Management in the Red River Basin

The Red River Basin in northern Vietnam (shown in Figure 1) is fed by three main tributaries (Da, Thao, and Lo) and flows through densely populated areas, including Hanoi. Hoa Binh, located on the Da River, is the largest hydropower reservoir in the basin and plays a central role in flood control and energy production. In this paper we focus on its daily operation over a dry-season horizon, when inflows are low and minimum-flow requirements are most constraining. At each day, the main decision is the release from the reservoir, which affects downstream flows and water levels at a set of control stations along the river network (e.g., Son Tay and Hanoi). Inflows from upstream sub-basins and tributaries are treated as exogenous, based on historical or forecast data. The state of the system includes the reservoir storage and water level, the flows and levels at river nodes, and quantities related to hydropower production.

The physical behaviour of the system follows standard hydrological and hydraulic relationships. Reservoir storage evolves according to a mass-balance equation of the form

$$S_{t+1} = S_t + I_t - R_t - L_t, \tag{1}$$

where S_t is storage at day t, I_t is the total inflow, R_t is the controlled release, and L_t accounts for other losses. Storage is linked to the reservoir water level by a level–volume curve, and power production depends on the release through turbines and the hydraulic head between the reservoir and the downstream reach. Flows are routed along the river network to update downstream discharges and water levels at the control stations. The detailed equations, parameter values, and calibration are inherited from a validated simulation model and are described in (Anonymous 2025).

Numeric Planning Background

A numeric planning problem is defined as a tuple $\langle F, X, A, I, G\rangle$, where F is a set of propositional variables, X a set of numeric variables, A a set of actions, I an initial state, and G a goal formula. States assign values to all variables; actions have preconditions and effects, are applicable when their preconditions hold, and update the state when applied according to the state transition induced by their effect. Solving the problem means finding a sequence of applicable actions that leads from I to a state satisfying G, possibly optimising a given metric.

Numeric planning problems are typically expressed in PDDL Level 2 (Fox and Long 2003), which supports parametric definitions of actions, facts, and fluents. Ground actions are obtained by instantiating parameters with objects, and full details on syntax, semantics, and grounding are given in Fox and Long (2003).

Water Management through Numeric Planning

We build on the validated PDDL2.1, planning-based simulator of (Anonymous 2025), but restrict it to the Hoa Binh reservoir with a daily discretisation (while preserving the essential hydrological couplings) and by using automated planning to generate feasible, high-quality release schedules and support scenario exploration through systematic re-planning under varying constraints and demand profiles.

We in particular exploit a planning factored and compact representation to model the system dynamics and encode the relevant operational constraints. In our control problem, at each day t, we need to decide the releases through turbines, bottom outlets, and spillways. The system state includes the reservoir storage and water level, as well as downstream flows and levels as computed by the underlying hydrological model. This setting captures the essential short-term operational lever available to reservoir operators while abstracting from the full basin network considered in the original model. Our planning problem is specified by four components: $\mathcal{S}$ contains storage, release, hydropower, and downstream-level variables; $\mathcal{A}$ comprises the open gate-related operators; $\mathcal{T}$ encodes the hydrological transitions; $\mathcal{C}$ imposes reservoir, downstream, and operational constraints. Moreover, our planning problem keeps track of the total amount of energy produced $\mathcal{J} = \sum_{t=1}^{T} E_t$, where E_t denotes the energy produced at day t. In order to maximise this value, we will then use the planning formulation in an iterative schema that will be described later. But first, we describe the main constraints that have to be kept in check, and then how this formulation can be used within an iterative schema that is aimed at incrementally finding plans that improve on the amount of energy produced at the end. In particular, we have explicit operational and safety constraints on reservoir storage and levels, turbine and outlet capacities, downstream flood thresholds, and minimum-flow requirements. These constraints are encoded as action preconditions, with potential violations tracked by numeric fluents for post-hoc assessment. Our formulation builds on the validated PDDL2.1 simulation model of (Anonymous 2025), but restricts the representation to a single reservoir and a simplified daily time structure, preserving the key hydrological dependencies (storage dynamics, downstream routing, and energy computation); energy production E_t depends on turbine releases and the hydraulic head derived from storage, with all coefficients provided as fixed numeric inputs in each instance.

