A Notion of Width for Numeric Planning Problems

Giacomo Rosa[1,*], **Jean Honorio**[1,*], **Nir Lipovetzky**[1,*], **Enrico Scala**[2,‡], **Sebastian Sardiña**[3,§]

[1]The University of Melbourne, Australia
[2]The University of Brescia, Italy
[3]RMIT University, Melbourne, Australia
[*]{g.rosa, jean.honorio, nir.lipovetzky}@unimelb.edu.au [‡]enrico.scala@unibs.it [§]sebastian.sardina@rmit.edu.au

Abstract

Width provides a structural framework for characterising the complexity of planning problems and explaining the effectiveness of novelty-based exploration. Although novelty has been applied to numeric features, existing work does not extend the underlying width theory to numeric planning. We define *numeric width*, a framework for capturing novelty over features in linear numeric problems. We show that it retains polynomial guarantees under mild conditions, providing the first extension of width theory to numeric planning. Experiments indicate that numeric width captures instance hardness and supports efficient exploration of linear numeric problems.

Introduction

Width-based search (Lipovetzky and Geffner 2012) has become an important approach in classical planning, where novelty-based heuristics are a key component of high-performing solvers (Lipovetzky and Geffner 2017; Katz et al. 2018; Corrêa et al. 2023; Singh et al. 2021; Rosa and Lipovetzky 2024; Rosa et al. 2026). Novelty guides exploration by prioritizing states that introduce new information relative to those previously encountered, enabling efficient coverage of the state space. Crucially, this notion is not only effective in practice, but is also grounded in the concept of *width*.

Width provides a structural characterization of planning problems by measuring the size of feature combinations required to capture goal-directed progress. This perspective explains the effectiveness of novelty: states that are novel with respect to bounded-size feature sets correspond to meaningful progress in low-width domains. As a result, problems of bounded width can be solved efficiently, and simple novelty-based search procedures enjoy guarantees that align closely with their empirical success.

Beyond the classical setting, novelty has also been applied to problems with numeric features, primarily in simulator-based environments (Lipovetzky, Ramirez, and Geffner 2015; Frances et al. 2017; Junyent, Jonsson, and Gómez 2019; Teichteil-Königsbuch, Ramirez, and Lipovetzky 2021), and more recently in numeric planning. Ramirez et al. (2018) define novelty over assignment-based representations that encode each state as a vector of variable values, treating both propositional and numeric variables uniformly. Teichteil-Königsbuch, Ramirez, and Lipovetzky (2021) instead introduce boundary-based features, where novelty is defined in terms of extending previously observed valuation ranges for numeric variables, effectively capturing progress through changes in value intervals. Chen and Thiébaux (2024) show that such novelty definitions over numeric features can be leveraged to improve the performance of heuristic search in numeric planning. Paolini and Scala (2025) test novelty over an alternative feature design which evaluates progress towards *subgoals* derived from the problem description. However, these approaches focus on empirical improvements and do not extend the underlying notion of width to numeric representations.

In this work, we address this gap by extending the concept of width to numeric planning. We define *numeric width*, capturing novelty over numeric planning features through conjunctions over propositional and distance-based subgoal features. Our formulation relies on numeric features similar to those proposed by Paolini and Scala (2025), capturing progress toward subgoals, but introduces a different treatment of feature conjunctions. We show both theoretically and empirically that these notions retain polynomial-time guarantees under conditions commonly satisfied in structured planning domains, providing the first extension of width theory to numeric representations.

Preliminaries

The planning model we adopt represents deterministic, fully observable, sequential planning problems, and is given by $\Phi = \langle S, s_0, S_G, A, f \rangle$ where S is the state space, $s_0 \in S$ is the initial state, and $S_G \subseteq S$ is the set of goal states. The set A contains deterministic actions, and $f : A \times S \mapsto S$ is a partial transition function mapping a state-action pair to its successor. In particular, $f(a, s)$ denotes the state $s' \in S$ obtained by applying action $a \in A$ in state $s \in S$, and is undefined when a is not applicable. We denote by $A(s) = \{a \in A \mid f(a, s) \text{ is defined}\}$ the set of actions applicable in state s. A ***plan*** is a sequence of actions $\langle a_0, \ldots, a_m \rangle$ that generates a state sequence $\langle s_0, \ldots, s_{m+1} \rangle$ such that, for all $i \in \{0, \ldots, m\}$, $a_i \in A(s_i)$ and $s_{i+1} = f(a_i, s_i)$. The plan is a solution if $s_{m+1} \in S_G$.

Classical Planning. A STRIPS[1] (Fikes and Nilsson 1971) classical planning problem is defined as $P = \langle F, O, I, G \rangle$,

[1]Stanford Research Institute Problem Solver

where F is a set of Boolean variables (fluents), O is a set of actions, $I \subseteq F$ specifies the initial state, and $G \subseteq F$ is a partial assignment defining the goal condition.

We assume unit-cost actions, so the cost of a plan coincides with its length. ***Optimal plans*** are therefore solutions of minimum length.

Novelty and Width. We adopt the notion of planning width from (Lipovetzky and Geffner 2012). An atomic conjunction is a set of atoms $t \subseteq F$. The size of the set t is denoted as $|t|$. A state s satisfies t, written $s \models t$, iff $t \subseteq s$. For $i \geq 1$, let

$$\mathcal{T}^i = \{t \subseteq F \mid |t| \leq i\}.$$

Rather than reasoning over states, width is defined over atomic conjunctions. For a conjunction t, let $P(t)$ be the problem obtained from P by replacing the goal with t. A conjunction is reachable if it is satisfied in some reachable state.

The relation between conjunctions is captured by a conjunction graph (Lipovetzky and Geffner 2012),

Definition 1 (Classical Planning Conjunction Graph)**.** *For $P = \langle F, O, I, G \rangle$, the graph $\mathcal{G}^i$ over conjunctions $\mathcal{T}^i$ is defined inductively as follows:*

1. *A conjunction $t \in \mathcal{T}^i$ is a root of $\mathcal{G}^i$ if $t \subseteq I$.*
2. *There is a directed edge $t \to t'$ in $\mathcal{G}^i$ if t is in $\mathcal{G}^i$ and for every optimal plan π for $P(t)$, there exists an action $a \in O$ such that $\pi \cdot a$ is an optimal plan for $P(t')$.*

Thus, all optimal plans achieving t can be extended into optimal plans achieving t'.

The ***width*** of a problem then relies on the constructed conjunction graph, and the concept of ***goal formula implication***: a goal formula G_1 implies G_2 if every optimal plan for $P(G_1)$ is also an optimal plan for $P(G_2)$.

Definition 2 (Optimal Classical Width)**.** *Let ϕ be a formula not satisfied in I. The width of ϕ relative to P is the minimum w such that there exists $t \in \mathcal{G}^w$ that implies ϕ. If ϕ holds in I or is reachable in one step, its width is* 0.

Definition 3 (Problem Width)**.** *The width of P, denoted $w(P)$, is the width of its goal G.*

Theorem 1. *If $w(P) = i$, then P can be solved optimally in time exponential in i (Lipovetzky and Geffner 2012).*

Iterated Width. The above bound is achieved by Iterated Width (IW). IW performs a sequence of breadth-first searches IW(i) for $i = 0, 1, 2, \ldots$ that prune certain states, until a solution is found. A conjunction t is said to be ***novel*** at a state s if $s \models t$ and no previously generated state satisfies t. The novelty of s is the size of the smallest such conjunction. Each IW(i) prunes a generated state s if it does not produce a novel conjunction of size at most i. This is the only modification with respect to standard breadth-first search. For $n = |F|$, IW(i) runs in $O(n^i)$. If $w(P) = w$, IW finds an optimal solution in at most $w + 1$ iterations, with overall complexity $O(n^w)$. Definitions 4 and 5 formalize these concepts.

Definition 4. *A newly generated state s produces a new conjunction of atoms t iff s is the first state generated in the search that makes t true. The size of the smallest new conjunction of atoms produced by s is called the novelty of s. When s does not generate a new conjunction, its novelty is set to n + 1 where n is the number of problem variables.*

Definition 5. *IW(i) is a breadth-first search that prunes newly generated states when their novelty measure is greater than i.*

Numeric Planning and Subgoals. Numeric planning is the fragment of deterministic, fully observable, and sequential planning that allows states to include numeric variables in addition to Boolean fluents. We consider sequential numeric planning with grounded actions (Fox and Long 2003). Let $P = \langle X, O, I, G \rangle$, with $X = X_p \cup X_n$ being the set of variables, where X_p are propositional fluents and X_n are real numeric variables, I specifies the initial state, O is a set of actions, and G is a goal condition.

A state s is a total assignment over X. For each $p \in X_p$, $[p]^s \in \{\top, \bot\}$, and for each $x \in X_n$, $[x]^s \in \mathbb{R}$. For an arithmetic expression ξ over X_n, $[\xi]^s$ denotes its evaluation in s.

A ***literal*** ℓ is either:

- a propositional literal p or $\neg p$, with $p \in X_p$, or
- a numeric atom of the form $\xi \trianglerighteq 0$, where $\trianglerighteq \in \{>, \geq, =\}$ and ξ is an affine expression over X_n.

A ***condition*** is a formula ϕ defined by:

$$\phi := \top \mid \bot \mid \ell \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi.$$

We write $s \models \phi$ if ϕ evaluates to true in s, and for a set of conditions $\mathcal{C}$, $s \models \mathcal{C}$ denotes that $s \models \phi$ for all $\phi \in \mathcal{C}$.

Definition 6 (Numeric Action)**.** *An action is a pair $\langle \mathrm{pre}(a), \mathrm{eff}(a) \rangle$, where $\mathrm{pre}(a)$ is a set of propositional and numeric conditions, and $\mathrm{eff}(a)$ is a set of assignments. A classic assignment is $p = \top$ or $p = \bot$; a numeric assignment is $x = \xi$, where ξ is an arithmetical expression over variables in X_n. We require that $\mathrm{eff}(a)$ does not contain multiple assignments to the same variable.*

In this paper, we focus on the fragment of numeric planning problems known as ***linear numeric planning*** (LNP). In LNP, both preconditions and effects are restricted to linear expressions. In particular, numeric effects update variables according to assignments of the form

$$x \leftarrow \sum_{y \in X_n} \alpha_{x,y} \cdot y + \beta_x,$$

where $\alpha_{x,y}, \beta_x \in \mathbb{R}$. As with width in classical planning, we also add a unit-cost-action assumption to the numeric planning model.

Example 1 (Sailing)**.** *Throughout this work, we use a simplified version of the Sailing domain as a running example. The state contains two numeric variables, x and y, representing the coordinates of a sailing boat on a two-dimensional grid. The initial state is $(x = 0, y = 0)$.*

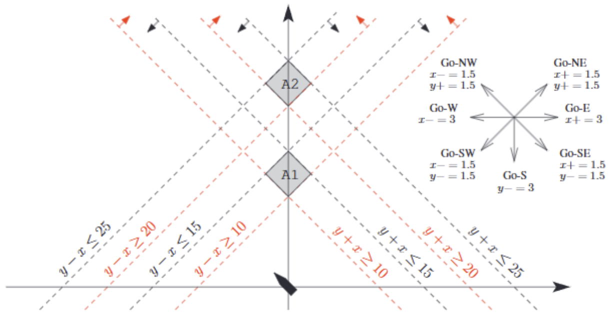


Figure 1: The Sailing domain (image from (Scala, Haslum, and Thiébaux 2016)).

The domain contains directional movement actions with no preconditions. For example, action effects can be:

$$\begin{aligned} \texttt{Go-W}:&\quad x \leftarrow x-3,\ y \leftarrow y,\\ \texttt{Go-NW}:&\quad x \leftarrow x-1.5,\ y \leftarrow y+1.5,\\ \texttt{Go-NE}:&\quad x \leftarrow x+1.5,\ y \leftarrow y+1.5. \end{aligned}$$

All boat movement actions are illustrated in Figure 1, and we assume all actions have unit cost.

The domain also contains an action `Save-P1` *with preconditions*

$$(y-x \geq 10) \wedge (y+x \geq 10) \wedge (y-x \leq 15) \wedge (y+x \leq 15),$$

and effect

$$\texttt{saved}(p1) = \top,$$

whose achievement constitutes the planning goal in our examples. Satisfaction of the preconditions of this action corresponds to the area `A1` *in Figure 1.*

Numeric Width and Novelty

Subgoal Features

Rather than defining novelty over arbitrary facts, we define it over a set of ***subgoals*** $\mathcal{S}$, namely the literals of action preconditions, and the literals present in the goal condition. Throughout the paper, we assume that negated conditions in the form $\neg\phi$ are compiled into a non-negated grounded literal (or literals, for negated equalities).