Below, we summarise the key ingredients of our encoding. Then we use this as a foundation for the domain-independent and the domain-dependent search strategies reported subsequently.

Propositional Variables

We use PDDL2.1 objects of type `day` and a small set of propositional predicates. The predicate `active(?day)` marks as active the current search state day `?day`, while `next(?day1 ?day2)` encodes the temporal succession between days. Additional Boolean flags, such as `gates_opened` and `network_updated`, are used to enforce the intended sequence of the operators `open_gate`, `update_state_network`, and `advance_day`. In this simplified formulation, we use propositional variables to track the active day and the phase of the daily operational cycle, rather than representing the full river network.

Numeric Fluents

Numeric fluents are derived from the same hydrological relationships used in (Anonymous 2025), but restricted to a single reservoir. We devise a PDDL2.1 object `hoa_binh` to model the dam, and define variables such as `storage(?dam)`, `release_bottom(?dam)`, `release_spillways(?dam)`, and `release(?dam)` to represent current planning-state storage and releases through bottom outlets, spillways, and turbines of the dam. The fluents `hydropower(?dam)` and `total_production(?dam)` capture, respectively, current planning-state daily and cumulative energy production.

Coefficients for the storage–level curve and the hydropower formula are represented as numeric fluents and are taken from the validated hydrological model. Releases are discretised as follows: turbine gates in half-gate steps (145.7 m^3/s), bottom outlets in full-gate steps (1833 m^3/s), and spillways in full-gate steps (2350 m^3/s). We encode an operational priority in which turbine gates are opened first, bottom outlets are used only when turbine capacity is insufficient, and spillways only in extreme conditions; this discretisation and priority scheme keep the branching factor manageable over a daily horizon.

Operators

We use three specialised operators representing the daily operational cycle:

- `open_gate(?dam)`,`open_bottom_gate(?dam)`, `open_spillways_gate(?dam)` increase the current search state releases for dam `?dam` by a fixed amount through turbines, bottom outlets, and spillways respectively subject to capacity limits and operational priorities.
- `update_state_network(?day)` updates storage, level, and hydropower at day `?day` using the mass-balance equation and the physical relationships, and accumulates total production.
- `advance_day(?day1 ?day2)` moves from day `?day1` to `?day2`, resets controllable releases, and activates the inputs for the next day.

These operators compress many low-level transitions of the original model into three structured steps, reducing the search space and focusing on the key control steps needed for the daily reservoir operations. Each `open_gate` operator is applicable only when the active day is t and the corresponding release is below its capacity limit. Multiple applications are allowed within the same day to reach a feasible release. `advance_day` resets releases so that the next day's decisions are taken afresh. Roughly, the task for the planner is to decide which and how many gates need to be opened. The other actions are instead mandatory and only serve the purpose of moving time forward.

In order to make the planning task more effective and to enable computationally more expensive techniques such as effect abstraction (EA) (Li et al. 2018), we add a precondition to the `open_gate` operators that limits their applicability to a predefined set of *control days*. EA is a technique developed in the ENHSP planning system (Scala et al. 2020) aimed at improving heuristic estimates for numeric planning problems with linear (but non-simple) numeric effects. A numeric effect is *simple* if it interacts only with linear conditions and is *constant-additive*, i.e., it can be expressed as $x \mathrel{+}= k$ with $k \in \mathcal{Q}$. Problems featuring only simple numeric effects can be relaxed more effectively by state-of-the-art numeric planning heuristics. EA therefore targets problems with linear numeric effects of the form $x \mathrel{+}= exp$, where exp is a numeric expression, and transforms them into an equivalent formulation that only contains simple numeric effects. This is achieved by partitioning the effect into a set of intervals, each associated with a condition, and by generating a corresponding action for each interval. Increasing the number of intervals typically improves heuristic accuracy, at the cost of higher computational overhead. We will experimentally study this accuracy–cost trade-off. Restricting `open_gate` to control days also reduces the number of applicable operator instances, helping amortize the additional computational cost introduced by EA.

Initial State and Goal

In the initial state we set `storage(hoa_binh)` to the initial storage s_1, initialise all input inflows for each day, and assign the coefficients of the physical equations. The first day `day_1` is marked as active by asserting `active(day_1)`, and the predicates `next(?day1 ?day2)` define the sequence of days over the planning horizon. The goal is to reach the last day `day_T` after completing the daily cycle, typically requiring `active(day_T)` and, in the optimisation-oriented formulation that we say later, additional constraints on `total_production(hoa_binh)`.

Iterative Requirement Refinement

In order to maximise the amount of energy produced at the end, we adopt an iterative requirement refinement strategy that repeatedly re-solves progressively stricter instances, using the current best plan to guide the refinement. Algorithms 1 and 2 describe our anytime wrapper around a numeric planner. It starts by computing a feasible baseline plan for the original instance and then iteratively re-solves progressively stricter instances derived from the current best plan. Each iteration modifies the planning problem by refining its requirements—either by raising a global target (domain-independent refinement) or by selectively strengthening domain-specific constraints (domain-dependent refinement). The intent is to "pull" the planner toward better solutions while maintaining feasibility whenever possible. The procedure returns the best feasible plan found within a given time budget and can be interrupted at any moment, always providing the best valid policy found so far.

The *Domain-Independent Iterative* (DII) method keeps the model generic and improves plan quality by progressively tightening a global energy requirement across multiple planner calls. We do so by enforcing in the goal that at the next iteration the planner has to find a solution which has a total amount of energy larger than what found before.

The *Domain-Dependent Iterative* (DDI) method exploits control knowledge expressing an indication on the amount of energy production day by day of the current best plan $best$. For each day t with insufficient production we update the minimum release (`(min_release hoa_binh ?d)` variable of Π') by

$$R_t^{\min} = R_t + \alpha_t \cdot (R_t^{\mathrm{d}} - R_t), \qquad (2)$$

where R_t^{d} is the release required to meet the energy demand on day t, R_t is the water release of $best$ at day t, and α_t is a scaling factor controlling the step size of the adjustment[1].

For each day t, in the DDI we encode the updated minimum release $R_t^{\min}$ directly in the planning state by initialising the controlled release for that day to $R_t^{\min}$. The search is carried over as follows: the planner may apply additional `open_gate` actions to raise the release R_t in discrete steps beyond $R_t^{\min}$ when this is compatible with the constraints and improves energy production. After each such action, `update_state_network` updates storage, downstream levels, and energy, and checks all operational constraints for day t. If a violation occurs, the search backtracks and may further adjust R_t. Once `update_state_network` produces a feasible state, the planner applies `advance_day` to move to $t{+}1$ and repeats the same pattern.

This search behaviour, implemented entirely within the planner, ensures that each DDI iteration increases releases only on days where production was previously insufficient to satisfy the energy demand. By enforcing feasibility locally through the interaction of `open_gate` actions and

[1]In our experiments, α_t decreases linearly from 0.05 for $t = 0$ to 0.01 at the end.

```
    :precondition (and
  (active ?d)
  (not (evaluated_network))
   (<= (min_release hoa_binh ?d) (release hoa_binh))
)
```

Figure 2: PDDL encoding of the precondition of the minimum release.

`update_state_network`, the planner explores a structured and much smaller search space than in an unconstrained setting, while still reflecting global effects such as storage dynamics and downstream propagation.

Intuitively, increasing the minimum releases on under-producing days shifts water towards configurations with higher hydraulic head (derived from reservoir storage/water level) and greater turbine use, biasing the search toward higher-production plans without introducing a global optimisation constraint. This mechanism is encoded in the planning model by tightening the precondition on the minimum release. A simplified example is shown in Figure 2.

Specifically, Algorithm 1 takes as input a planning instance Π (domain and problem, including the optimisation metric) and a global time budget T. It begins by invoking the planner on the original instance (step 1), to obtain an initial feasible plan that can be used immediately as an operational policy. This baseline plan is stored as the current best solution (step 2) representing the best feasible policy found so far.

The algorithm then enters an anytime refinement phase. As long as time remains within T and the instance can still be meaningfully tightened (step 3), it repeatedly constructs a refined version of the planning problem by strengthening requirements based on the current plan Π' (step 5). The purpose of REFINEREQUIREMENTS is to make the next planning call slightly more demanding in a way that promotes improved solutions and does using two complementary planning approaches.