Write $\mathcal{S} = \mathcal{S}_B \cup \mathcal{S}_N$, where $\mathcal{S}_B$ is the set of propositional subgoals and $\mathcal{S}_N$ is the set of numeric subgoals. For each state s and subgoal g, we define:

- ***Propositional feature.*** For $g \in \mathcal{S}_B \cup \mathcal{S}_N$,

  $$\psi_g(s) \in \{0,1\}, \quad \psi_g(s) = 1 \text{ iff } s \models g.$$

  Note that propositional features are created to evaluate the achievement of both propositional as well as numeric subgoals in a state s.

- ***Distance feature.*** For $g \in \mathcal{S}_N$,

  $$\delta_g(s) = \begin{cases} \max(0, -[\xi_g]^s) & \text{if } g \text{ is an inequality,} \\ |[\xi_g]^s| & \text{if } g \text{ is an equality,} \end{cases}$$

  where $[\xi_g]^s$ is the expression of numeric subgoal g evaluated in state s. Note that we previously defined numeric literals to be in normal form (i.e. $\xi \trianglerighteq 0$), hence the definition for when g is an inequality evaluates the distance to that subgoal's achievement.

Thus, $\delta_g(s) = 0$ iff $s \models g$, and $\psi_g(s) = 1$ iff $\delta_g(s) = 0$.

Numeric Novelty. A propositional feature is said to be ***novel*** when achieved for the first time. A distance feature is said to be ***novel*** when its best seen value is strictly improved, i.e., when for some numeric subgoal g, $\delta_g(s) < \min\{\delta_g(s') : s' \text{ previously generated}\}$. Distance 0 is a floor.

Example 2 (Sailing subgoals and distance features)**.** *Consider again the Sailing domain from Example 1 in Figure 1. The action* `Save-P1` *induces the numeric subgoals*

$$g_1 : y-x \geq 10, \qquad g_2 : y+x \geq 10,$$

$$g_3 : y-x \leq 15, \qquad g_4 : y+x \leq 15,$$

derived from its preconditions, and an additional propositional subgoal $\texttt{saved}(p1) = \top$ *is derived from the problem's goal condition (note that we are disregarding the area* `A2` *in the image). Note that* g_3 *and* g_4 *are trivially satisfied at the initial state, and thus we focus on the achievement of* g_1 *and* g_2*, necessary for achieving the goal condition.*

For a state s*, we define distance features measuring progress toward these subgoals:*

$$\delta_{g_1}(s) = \max(0, 10-(y-x)),$$

$$\delta_{g_2}(s) = \max(0, 10-(y+x)).$$

Actions have different effects on the progress towards subgoals, that is, the reduction in the value of distance features evaluated at different states. At the initial state $(x=0, y=0)$*, both distances are* 10*. After applying* `Go-W`*, the state becomes* $(x=-3, y=0)$*, yielding* $\delta_{g_1} = 7$ *and* $\delta_{g_2} = 13$*, thus improving progress towards one subgoal while worsening progress towards the other. In contrast, after applying action* `Go-NW` *at the initial state, the state becomes* $(x=-1.5, y=1.5)$*, which evaluates to* $\delta_{g_1} = 7$ *and* $\delta_{g_2} = 10$*, thus improving the progress towards* g_1 *without worsening the distance to* g_2*. Other actions, such as* `Go-S`*, then, cannot reduce the value of distance features towards* g_1 *nor* g_2*.*

Conjunction Schemas

In classical width, an atomic conjunction is a set of propositional atoms, and it is said to be satisfied in a state s if all atoms in an atomic conjunction are true in s.

Since we now define novelty over propositional and distance features, we also define conjunctions with two main differences over the classical width counterpart:

1. conjunctions are over propositional and numeric subgoals, rather than propositional atoms,
2. conjunctions may contain a propositional and a distance component, indicative of propositional and distance features.

We start by presenting the ***Boolean component*** of a conjunction, then present the distance component. The Boolean component of a conjunction is a set of subgoals $b \subseteq \mathcal{S}$. We say a state s satisfies b, written $s \models b$, iff $\forall g_k \in b, s \models g_k$; if b is empty, s satisfies b. This is equivalent to saying every propositional feature $\psi_{g_k}(s) = 1$ for subgoals in b. We use notation b^i to denote a Boolean component of size $|b| = i$.

The ***distance component*** of a conjunction is a tuple $d = \langle D, \theta \rangle, \quad D \subseteq \mathcal{S}_N, \quad \theta \in \mathbb{R}$. We say a state s satisfies d, written $s \models d$, iff $\sum_{g \in D} \delta_g(s) \leq \theta$; if $D = \emptyset$, then $s \models d$. We define the size of d as $|d| := |D|$ and use notation d^i to denote a distance component of size i.

Then, we define a ***numeric subgoal conjunction*** of size k to be a tuple $t = \langle b^i, d^j \rangle$, where $k = i + j$. We say t is satisfied in s, written $s \models t$, iff $s \models b^i$ and $s \models d^j$. For simplicity, when using the term *conjunction* in the following sections, we are referring to a *numeric subgoal conjunction*. When the propositional or distance component of the conjunction has a size of 0 (i.e. b^0 or d^0), we omit that component from the tuple (e.g. $\langle b^2, d^0 \rangle$ becomes $\langle b^2 \rangle$).

Conjunction sets. We use Λ^k to denote the set of all possible conjunctions of size $\leq k$ in a problem. That is, Λ^1 corresponds to all possible conjunctions in the form $\langle b^1 \rangle$ and $\langle d^1 \rangle$. Λ^2 contains additionally all conjunctions in the form $\langle b^2 \rangle$, $\langle b^1, d^1 \rangle$, $\langle d^2 \rangle$. Unlike the classical setting, Λ^k is generally infinite due to the real-valued threshold θ in distance components. In particular, if a state satisfies $\langle D, \theta \rangle$, it also satisfies every $\langle D, \theta' \rangle$ with $\theta' > \theta$. However, each state induces a unique tightest bound $\theta_{\min} = \sum_{g \in D} \delta_g(s)$, which is tighter than all larger thresholds over the same structure D. As shown later, this allows the relevant conjunction graph to remain finite under mild assumptions.