Regardless of whether the iterative schema used the DII or DDI, after building the refined instance Π', the algorithm calls the planner again (step 5), and evaluates the resulting candidate plan. If π' is feasible under the refined requirements and strictly improves the optimisation metric (step 6), $best$ is updated (step 7) and the refinement loop continues. Otherwise, the loop is terminated early, since further refinements would only make the problem more restrictive and are unlikely to yield improvements within the remaining budget. Finally, the algorithm returns $best$, i.e., the best feasible plan discovered within time T. Because the best-so-far solution is always maintained, the procedure is anytime: it can be interrupted at any moment while still returning a valid operational policy.

Experimental Analysis

We empirically analyse how heuristic search behaves on a set of real numeric planning instances derived from historical operation of the Hoa Binh reservoir. We built our planning problems using the available historical data and defined 11 problems in total, one for each year from 2001 to 2011.

Algorithm 1: Anytime refinement wrapper around a numeric planner

Require: Planning instance Π; time budget T; mode $m \in \{\mathrm{DII}, \mathrm{DDI}\}$
Ensure: Best feasible plan found within T
$\pi \leftarrow$ PLANNER(Π) {baseline feasible plan}
$best \leftarrow \pi$
while TIME() $< T$ **and** CANREFINE(Π, $best$, m) **do**
  $\Pi' \leftarrow$ REFINEREQUIREMENTS($\Pi, best, m$)
  $\pi' \leftarrow$ PLANNER(Π')
  if FEASIBLE(π') **and** IMPROVES(π', $best$) **then**
    $best \leftarrow \pi'$
  else
    break
  end if
end while
return $best$

Algorithm 2: REFINEREQUIREMENTS

Require: Planning instance Π; current $best$ plan; mode m
Ensure: Refined instance Π'
$\Pi' \leftarrow \Pi$
if $m =$ DII **then**
  Increase a global target/bound in Π' using $best$ (e.g., raise minimum required energy)
else
  Identify underperforming periods from $best$ (e.g., selected control days)
  Selectively tighten structured constraints in Π' (e.g., raise minimum releases on those days)
end if
return Π'

We used the 2001–2011 dry-season dataset because it aligns with the validated simulator and involves fewer constraints related to data sharing and anonymization, since more recent operational data are considered sensitive by the Vietnamese authorities. Each problem follows the structure described in the previous section.

To solve the planning problems, we use the ENHSP [2], a PDDL automated planning system (Scala et al. 2016). Unless otherwise stated, all experiments use ENHSP configured with greedy best-first search and the additive heuristic h_{add}. Experiments run on a machine with 24 Intel(R) Xeon(R) CPU E5-2620 0 2.00GHz and 125GB of RAM.

We evaluate our two approaches: the DII approach solved with off-the-shelf planners and standard numeric heuristics, and the DDI that applies the custom search schema described above. For the DII approach we run ENHSP with Effect Abstraction (EA), varying the EA parameter between 1 and 4 and the control days to assess their impact on search performance. As baselines for the comparison, we use the first solution of the DDI approach (corresponding to the minimal releases that satisfy the constraints), and the historical solutions computed by experts. As performance metrics, we consider cumulative energy production over the horizon, the number and magnitude of constraint violations (e.g., exceedance of flood thresholds and violations of minimum-flow constraints), and planning time.

[2] https://sites.google.com/view/enhsp/

Table 1: Sensitivity of the DII refinement to Effect Abstraction (EA) precision and the number of control days. Values report the best final cumulative production (MWh) aggregated over all years (higher is better) excluding 2003 and 2005.

Control days	EA=1	EA=2	EA=3	EA=4
45	4708081	4709976	**4948109**	4711192
60	4753607	4755885	**4998599**	4796819
82	4758186	4762284	**5002408**	4761616
122	4537273	4534652	**5142595**	4770893

Search behaviour.

We first study the sensitivity of the DII approach to EA precision and to the number of control days (Table 1), measured in terms of the best final cumulative production aggregated over all years except 2003 and 2005. These two years are excluded because all EA configurations encountered difficulties in finding solutions for them. The best-performing configurations are obtained with a moderate EA precision (EA=3), indicating a good compromise between heuristic informativeness and computational cost. By contrast, increasing the precision further (EA=4) often worsens performance, suggesting that more expensive relaxations are not always beneficial when delayed numeric interactions are predominant. Likewise, increasing the number of control days can improve solution quality by enabling finer adaptation, but only up to a certain point; beyond that, the larger branching space can impair the ability of the heuristic search to progress.