Why additive aggregation is used. A Pareto-front improvement criterion for numeric conjunctions of size > 2 may appear a more natural extension of classical novelty to numeric variables, as improvements along one numeric dimension do not mask lack of improvement along others, preserving novelty independently across multiple subgoals. Such a formulation would also more closely resemble approaches such as boundary extension novelty (Teichteil-Königsbuch, Ramirez, and Lipovetzky 2021), which similarly preserve novelty independently across numeric variables, though not with respect to explicit subgoal structures. However, this would require maintaining sets of tight distance vectors, whose size and computational cost may grow rapidly with conjunction size, and would weaken the tightness relations induced between conjunctions that we define in the next section. By contrast, our combination of a Boolean component with a distance component with additive aggregation induces stronger tightness relations through a single scalar ordering over aggregate bounds, that has practical benefits on both time and memory footprint of computations. For practical considerations, as well as for the simplicity and interpretability of the conjunction graph introduced in the next section, we therefore adopt additive aggregation in this work.

Numeric Width

Numeric conjunction graph

We write $\Pi(t)$ to denote the planning problem that is like Π but with goal t (i.e. the last state induced by π satisfies t). We say that π is an optimal plan for $\Pi(t)$ if the last state induced by π satisfies t, and there is no shorter plan π' whose last induced state satisfies t.

Definition 7 (Distance bound tightness)**.** *We say a distance component of a conjunction $d_1^i = \langle D_1, \theta_1 \rangle$* ***is tighter than*** *another distance component $d_2^i = \langle D_2, \theta_2 \rangle$ iff $D_1 = D_2$ and $\theta_1 < \theta_2$. That is, they represent the same numeric subgoal(s), and d_1 is closer to achieving the underlying subgoal than d_2.*

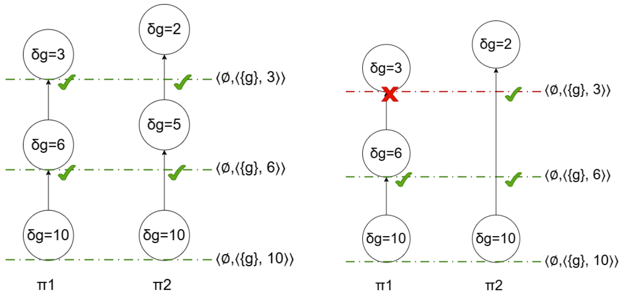
Definition 8 (Conjunction tightness)**.** *We say a conjunction $t = \langle b_1^i, d_1^j \rangle$* ***is tighter than*** *a conjunction $t' = \langle b_2^i, d_2^j \rangle$ iff $b_1^i = b_2^i$ and d_1^j is tighter than d_2^j.*

Definition 9 (Numeric Conjunction Graph)**.** *Conjunction graph $\mathcal{G}^k$ is the graph with vertices from Λ^k defined inductively:*

1. *t is a root vertex in $\mathcal{G}^k$ iff t is true in I,*
2. *$t \rightarrow t'$ is a directed edge in $\mathcal{G}^k$ iff*
   - *t is in $\mathcal{G}^k$ and for every optimal plan π for $\Pi(t)$ there is an action $a \in O$ such that π followed by a is an optimal plan for $\Pi(t')$, and*
   - *no other child node of t is tighter than t'.*

The tight-child-node condition is the key addition to the definition of conjunction graphs over optimal plans in classical problems, addressing our addition of distance components to conjunctions in the graph. For an edge $t_1 \rightarrow t_2$, where $t_2 = \langle b^i, \langle D, \theta \rangle \rangle$, the value of θ is such that all plans that satisfy t_1 optimally can be extended by one action to satisfy t_2 optimally. A subset of these plans *may* satisfy a tighter bound than others with the same plan length; however, the width definition requires that *all* plan extensions satisfy the new bound. Without the tightness requirement, however, the achievement of conjunction t_2 would imply an infinite number of child nodes in the conjunction graph: the set of plans that optimally satisfy conjunction $\langle b^i, \langle D, \theta \rangle \rangle$ are guaranteed to optimally satisfy any conjunction $t_2 = \langle b^i, \langle D, \theta + \epsilon \rangle \rangle$ for a small enough, positive ϵ. We thus add the *tight* conjunction requirement to the definition, such that there is no smaller $\theta' < \theta$ that all extended optimal plans can satisfy with the same plan length.

Example 3 (Conjunction Graph Vertex Extension)**.** ***Case 1:*** *As illustrated in Figure 2a, assume two plans, π_1 and π_2, being the only two plans that satisfy $\Pi(t)$ optimally with plan length c, for a conjunction $t = \langle d_g^1 \rangle = \langle \emptyset, \langle \{g\}, 10 \rangle \rangle$ in the conjunction graph, and subgoal g. Precisely, for states s_1 and s_2 induced by π_1 and π_2, both $\delta_g(s_1) = 10$ and $\delta_g(s_2) = 10$. After one more action, π_1' achieves state s_1', with $\delta_g(s_1') = 6$, and π_2' achieves state s_2', with $\delta_g(s_2') = 5$. In the conjunction graph, we can add a node $t \rightarrow t'$ where $t' = \langle \emptyset, \langle \{g\}, 6 \rangle \rangle$, because both π_1' and π_2' satisfy t' optimally with plan length $c + 1$. Note that all conjunctions $t' = \langle \emptyset, \langle \{g\}, \theta \rangle \rangle$ with $\theta > 6$ are less tight than $\langle \emptyset, \langle \{g\}, 6 \rangle \rangle$. Similarly, after another action, if π_1'' achieves state s_1'', with*



(a) The first case described in Example 3. Both plans can be extended into optimal plans to $\langle\emptyset, \langle\{g\}, 3\rangle\rangle$.

(b) The second case described in Example 3. Plan π_1' cannot be extended into an optimal plan to $\langle\emptyset, \langle\{g\}, 3\rangle\rangle$.

Figure 2: Each node represents a state achieved by plans π_1'' and π_2'' respectively, with $\delta_g = x$ representing $\delta_g(s)$ as described in the example. The horizontal lines, ticks and cross represent whether one action successfully extends a plan into an optimal plan for tuples described in the example.

$\delta_g(s_1') = 3$, and π_2'' achieves state s_2'', with $\delta_g(s_2') = 2$, we can add another edge $t' \rightarrow \langle\emptyset, \langle\{g\}, 3\rangle\rangle$.