In terms of coverage, DII (with $EA = 3$ and 45 control days) times out on 2 out of 11 years (2003 and 2005 in Table 2) even with a 4-hour limit, illustrating that domain-independent guidance can be brittle on some instances. In contrast, the domain-dependent operator model plus the DDI wrapper finds and refines a feasible plan for every year within the 1-hour limit, suggesting that the injected structure reduces pathological search plateaus.

Plan quality.

Both incremental methods improve the initial feasible solutions, leading to increased energy production (Fig. 3). Across the considered years, the best DDI and (when successful) DII plans are generally close to historical production, while satisfying all modelled constraints that historical operations sometimes violate. This is significant because the expert solutions often rely on manual, case-specific adjustments and accept violations (discussed next).

Constraint violations in historical solutions.

Finally, we analyse the behaviour of the resulting policies with respect to critical events such as flood peaks and dry

Table 2: Final dry-season production (MWh) for the minimum-release baseline (MinRel), the best DII configuration we found (EA=3 with 45 control days), the best DDI plan, and the historical operation. "–" indicates that DII did not return a plan within the time limit for that year.

Year	MinRel	DII	DDI	Historical
2001	4015076	4531269	4715532	**4980990**
2002	3685244	4439423	4601111	**4687205**
2003	3388685	–	3831180	**3975801**
2004	3173931	3581183	**3724350**	3692424
2005	4158921	–	**4310098**	4233076
2006	3324348	4153659	4136276	**4224928**
2007	3478161	4247932	4365892	**4390841**
2008	4645545	5142595	5342152	**5479323**
2009	2299380	3130733	**3247572**	3245259
2010	3108997	3453692	3328938	**3518277**
2011	3899850	3688235	4415779	**4625165**

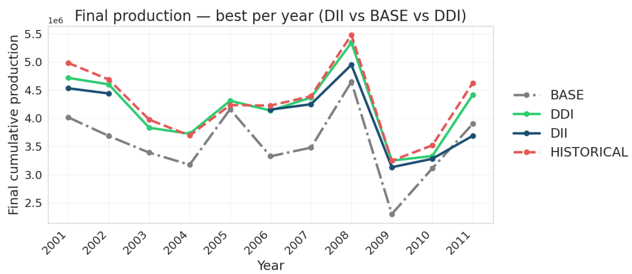

Figure 3: Comparison of final dry-season cumulative production (MWh) for the Minimum-release baseline, DDI, DII, and Historical operations.

periods. We compare our methods with the provided historical data, which reflect the decisions made by expert dam operators. However, these operational decisions are not consistent with the modeling assumptions adopted in this work, as dam operators may deliberately accept some constraint violations. Indeed, several constraints of interest in our problem are, in fact, violated in the historical records.

The final years of the period considered show a noticeable increase in the number of violations (see Fig. 4). The most significant rise concerns the water level in Hanoi, where the number of minimum-level violations grows sharply. This trend may be explained by particularly severe dry seasons

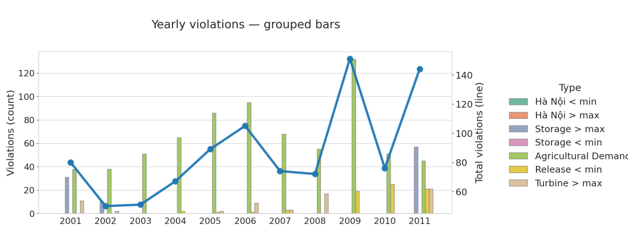

Figure 4: Violations count per year of the historical solutions.

Table 3: Violations count by month (season Sep–May)

Constraint	**Sep**	**Oct**	**Nov**	**Dec**	**Jan**	**Feb**	**Mar**	**Apr**	**May**
Agricultural demand	0	0	0	0	105	**252**	105	9	1
Hanoi $<$ min	0	0	3	19	37	48	**55**	32	0
Hanoi $>$ max	0	0	0	0	0	0	0	0	0
Reservoir $< 80\,\mathrm{m}$	0	0	0	0	0	0	0	0	**10**
Reservoir $> 117\,\mathrm{m}$	0	4	**12**	0	1	0	0	0	0
Release $< 214\,\mathrm{m}^3/\mathrm{s}$	1	8	5	1	7	**32**	12	4	2
Turbine $> 2360\,\mathrm{m}^3/\mathrm{s}$	0	1	0	0	0	0	0	0	**26**

during those years, but also by morphological changes in the riverbed. In fact, around that period a new upstream hydropower plant was constructed, which likely altered sediment transport dynamics and consequently affected the downstream riverbed configuration and flow conditions.

The developed models and incremental methods were also compared with the provided historical data, which reflect the decisions made by expert dam operators. However, these operational decisions are not always fully consistent with the modelling assumptions adopted in this work: several constraints of interest in our problem are, in fact, violated in the historical records.

As shown in Table 3, the agricultural demand considered in our model is often not satisfied during the months from January to March, which correspond to the most critical phase of the dry season. Another frequently violated constraint is the water level in Hanoi: during the central months of the dry season, the level often drops below the minimum threshold required for navigability, resulting in potential disruptions to transportation and rural activities in the area.

Related Work

The most used technique for water resource management is Stochastic Dynamic Programming (SDP), which has been predominantly used since the 1960s because of its capacity of preserving a realistic problem structure thanks to loose constraints on systems representation. However SDP has three relevant limitations that, considering the emerging challenges in water system operations, makes it partially unfit to deal with the newer problems. The first challenge of dealing with high-dimensional systems, *curse of dimensionality*, places practical limits on the complexity of systems that can be effectively modelled and solved. This makes computationally infeasible handle systems with more than two or three reservoirs. The second challenge, known as the *curse of modeling*, demands that all variables included in the operational policy must have dynamic models describing their behavior. This necessity introduces additional state variables, compounding the complexity of the problem. Furthermore, the third curse, *curse of multiple objectives*, imposes restrictions on the number of objective functions, which SDP can consider since it is inherently designed for single-objective optimization. To explore multiple objectives, one must perform repeated scalarized single-objective optimizations for each Pareto optimal point, resulting in a combinatorial increase in computational cost as the number of objectives grows (Giuliani et al. 2021).

In recent years, researchers have endeavored to address the challenges associated with SDP to facilitate practical applications in real-world scenarios. We have decided to approach the problem with AI Planning. A very similar domain in which AI Planning has been proven effective is Urban Traffic Control (UTC) (Vallati et al. 2016; McCluskey and Vallati 2017). Both Traffic Control problem and our Water Resources Management problem are optimization problems subject to constraints that share a similar structure; rather than a network of roads and traffic lights we have rivers and reservoirs. In UTC domain, a region of the road network become a directed graph, where vertices stand for intersections, entry or exit points, and edges stand for road sections. Traffic lights control each crossroad, and between each couple of road sections there is a flow rate which determines how many vehicles can go through the intersection in a time unit. The planner must decide when and which traffic light to switch off in order to make the traffic flow, and keep the roads under a certain congestion threshold set in the problem goal. In a similar way, in a water management domain the planner must decide when to open and close the gates of a dam to make the river flow in a controlled manner.

Conclusions and Future Work

In this paper, we studied the feasibility of a planning-based approach for the operation of the Hoa Binh reservoir, a key component of the Red River Basin in Vietnam. Our results indicate that, with suitable domain modeling and offline iterative refinements, current numeric planners can produce plans that are competitive with historical operations under stricter constraint adherence in this case study.

Beyond solution quality, the experiments also highlight a search-centric takeaway: domain-independent guidance can be unreliable on some years (DII times out on 2003 and 2005), while lightweight domain structure can markedly improve robustness (DDI finds and refines plans on all years). This suggests that, for realistic numeric planning with delayed constraints, improving coverage may require combining heuristic guidance with explicit abstractions or control knowledge that reduce plateaus and redundant branching.