Case 2: *Now, assume the same initial situation, with both π_1 and π_2 inducing conjunction $\langle\emptyset, \langle\{g\}, 10\rangle\rangle$ in the conjunction graph. After one more action, π_1' achieves state s_1', with $\delta_g(s_1') = 6$, but now π_2' achieves state s_2', with $\delta_g(s_2') = 2$. We can say that both π_1' and π_2' still achieve $t' = \langle\emptyset, \langle\{g\}, 6\rangle\rangle$ optimally with cost $c{+}1$, and thus the edge $t \rightarrow t'$ persists. However, at step $c{+}2$, plan π_1'' achieves state s_1'', with $\delta_g(s_1') = 3$, but this is no longer an optimal achievement of $\langle\emptyset, \langle\{g\}, 3\rangle\rangle$, because plan π_2' already satisfied this tuple optimally with plan length $c + 1$. Thus, we cannot add an edge $t' \rightarrow t''$ to any conjunction t'' in the conjunction graph, because there is an optimal plan to $\langle\emptyset, \langle\{g\}, 6\rangle\rangle$ that cannot be extended with one action into an optimal plan to any $\langle\emptyset, \langle\{g\}, \theta'\rangle\rangle$ where $\theta' < 6$.*

Corollary 1. *A path $t_0, t_1, ..., t_n$ is in $\mathcal{G}^k$ when t_0 is true in the initial situation and any optimal plan for t_i can be extended by means of a single action into an optimal plan for t_{i+1}, $0 \leq i < n$.*

Numeric Width

The classical width definitions are then the same as Definitions 2 and 3 for the classical planning case, but using the new conjunction graph construction from Definition 9.

As with the definition of classical width in (Lipovetzky and Geffner 2012), it relies on the concept of ***optimal goal formula implication***. We say a goal formula G_1 optimally implies G_2 if every optimal plan for $\Pi(G_1)$ is also an optimal plan for $\Pi(G_2)$.

Boolean Component Formula Implication. For conjunctions with only a Boolean component $b^i = \{g_1, ..., g_i\}$ and an empty distance component, its satisfaction is equivalent to the satisfaction of formula $g_1 \wedge ... \wedge g_i$.

Distance Component Formula Implication. However, we need to align this definition with the distance components of conjunctions introduced in numeric conjunction graphs. For a distance component $d^1 = \langle\{g\}, \theta\rangle$, g is a literal in normal form $\xi \trianglerighteq 0$. Satisfying $\langle\{g\}, \theta\rangle$ then is equivalent to the achievement of a formula $\theta - \delta_g(s) \geq 0$. Intuitively, equalities become areas of space delimited by two inequalities above and below the original equality condition, whereas inequalities shift the previous numeric inequality.

However, for distance components of a conjunction over multiple literals satisfying $\langle D, \theta\rangle$, since we check satisfaction by summing over distance features $\delta_g(s)$, the formula becomes $\theta - \sum_{g_i \in D} \delta_{g_i}(s) \geq 0$. Note that these formulas *are not linear*, and introduce $\max$ operators and absolute values which are normally not present in linear numeric action preconditions and goal conditions. Still, these are *piecewise linear* formulas that define regions of a state space that may imply other formulas. They thus represent an additional theoretical tool, in addition to the more intuitive formulas derived from the satisfaction of conjunctions with only Boolean components, to define the width of a problem through optimal implication.

Definition 10 (Numeric Width of a Formula)**.** *Let ϕ be a formula not satisfied in I. The width of ϕ relative to P is the minimum w such that there exists $t \in \mathcal{G}^w$ that optimally implies ϕ. If ϕ holds in I or is reachable in one step, its width is* 0.

Definition 11 (Numeric Width of a Problem)**.** *The width of P, denoted $w(P)$, is the width of its goal G.*

Observation 1. *When applied to a problem without numeric variables or conditions, the numeric width is equivalent to classical width.*

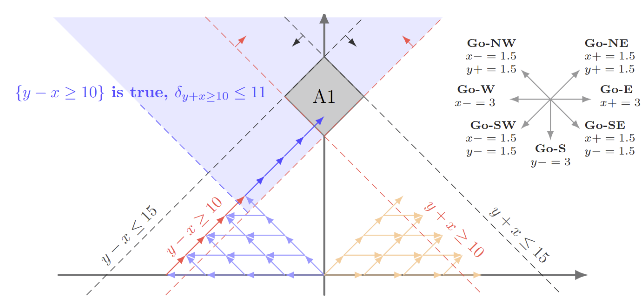


Figure 3: The dark blue arrows represent actions that extend optimal plans to subgoal $y - x \geq 10$ and achieve new conjunctions optimally in the width 2 conjunction graph of *Sailing*. The red arrows represent actions extending optimal plans to $y - x \geq 10$ that, however, induce *suboptimal* plans that satisfy conjunctions $\langle\{y - x \geq 10\}, \langle\{y + x \geq 10\}, \theta\rangle\rangle$ for some $\theta < 22$.

Example 4 (Sailing Conjunction Graph Width 1)**.** *Consider again the Sailing domain from the previous examples, and the goal $A1$, requiring satisfaction of initially unsatisfied numeric subgoals*

$$g_1 : y - x \geq 10, \qquad g_2 : y + x \geq 10.$$

At the initial state ($x = 0, y = 0$), both subgoals have distance 10*. With width 1, conjunctions consist only of $\langle b^1\rangle$ and*

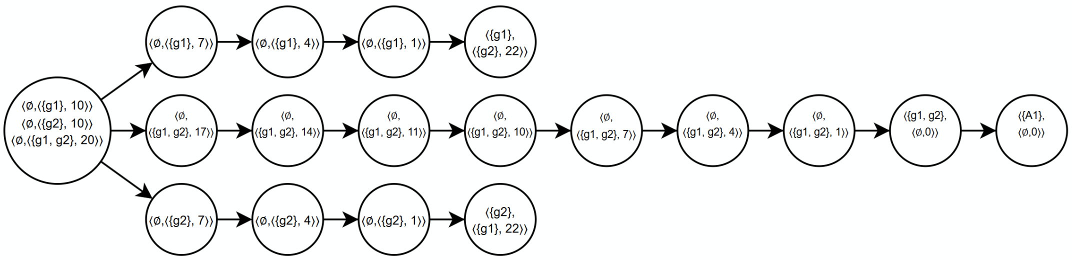


Figure 4: Portion of Width-2 Numeric Conjunction Graph $\mathcal{G}^2$ for Example 5.