Future work includes extending the planning model from a single reservoir to the full Red River Basin, designing heuristics that explicitly exploit reservoir and river-network structure, and moving from open-loop seasonal plans to closed-loop strategies that react to updated inflow forecasts and uncertainty. Another direction is to report richer heuristic search diagnostics (e.g., expansions, plateau statistics, and anytime curves) to better characterise when and why domain-independent guidance fails.

Implications for search. From a heuristic-search viewpoint, the main lesson is that *search-space shaping* can be as important as the choice of a heuristic in large numeric planning tasks with delayed interactions. Two sources of difficulty stand out in this domain: (i) *action-sequence redundancy*, because over the planning horizon many distinct action sequences yield the same aggregate release profile (and hence similar downstream dynamics), and (ii) *delayed feasibility*, because constraint violations can occur only after several days of routing and storage depletion. Both effects can create wide heuristic plateaus for domain-independent numeric relaxations.

Restricting decisions to control days addresses point (i) by compiling stretches of intermediate days into macro-decisions, while iterative tightening addresses point (ii) by turning a hard optimisation problem into a sequence of easier feasibility problems with progressively stronger requirements (Algorithm 1). More generally, these results suggest promising directions for applying numeric planning to water infrastructure control.

Limitations. Our study is intentionally conservative and simplified: we plan in an open-loop setting with fixed inflows for the season, we optimise a single objective (energy) under hard constraints, and we discretise releases with an operational priority. These choices make the search problem well defined and highlight heuristic-search issues, but they also limit policy flexibility. A natural next step is to combine the search-control ideas studied here with receding-horizon replanning and explicit robustness margins to accommodate forecast errors and unmodelled events.

Decision support. Finally, while our optimisation objective is hydropower production, the search-based formulation can naturally be used as a *scenario-exploration* tool: by changing constraints, demand profiles, or tightening schedules, operators can generate alternative feasible policies and compare their trade-offs. More broadly, search provides a mechanism for exploring large structured decision spaces, not only for producing a single "optimal" plan. In particular, rerunning search under alternative thresholds or demand scenarios provides a transparent audit trail (which constraints bind, when, and why) that can complement existing hydrological decision-support workflows.

References

Anonymous. 2025. Reference omitted for double-blind review. Details omitted for double-blind review.

Castelletti, A.; Pianosi, F.; and Soncini-Sessa, R. 2008. Water reservoir control under economic, social and environmental constraints. *Automatica*, 44(6): 1595–1607.

Fox, M.; and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *J. Artif. Intell. Res.*, 20: 61–124.

Giuliani, M.; Lamontagne, J.; Reed, P.; and Castelletti, A. 2021. A state-of-the-art review of optimal reservoir control for managing conflicting demands in a changing world. *Water Resources Research*, 57(12): e2021WR029927.

Hien, N. T.; Yen, N. H.; Balistrocchi, M.; Cat, V. M.; Roberto, R.; et al. 2020. Salinity dynamics under sea level rise scenarios in the Red–Thai Binh River delta, Vietnam. In *Proceedings of the 22nd IAHR-APD Congress 2020, Sapporo, Japan*.

Li, D.; Scala, E.; Haslum, P.; and Bogomolov, S. 2018. Effect-Abstraction Based Relaxation for Linear Numeric Planning. In *IJCAI*, 4787–4793. ijcai.org.

McCluskey, T.; and Vallati, M. 2017. Embedding automated planning within urban traffic management operations. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 27, 391–399.

Piacentini, C.; Alimisis, V.; Fox, M.; and Long, D. 2015. An extension of metric temporal planning with application to AC voltage control. *Artificial intelligence*, 229: 210–245.

Scala, E.; Haslum, P.; Thiébaux, S.; et al. 2016. Heuristics for Numeric Planning via Subgoaling. In *IJCAI*, 3228–3234.

Scala, E.; Saetti, A.; Serina, I.; and Gerevini, A. E. 2020. Search-Guidance Mechanisms for Numeric Planning Through Subgoaling Relaxation. In *ICAPS*, 226–234. AAAI Press.

Uhlenbrook, S.; Connor, R.; et al. 2019. The United Nations world water development report 2019: leaving no one behind.

Vallati, M.; Magazzeni, D.; De Schutter, B.; Chrpa, L.; and McCluskey, T. 2016. Efficient macroscopic urban traffic models for reducing congestion: A PDDL+ planning approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.