$\langle d^1 \rangle$ components. Thus, novelty distinguishes only achievement of individual subgoals and improvements of individual distance bounds. Consider first conjunctions associated with g_1. Actions `Go-W` *and* `Go-NW` *both reduce δ_{g_1} optimally, for example from 10 to 7, then 4, then 1, and finally 0. Hence, any sequence of four such actions is optimal for achieving g_1. However, these optimal plans to g_1 cannot be extended into optimal plans toward the conjunction of g_1 and g_2. Optimal improvements toward g_2 from the initial state instead require actions such as* `Go-E` *and* `Go-NE`*. Thus, after spending four actions optimally achieving g_1, any subsequent progress toward g_2 yields a plan whose total cost exceeds the optimal cost of achieving the same distance bounds towards g_2. Width 1 therefore explains optimal progress toward each subgoal independently, but not toward their joint achievement, and thus the goal condition* `A1`*.*

Example 5 (Sailing Conjunction Graph Width 2)**.** *At width 2, conjunctions additionally include $\langle b^2 \rangle$, $\langle b^1, d^1 \rangle$, and $\langle d^2 \rangle$ components. Consider first conjunctions of the form*

$$\langle \{g_1\}, \langle \{g_2\}, \theta \rangle \rangle,$$

which capture states that have achieved g_1 while improving the distance bound toward g_2. These still fail to induce a complete optimal extension path. Although both `Go-W` *and* `Go-NW` *achieve $\psi_{g_1} = 1$ optimally, only plans using repeated* `Go-NW` *preserve future optimal progress toward g_2. Plans using* `Go-W` *only achieve suboptimal improvements toward g_2 (represented by the blue dotted line in Figure 3). Hence, not all optimal plans achieving $\psi_{g_1} = 1$ can be extended into optimal plans achieving lower bounds for g_2.*

Now consider the set of $\langle d^2 \rangle$ conjunctions of the form

$$\langle \emptyset, \langle \{g_1, g_2\}, \theta \rangle \rangle,$$

where satisfaction is determined by the aggregate distance $\delta_{g_1}(s) + \delta_{g_2}(s) \leq \theta$. From the initial state, this aggregate distance is 20. From Example 3, only combinations of `Go-NW`*,* `Go-NE` *satisfy tight conjunctions that improve this bound optimally. Along the plan* `Go-NW` *$\times$ 4, the aggregate θ decreases from 20 to 17, then 14, then 11, and finally 10. This occurs because* `Go-NW` *improves g_1 without worsening g_2: once δ_{g_1} decreases, δ_{g_2} remains unchanged due to the $\max(0, -\xi_g(s))$ definition of distance features. Crucially, every optimal plan achieving one aggregate bound can be extended into an optimal plan achieving the next bound. Unlike $\langle b^1, d^1 \rangle$ conjunctions, the aggregate distance captures progress toward the conjunction itself, rather than toward one subgoal conditioned on another. Consequently, the goal is reached through a sequence of optimal improvements over $\langle d^2 \rangle$ conjunctions, and thus $A1$ has numeric width 2.*

Figure 4 represents the numeric conjunction graph built for this problem, which parallels this example.

Theorem 2 (Sailing numeric width)**.** *Atomic goals in Sailing have a numeric width of 2.*

Proof sketch. Single atom goals in sailing consist of areas such as $A1$, in any achievable position on the (x, y)-axis (avoiding areas too small to be achieved given movement constraints). A symmetric argument to that in Example 5 without loss of generality proves the theorem for any valid position of $A1$ and any number of boats, as it will optimally extend plans belonging to the boat which is closer to the joint distance component $\{g_1, g_2\}$. □

Conjunction Complexity

As previously presented, our definition of conjunction implies an infinite set of conjunctions of bounded size Λ^k, due to the bound $\theta \in \mathbb{R}$ in the distance component of conjunctions $d^i = \langle D, \theta \rangle$. Still, we have that

1. for every $D \subseteq \mathcal{S}_N$, if a state s satisfies a conjunction $t = \langle D, \theta_{min} \rangle$, where $\sum_{g_i \in D} \delta_{g_i}(s) = \theta_{min}$, then t is tighter than all other conjunctions $\langle D, \theta' \rangle$ satisfied by s where $\theta' > \theta_{min}$, and does not satisfy any conjunction $\langle D, \theta'' \rangle$ where $\theta'' < \theta_{min}$;
2. in a numeric conjunction graph, an edge to a new conjunction $t \rightarrow t'$ requires that t' is tighter than any other child node of t.

Since θ defines a bound that can only improve through decreasing values, these properties allow the number of conjunctions appearing in a numeric conjunction graph $\mathcal{G}^k$ to be bounded in terms of the maximum number of achievable improvements in a planning problem. Given the fact that we are working with linear numeric planning, it is important to note that this number does not have to be bounded, but basic assumptions on the boundedness of improvements usually hold in numeric planning domains.

Let $n_B = |\mathcal{S}_B \bigcup \mathcal{S}_N|$, $n_N = |\mathcal{S}_N|$, $n = n_B + n_N$. That is, there is one propositional feature for each boolean and numeric subgoal, and one distance feature for each numeric subgoal. Let the ***subgoal structure*** of a conjunction in Λ^k denote the combination of subgoals appearing in its propositional and distance components, disregarding the value θ.

For a conjunction of size k containing j propositional subgoals, and $k-j$ numeric subgoals, the number of distinct subgoal structures is $\binom{n_B}{j}\binom{n_N}{k-j}$ Then, the total number of subgoal structures across all conjunctions of size $\leq k$ is

$$\sum_{l=0}^{k}\sum_{j=0}^{l}\binom{n_B}{j}\binom{n_N}{l-j} = \sum_{l=0}^{k}\binom{n}{l} \in O(n^k),$$

and is thus bounded polynomially in n, and exponentially in k, and does not introduce an additional asymptotic factor.

Bound multiplicity. The additional factor in the asymptotic number of conjunctions in the numeric conjunction graph, as opposed to its classical counterpart, comes from numeric bounds θ in the distance component of conjunctions. However, only bounds that are tight and that correspond to strict improvements can appear in the numeric conjunction graph. Let $\mathcal{D}_k$ denote an upper bound, over all numeric structures D of size at most k, on the number of distinct tight improving bounds θ that can appear for D. The number of conjunctions of size at most k appearing in the graph is then bounded by $O(\mathcal{D}_k n^k)$.

Proposition 1. *The number of conjunctions in the numeric conjunction graph is bounded by* $O(\mathcal{D}_k n^k)$.

Bounding $\mathcal{D}_k$. Let V upper-bound the initial aggregate distance of every numeric structure D with $|D| \leq k$, i.e., $s_0 \models \langle D, V\rangle$. Let $\varepsilon > 0$ lower-bound every strict optimal improvement appearing in the numeric conjunction graph, over all such D. Then the number of tight improving bounds that may appear for any such D is bounded by

$$\mathcal{D}_k \leq \left\lceil \frac{V}{\varepsilon} \right\rceil.$$

Thus, bounded chains of aggregate improvements play the role that Boolean finiteness plays in classical width. While this requirement does not hold for arbitrary numeric planning problems, many structured domains naturally satisfy it. This includes *simple numeric planning* problems, which strengthen linear numeric planning by restricting numeric effects to fixed assignments and additive updates over constants. In finite grounded instances, such fixed action effects induce a minimum improvement granularity, as they admit only finitely many possible optimal improvements toward subgoal conjunction bounds. The *Sailing* domain, used as a running example, satisfies this condition naturally: movement occurs over a bounded two-dimensional lattice with fixed step sizes, yielding a bounded and predictable number of distance improvements.

The distance term $\mathcal{D}_k$ is not merely a limitation, but a necessity when defining complexity in numeric domains. Even in *simple numeric planning*, a problem with one numeric variable may require one action to be repeated $1{,}000$ times, so the number of variables alone cannot capture the cost of numeric progress. Indeed, translations of SNP problems to classical planning (Bonassi, Percassi, and Scala 2025) must encode this bounded numeric range propositionally: under unary discretisation, this progress appears as $O(\mathcal{D}_k n)$ propositional atoms; under binary bit-blasting, it uses only $O(n \log \mathcal{D}_k)$ Boolean variables but requires additional propositional structure to encode arithmetic.

Effective Numeric Width

Definition 12. $\mathrm{IW}(i)$ *is a breadth-first search that prunes newly generated states when their numeric novelty measure is greater than i.*

Definition 13 (Effective Numeric Width)**.** *The **effective numeric width** of a numeric planning problem P is the minimum integer w such that* $\mathrm{IW}(w)$ *solves P.*

Proposition 2. *Let t be a conjunction appearing in a numeric conjunction graph $\mathcal{G}^w$. Any state first achieving t is novel for* $\mathrm{IW}(w)$.

Proof. By construction, a tight conjunction t represents a conjunction bound that has not previously been satisfied by a shorter plan. Therefore, when a state first achieves t, no previously generated state belonging to a shorter plan could satisfy that conjunction, or a conjunction tighter than t. Hence the conjunction is novel under $\mathrm{IW}(|t|)$. ☐

Proposition 3. *If a problem P has numeric width w, then its effective numeric width is at most w.*

Proof. By definition of numeric width, there exists a sequence of conjunctions of size at most w such that every optimal plan achieving one conjunction can be extended into an optimal plan achieving the next. Therefore, $\mathrm{IW}(w)$ cannot prune all states along this sequence, since each step produces a conjunction of novelty at most w. This guarantees achievement of all successor conjunctions in the conjunction graph: either the current optimal path extends to the successor conjunction, making it novel, or another optimal path achieves that conjunction earlier (with same cost). Hence, $\mathrm{IW}(w)$ solves P optimally, and the minimum width at which IW solves P is at most w. ☐

Theorem 3. *For a solvable problem P with numeric width w,* $\mathrm{IW}(w)$ *solves P optimally in time exponential in w, polynomial in n, and linear in $\mathcal{D}_k$.*

Proof. Follows from Propositions 1 and 3. ☐

Empirical Study

We perform an empirical analysis using an $\mathrm{IW}(w)$ solver to study the width of numeric planning problems. We implemented $\mathrm{IW}(w)$ in the JPDDLPLUS planning library (Scala 2024). As in (Lipovetzky and Geffner 2012), we focus on *single atom goal* decompositions of problems. These problems are obtained by decomposing each original instance with a conjunctive goal condition $g_1 \wedge \cdots \wedge g_n$ into n instances, each identical except for the goal, which is set to the single condition g_i for instance i. We use the set of problems from the satisficing track of the *International Planning Competition (IPC) 2023* (Taitler et al. 2024) as base instances from which to obtain the single atom goal instances for our experiment. The *IPC 2023* dataset contains 20 domains, with 20 instances each.

Domain	I	$w = 0$	$w = 1$	$w = 2$	$w > 2$
block-grouping	6850	75%	25%	0%	0%
counters	428	38%	62%	0%	0%
delivery	460	2%	0%	98%	0%
drone	1382	4%	16%	46%	34%
expedition	40	0%	0%	0%	100%
plant watering	284	7%	0%	2%	91%
farmland	130	43%	57%	0%	0%
fo-counters	210	0%	100%	0%	0%
fo-farmland	148	42%	58%	0%	0%
fo-sailing	50	0%	0%	0%	100%
hydropower	20	0%	0%	0%	100%
markettrader	20	0%	0%	0%	100%
mprime	30	6%	0%	72%	22%
pathways metric	364	0%	0%	12%	88%
rover	179	0%	64%	36%	0%
sailing	118	0%	0%	100%	0%
settlers numeric	198	0%	18%	14%	68%
sugar	58	0%	6%	59%	35%
tpp	490	4%	13%	45%	38%
zenotravel	298	10%	6%	84%	0%
Summary	10248	11.5%	21.3%	28.4%	38.7%

Table 1: Effective width of single-atom goal instances. I denotes the number of instances for each domain, and $w = x$ the percentage of problems with effective numeric width x. An effective numeric width of 0 corresponds to goals already satisfied at the initial state. The summary row reports the average percentage of instances with a given effective numeric width across domains, giving equal weight to each domain independently of its number of instances. Results indicate that a substantial proportion of single-atom goal instances have effective numeric width ≤ 2.

Single-atom Instance Width. Table 1 presents the proportion of problems solved by a numeric IW(w) planner. On average, 61.3% of single-atom goals across domains are solved by IW(w) with $w \leq 2$, suggesting that the structural relevance of width observed in classical planning, namely its ability to capture compact solution paths through low-order novelty, also extends meaningfully to numeric planning under our framework. In particular, the results for *Sailing* match our analysis (Theorem 2), with all single-atom instances exhibiting an effective numeric width of 2.

High Atomic Width Domains. We still note 7 domains — *expedition, plant watering, first-order-sailing, hydropower, market trader, pathways metric*, and *settlers numeric* — which exhibit high effective numeric width ($w > 2$) in most or all tested single-atom instances. Many of these domains contain few conjunctive goal conditions in their original instances, making single-atom goals substantially more challenging and, in domains such as *hydropower* and *market trader*, effectively equivalent to solving the satisficing instances themselves. This is not always the case for domains with few atomic goals, however: *mprime* expands from 20 original instances to only 30 single-atom instances, yet still exhibits 78% of instances with effective numeric width ≤ 2. By contrast, *pathways metric* and *settlers numeric* decompose into many single-atom instances, the majority of which nevertheless retain high effective numeric width.

Sailing vs. First-order Sailing. It is also informative to compare *Sailing* with *First-order Sailing* (*fo-sailing*) to better understand the domains and our notion of numeric width. *Fo-Sailing* introduces a speed variable v, together with acceleration and deceleration actions that modify movement magnitude. Movement actions therefore become velocity-dependent, e.g.

$$\texttt{go-NE:}(x, y) \leftarrow (x + 1.5 \cdot v,\ y + 1.5 \cdot v).$$

The actions `Accelerate` and `Decelerate` increase or decrease the velocity respectively, and the goal action additionally requires the velocity inside the target region to satisfy $v \leq 1$.

Unlike base *Sailing*, *fo-sailing* exhibits empirical width > 2 for every single-atom instance. We hypothesize that novelty preserves large-magnitude movement bounds induced by acceleration, effectively favouring continued acceleration toward subgoals while pruning slower movement. As a result, the boat overshoots the goal region rather than satisfying the final bounded-area condition.

Importantly, the additional precondition $v \leq 1$ introduces a further relevant subgoal that can preserve the novelty of slower movement and deceleration. However, since the underlying 2-dimensional movement problem in *Sailing* already requires width 2, guaranteeing progress toward the conjunction of movement and velocity-control conditions would require conjunctions additionally conditioned on $[v \leq 1] = \top$, effectively requiring width 3. Furthermore, this progress may not be optimal, achieving an effective numeric width that guarantees achievement, but not optimal achievement, of conjunctions and goals. This provides a direction for future extensions of our framework, either through a pareto-set of numeric bound improvements, or a notion of numeric width that admits suboptimal conjunction achievement.

Conclusion

In this work, we extended the notion of width to numeric planning, providing a principled foundation for novelty over numeric features. Our definition of numeric width recovers polynomial-time guarantees similar to those in the classical setting under mild conditions, with an additional factor that accounts for progress towards the satisfaction of numeric conditions. Empirically, these notions capture structural properties of numeric planning instances and can significantly reduce the search space, particularly in domains with decomposable goals. This is proven theoretically for the $Sailing$ domain, and captured empirically over single-atom goal decompositions of satisficing numeric planning problems. Our work establishes numeric width as a meaningful measure of problem hardness beyond propositional planning, and a template for alternative definitions of numeric width and related techniques.

References

Bonassi, L.; Percassi, F.; and Scala, E. 2025. BLAST: Bit-Blasting Numbers for Classical Planning. In *Proceedings of the International Symposium on Combinatorial Search*, volume 18, 249–250.

Chen, D. Z.; and Thiébaux, S. 2024. Novelty heuristics, multi-queue search, and portfolios for numeric planning. In *Proceedings of the International Symposium on Combinatorial Search*, volume 17, 203–207.

Corrêa, A. B.; Francès, G.; Hecher, M.; Longo, D. M.; and Seipp, J. 2023. Scorpion Maidu: Width Search in the Scorpion Planning System. In *Proceedings of the 10th International Planning Competition (IPC-2023)*.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence (AIJ)*, 2(3-4): 189–208.

Fox, M.; and Long, D. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20: 61–124.

Frances, G.; Ramírez Jávega, M.; Lipovetzky, N.; and Geffner, H. 2017. Purely declarative action descriptions are overrated: Classical planning with simulators. International Joint Conferences on Artificial Intelligence Organization (IJCAI).

Junyent, M.; Jonsson, A.; and Gómez, V. 2019. Deep policies for width-based planning in pixel domains. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, 646–654.

Katz, M.; Lipovetzky, N.; Moshkovich, D.; and Tuisov, A. 2018. Merwin planner: Mercury enchanced with novelty heuristic. *IPC 2018 planner abstracts*, 53–56.

Lipovetzky, N.; and Geffner, H. 2012. Width and serialization of classical planning problems. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 540–545.

Lipovetzky, N.; and Geffner, H. 2017. Best-first width search: Exploration and exploitation in classical planning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 31.

Lipovetzky, N.; Ramirez, M.; and Geffner, H. 2015. Classical planning with simulators: Results on the atari video games. In *Proc. IJCAI*.

Paolini, A.; and Scala, E. 2025. Exploration and Exploitation in Numerical Planning - Master Thesis. In *Universitá degli studi di Brescia*.

Ramirez, M.; Papasimeon, M.; Lipovetzky, N.; Benke, L.; Miller, T.; Pearce, A. R.; Scala, E.; Zamani, M.; et al. 2018. Integrated Hybrid Planning and Programmed Control for Real Time UAV Maneuvering. In *AAMAS*, 1318–1326.

Rosa, G.; Honorio, J.; Lipovetzky, N.; and Sardiña, S. 2026. Goal Recognition as Planning: Deriving Heuristics from Intention Models - Extended Version. In *arXiv preprint arXiv:2603.14824*. https://arxiv.org/abs/2603.14824.

Rosa, G.; and Lipovetzky, N. 2024. Count-based novelty exploration in classical planning. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 4181–4189.

Scala, E. 2024. The JPDDLPLUS API. https://github.com/hstairs/jpddlplus. Accessed: 2026-05-05.

Scala, E.; Haslum, P.; and Thiébaux, S. 2016. Heuristics for numeric planning via subgoaling.

Singh, A.; Lipovetzky, N.; Ramirez, M.; and Segovia-Aguas, J. 2021. Approximate novelty search. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, 349–357.

Taitler, A.; Alford, R.; Espasa, J.; Behnke, G.; Fišer, D.; Gimelfarb, M.; Pommerening, F.; Sanner, S.; Scala, E.; Schreiber, D.; et al. 2024. The 2023 International Planning Competition. https://ipc2023.github.io/.

Teichteil-Königsbuch, F.; Ramirez, M.; and Lipovetzky, N. 2021. Boundary extension features for width-based planning with simulators on continuous-state domains. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 4183–4189.